
ATLANTIS THINKING MACHINES

VOLUME 4

SERIES EDITOR: KAI-UWE KÜHNBERGER

Atlantis Thinking Machines

Series Editor:

Kai-Uwe Kühnberger

Institute of Cognitive Science

University of Osnabrück, Germany

(ISSN: 1877-3273)

Aims and scope of the series

This series publishes books resulting from theoretical research on and reproductions of general Artificial Intelligence (AI). The book series focuses on the establishment of new theories and paradigms in AI. At the same time, the series aims at exploring multiple scientific angles and methodologies, including results from research in cognitive science, neuroscience, theoretical and experimental AI, biology and from innovative interdisciplinary methodologies.

For more information on this series and our other book series, please visit our website at:

www.atlantis-press.com/publications/books



AMSTERDAM – PARIS – BEIJING

© ATLANTIS PRESS

www.allitebooks.com

Theoretical Foundations of Artificial General Intelligence

Pei Wang (Ed.)

Department of Computer and Information Sciences, Temple University,
1805 N. Broad Street, Philadelphia, PA 19122, USA

Ben Goertzel (Ed.)

Novamente LLC/Biomind LLC,
1405 Bernerd Place, Rockville, MD 20851, USA



AMSTERDAM – PARIS – BEIJING

Atlantis Press

8, square des Bouleaux
75019 Paris, France

For information on all Atlantis Press publications, visit our website at: www.atlantis-press.com

Copyright

This book, or any parts thereof, may not be reproduced for commercial purposes in any form or by any means, electronic or mechanical, including photocopying, recording or any information storage and retrieval system known or to be invented, without prior permission from the Publisher.

Atlantis Thinking Machines

Volume 1: Enaction, Embodiment, Evolutionary Robotics. Simulation Models for a Post-Cognitivist Science of Mind - Marieke Rohde, Ezequiel A. Di Paolo

Volume 2: Real-World Reasoning: Toward Scalable, Uncertain Spatiotemporal, Contextual and Causal Inference - Ben Goertzel, Nil Geisweiller, Lúcio Coelho, Predrag Janicic, Cassio Pennachin

Volume 3: Integration of World Knowledge for Natural Language Understanding - Ekaterina Ovchinnikova

ISBNs

Print: 978-94-91216-61-9

E-Book: 978-94-91216-62-6

ISSN: 1877-3273

© 2012 ATLANTIS PRESS

Contents

1. Introduction	1
<i>Pei Wang and Ben Goertzel</i>	
1.1 The Matter of Artificial General Intelligence	1
1.2 The Matter of Theoretical Foundation	3
1.3 The Matter of Objective	4
1.4 The Matter of Approach	5
1.5 Challenges at the Heart of the Matter	6
1.6 Summary	7
Bibliography	8
2. Artificial Intelligence and Cognitive Modeling Have the Same Problem	11
<i>Nicholas L. Cassimatis</i>	
2.1 The Intelligence Problem	11
2.2 Existing Methods and Standards are not Sufficient	13
2.3 Cognitive Modeling: The Model Fit Imperative	16
2.4 Artificial Intelligence and Cognitive Modeling Can Help Each Other	21
2.5 Conclusions	24
Bibliography	24
3. The Piaget-MacGuyver Room	25
<i>Selmer Bringsjord and John Licato</i>	
3.1 Introduction	25
3.2 More on Psychometric AGI	28
3.3 Descartes' Two Tests	34
3.4 Piaget's View of Thinking & The Magnet Test	35
3.5 The LISA model	38
3.6 Analogico-Deductive Reasoning in the Magnet Test	39
3.7 Next Steps	45
Bibliography	46

4. Beyond the Octopus: From General Intelligence toward a Human-like Mind	49
<i>Sam S. Adams and Steve Burbeck</i>	
4.1 Introduction	49
4.2 Octopus Intelligence	50
4.3 A “Ladder” of Intelligence	53
4.4 Linguistic Grounding	55
4.5 Implications of the Ladder for AGI	57
4.6 Conclusion	63
Bibliography	64
5. One Decade of Universal Artificial Intelligence	67
<i>Marcus Hutter</i>	
5.1 Introduction	68
5.2 The AGI Problem	70
5.3 Universal Artificial Intelligence	74
5.4 Facets of Intelligence	77
5.5 Social Questions	79
5.6 State of the Art	81
5.7 Discussion	83
Bibliography	84
6. Deep Reinforcement Learning as Foundation for Artificial General Intelligence	89
<i>Itamar Arel</i>	
6.1 Introduction: Decomposing the AGI Problem	89
6.2 Deep Learning Architectures	91
6.3 Scaling Decision Making under Uncertainty	94
6.4 Neuromorphic Devices Scaling AGI	99
6.5 Conclusions and Outlook	101
Bibliography	102
7. The LIDA Model as a Foundational Architecture for AGI	103
<i>Usef Faghihi and Stan Franklin</i>	
7.1 Introduction	103
7.2 Why the LIDA Model May Be Suitable for AGI	104
7.3 LIDA Architecture	105
7.4 Cognitive Architectures, Features and the LIDA Model	108
7.5 Discussion, Conclusions	117
Bibliography	118

8. The Architecture of Human-Like General Intelligence	123
<i>Ben Goertzel, M. Iklé, and J. Wignore</i>	
8.1 Introduction	123
8.2 Key Ingredients of the Integrative Human-Like Cognitive Architecture Diagram . .	125
8.3 An Architecture Diagram for Human-Like General Intelligence	128
8.4 Interpretation and Application of the Integrative Diagram	136
8.5 Cognitive Synergy	138
8.6 Why Is It So Hard to Measure Partial Progress Toward Human-Level AGI?	140
8.7 Conclusion	143
Bibliography	144
9. A New Constructivist AI	145
<i>Kristinn R. Thórisson</i>	
9.1 Introduction	145
9.2 The Nature of (General) Intelligence	147
9.3 Constructionist AI: A Critical Look	151
9.4 The Call for a New Methodology	156
9.5 Towards a New Constructivist AI	158
9.6 Conclusions	167
Bibliography	169
10. Towards an Actual Gödel Machine Implementation	173
<i>Bas R. Steunebrink and Jürgen Schmidhuber</i>	
10.1 Introduction	173
10.2 The Gödel Machine Concept	175
10.3 The Theoretical Foundations of Self-Reflective Systems	178
10.4 Nested Meta-Circular Evaluators	184
10.5 A Functional Self-Reflective System	186
10.6 Discussion	191
Appendix: Details of Notation Used	192
Bibliography	194
11. Artificial General Intelligence Begins with Recognition	197
<i>Tsvi Achler</i>	
11.1 Introduction	197
11.2 Evaluating Flexibility	200
11.3 Evaluation of Flexibility	209
11.4 Summary	215
Bibliography	216

12. Theory Blending as a Framework for Creativity in Systems for General Intelligence	219
<i>Maricarmen Martinez et al.</i>	
12.1 Introduction	219
12.2 Productivity and Cognitive Mechanisms	221
12.3 Cross-Domain Reasoning	222
12.4 Basic Foundations of Theory Blending	226
12.5 The Complex Plane: A Challenging Historical Example	228
12.6 Outlook for Next Generation General Intelligent Systems	233
12.7 Conclusions	238
Bibliography	238
13. Modeling Emotion and Affect	241
<i>Joscha Bach</i>	
13.1 Introduction	241
13.2 Emotion and Affect	244
13.3 Affective States Emerging from Cognitive Modulation	246
13.4 Higher-Level Emotions Emerging from Directing Valenced Affects	250
13.5 Generating Relevance: the Motivational System	252
13.6 Motive Selection	257
13.7 Putting it All Together	259
Bibliography	261
14. AGI and Machine Consciousness	263
<i>Antonio Chella and Riccardo Manzotti</i>	
14.1 Introduction	263
14.2 Consciousness	264
14.3 Machine Consciousness	267
14.4 Agent's Body	270
14.5 Interactions with the Environment	271
14.6 Time	273
14.7 Free Will	274
14.8 Experience	275
14.9 Creativity	277
14.10 Conclusions	279
Bibliography	280
15. Human and Machine Consciousness as a Boundary Effect in the Concept Analysis Mechanism	283
<i>Richard Loosemore</i>	
15.1 Introduction	283
15.2 The Nature of Explanation	288
15.3 The Real Meaning of Meaning	297
15.4 Some Falsifiable Predictions	301
15.5 Conclusion	303
Bibliography	304

16. Theories of Artificial Intelligence	305
<i>Pei Wang</i>	
16.1 The Problem of AI Theory	305
16.2 Nature and Content of AI Theories	308
16.3 Desired Properties of a Theory	312
16.4 Relations among the Properties	317
16.5 Issues on the Properties	318
16.6 Conclusion	320
Bibliography	321
Index	325

Chapter 1

Introduction: What Is the Matter Here?

Pei Wang¹ and Ben Goertzel²

¹ *Temple University, USA*

² *Novamente LLC, USA*

pei.wang@temple.edu, ben@goertzel.org

This chapter provides a general introduction to the volume, giving an overview of the AGI field and the current need for exploration and clarification of its foundations, and briefly summarizing the contents of the various chapters.

1.1 The Matter of Artificial General Intelligence

Artificial General Intelligence (AGI), roughly speaking, refers to AI research and development in which “intelligence” is understood as a general-purpose capability, not restricted to any narrow collection of problems or domains, and including the ability to broadly generalize to fundamentally new areas [4]. The precise definition of AGI is part of the subject matter of the AGI field, and different theoretical approaches to AGI may embody different slants on the very concept of AGI. In practical terms, though, the various researchers in the field share a strong common intuition regarding the core concerns of AGI – and how it differs from the “narrow AI” that currently dominates the AI field.

In the earliest days of AI research, in the middle of the last century, the objective of the field was to build “thinking machines” with capacity comparable to that of the human mind [2, 6, 9]. In the decades following the founding of the AI field, various attempts arose to attack the problem of human-level artificial general intelligence, such as the General Problem Solver [7] and the Fifth Generation Computer Systems [3]. These early attempts failed to reach their original goals, and in the view of most AI researchers, failed to make dramatic conceptual or practical progress toward their goals. Based on these experiences,

the mainstream of the AI community became wary of overly ambitious research, and turned toward the study of domain-specific problems and individual cognitive functions. Some researchers view this shift as positive, arguing that it brought greater rigor to the AI field – a typical comment being that “it is now more common to build on existing theories than to propose brand new ones, to base claims on rigorous theorems or hard experimental evidence rather than on intuition, and to show relevance to real-world applications rather than toy examples.” [8]. However, an alternate view would be that this greater focus on narrow problems and mathematical and experimental results has come at a great cost in terms of conceptual progress and practical achievement. The practical achievements of applied AI in the last decades should not be dismissed lightly, nor should be the progress made in various specialized AI algorithms. Yet, ultimately, the mounting corpus of AI theorems and experimental results about narrow domains and specific cognitive processes has not led to any kind of clear progress toward the initial goals of the AI field. AI as a whole does not show much progress toward its original goal of general-purpose systems, since the field has become highly fragmented, and it is not easy, if possible at all, to put the parts together to get a coherent system with general intelligence [1].

Outside the mainstream of AI, a small but nontrivial set of researchers has continued to pursue the perspective that intelligence should be treated as a whole. To distinguish their work from the bulk of AI work focused on highly specific problems or cognitive processes (sometimes referred to as “Narrow AI”), the phrase “Artificial General Intelligence” (AGI) has sometimes been used. There have also been related terms such as “Human-Level AI” [5]. The term AGI is meant to stress the general-purpose nature of intelligence – meaning that intelligence is a capacity that can be applied to various (though not necessarily all possible) environments to solve problems (though not necessarily being absolutely correct or optimal). Most AGI researchers believe that general-purpose intelligent systems cannot be obtained by simply bundling special-purpose intelligent systems together, but have to be designed and developed differently [11]. Though AGI projects share many problems and techniques with conventional AI projects, they are conceived, carried out, and evaluated differently. In recent years, the AGI community has significantly grown, and now has its regular conferences and publications.

1.2 The Matter of Theoretical Foundation

Like all fields of science and technology, AGI relies on a subtle interplay of theory and experiment. AGI has an engineering goal, the building of practical systems with a high level of general intelligence; and also a scientific goal, the rigorous understanding of the nature of general intelligence, and its relationship with an intelligent system's internal structures and processes, and the properties of its environment. This volume focuses on the theoretical aspect of AGI, though drawing connections between the theoretical and engineering aspects where this is useful to make the theory clearer. Even for those whose main interest is AGI engineering, AGI theory has multiple values: a good theory enables an engineer and empirical researcher to set objectives, to justify assumptions, to specify roadmaps and milestones, and to direct evaluation and comparison.

Some AGI research is founded on theoretical notions in an immediate and transparent way. Other AGI research is centered on system-building, with theory taking a back burner to building things and making them work. But every AGI project, no matter how pragmatic and empirical in nature, is ultimately based on some ideas about what intelligence is and how to realize it in artifacts. And it is valuable, as part of the process of shaping and growing an AGI project, that these ideas be clarified, justified, and organized into a coherent theory. Many times the theory associated with an AGI project is partially presented in a formal and symbolic form, to reduce the ambiguity and fuzziness in natural languages; but this is not necessarily the case, and purely verbal and conceptual theories may have value also. Some of the theories used in AGI research are inherited from other fields (such as mathematics, psychology, and computer science), and some others are specially invented for AGI. In cases where AGI theories are inherited from other fields, careful adaptations to the context of AGI are often required.

At the current stage, there is no single widely accepted theory of AGI, which is why this book uses a plural “foundations” in its title. For any AGI project, the underlying (explicit or implicit) theoretical foundation plays a crucial role, since any limitation or error in the theory will eventually show up in the project, and it is rarely possible to correct a *theoretical* mistake by a *technical* remedy. Comparing and evaluating the various competing and complementary theoretical foundations existing in the field is very important for AGI researchers, as well as for other interested individuals.

The existing AGI literature contains many discussions of AGI theory; but these are often highly technical, and they are often wrapped up together with highly specific discussions of system architecture or engineering, or particular application problems. We felt it

would be valuable – especially for readers who are not intimately familiar with the AGI field – to supplement this existing literature with a book providing a broad and relatively accessible perspective on the theoretical foundations of AGI. Rather than writing a volume on our own, and hence inevitably enforcing our own individual perspectives on the field, we decided to invite a group of respected AGI researchers to write about what they considered as among the most important theoretical issues of the field, in a language that is comprehensible to readers possessing at least modest scientific background, but not necessarily expertise in the AGI field. To our delight we received many valuable contributions, which are organized in the following chapters.

These chapters cover a wide spectrum of theoretical issues in AGI research. In the following overview they are clustered into three groups: the nature of the AGI problem and the objective of AGI research, AGI design methodology and system architecture, and the crucial challenges facing AI research.

1.3 The Matter of Objective

In the broadest sense, all works in AI and AGI aim at reproducing or exceeding the general intelligence displayed by the human mind in engineered systems. However, when describing this “intelligence” using more detailed and accurate words, different researchers effectively specify different objectives for their research [10]. Due to its stress on the general and holistic nature of intelligence, the AGI field is much less fragmented than the mainstream of AI [1], with many overarching aims and recurring themes binding different AGI research programs together. But even so, substantial differences in various researchers’ concrete research objectives can still be recognized.

The chapter by **Nick Cassimatis** provides a natural entry to the discussion. One key goal of AGI research, in many though not all AGI research paradigms, is to build computer models of human intelligence; and thus, in many respects, AGI is not all that different from what is called “cognitive modeling” in cognitive science. Cassimatis shows the need for an “intelligence science”, as well as carefully selected challenge problems that must be solved by modeling the right data.

The chapter by **Selmer Bringsjord and John Licato** addresses the question of how to define and measure artificial general intelligence, via proposing a “psychometric” paradigm in which AGI systems are evaluated using intelligence tests originally defined for humans. Since these tests have been defined to measure the “g factor”, which psychologists consider

a measure of human general intelligence, in a sense this automatically places a focus on general rather than specialized intelligence.

Though human-level intelligence is a critical milestone in the development of AGI, it may be that the most feasible route to get there is via climbing a “ladder of intelligence” involving explicitly nonhuman varieties of intelligence, as suggested in the chapter by **Sam Adams and Steve Burbeck**. The authors describe some interesting capabilities of octopi, comparing them to those of human beings, and argues more broadly that each of the rungs of the ladder of intelligence should be reached before trying a higher level.

The chapter by **Marcus Hutter** represents another alternative to the “human-level AGI” objective, though this time (crudely speaking) from above rather than below. Hutter’s Universal Artificial Intelligence is a formalization of “ideal rational behavior” that leads to optimum results in a certain type of environment. This project attempt “to capture the essence of intelligence”, rather than to duplicate the messy details of the human mind. Even though such an ideal design cannot be directly implemented, it can be approximated in various ways.

1.4 The Matter of Approach

Just as important as having a clear objective for one’s AGI research, is having a workable strategy and methodology for achieving one’s goals. Here the difference between AGI and mainstream AI shows clearly: while conventional AI projects focus on domain-specific and problem-specific solutions (sometimes with the hope that they will be somehow eventually connected together to get a whole intelligence), an AGI project often starts with a blueprint of a whole system, attempting to capture intelligence as a whole. Such a blueprint is often called an “architecture”.

The chapter by **Itamar Arel** proposes a very simple architecture, consisting of a perception module and an actuation module. After all, an AGI system should be able to take proper action in each perceived situation. Both modules use certain (different) types of learning algorithm, so that the system can learn to recognize patterns in various situations, as well as to acquire proper response to each situation. Unlike in mainstream AI, here the perception module and the actuation module are designed together; and the two are designed to work together in a manner driven by reinforcement learning.

Some other researchers feel the need to introduce more modules into their architectures, following results from psychology and other disciplines. The model introduced in

the chapter by **Usef Faghihi and Stan Franklin** turns certain existing theories about human cognition into a coherent design for a computer system, which has a list of desired properties. This architecture is more complicated than Arel's, which can be both an advantage and a disadvantage.

The chapter by **Ben Goertzel *et al.*** provides an integrative architecture diagram that summarizes several related cognitive architectures, and a defense of this approach to architectural and paradigmatic integration. It is argued that various AGI architectures, that seem different on the surface, are actually fundamentally conceptually compatible, and differ most dramatically in which parts of cognition they emphasize. Stress is laid on the hypothesis that the dynamics of an AGI system must possess "cognitive synergy", that is, multiple processes interacting in such a way as to actively aid each other when solving problems.

There are also researchers who do not want to design a fixed architecture for the system, but stress the importance of letting an AGI system construct and modify its architecture by itself. The chapter by **Kris Thórisson** advocates a "constructivist" approach to AI, which does not depend on human designed architectures and programs, but on self-organizing architectures and self-generated code that grow from proper "seeds" provided by the designer.

Just because a system has the ability for self-modification, does not necessarily mean that all the changes it makes will improve its performance. The chapter by **Bas Steunebrink and Jürgen Schmidhuber** introduces a formal model that reasons about its own programs, and only makes modifications that can be proved to be beneficial. Specified accurately in a symbolic language, this model is theoretically optimal under certain assumptions.

1.5 Challenges at the Heart of the Matter

Though AGI differs from mainstream AI in its holistic attitude toward intelligence, the design and development of an AGI system still needs to be carried out step by step, and some of the topics involved are considered to be more important and crucial than the others. Each chapter in this cluster addresses an issue that the author(s) takes to be one, though by no means the only one, major challenge in their research toward AGI.

The chapter by **Tsvi Achler** considers recognition as the foundation of other processes that altogether form intelligence. To meet the general-purpose requirements of AGI, a more flexible recognition mechanism is introduced. While the majority of current recognition al-

gorithms are based on a “feedforward” transformation from an input image to a recognized pattern, the mechanism Achler describes has a bidirectional “feedforward-feedback” structure, where the system’s expectation plays an important role.

Creativity is an aspect where computers are still far behind human intelligence. The chapter by **Maricarmen Martinez *et al.*** proposes analogy making and theory blending as ways to create new ideas. In this process, problem-solving theories are generalized from a source domain, then applied in a different target domain to solve novel problems. There is evidence showing that such processes indeed happen in human cognition, and are responsible for much of the creativity and generality of intelligence.

Contrary to many peoples’ assumption that “intelligence” is cold and unemotional, **Joscha Bach** argues that a model of intelligence must cover emotion and affect, since these play important roles in motivational dynamics and other processes. Emotion emerges via the system’s appraisal of situations and objects, with respect to the system’s needs and desires; and it in turn influences the system’s responses to those situations and objects, as well as its motivations and resource allocation.

Consciousness is one of the most mysterious phenomena associated with the human mind. The chapter by **Antonio Chella and Riccardo Manzotti** concludes that consciousness is necessary for general intelligence, and provides a survey of the existing attempts at producing similar phenomena in computer and robot systems. This study attempts to give some philosophical notions (including consciousness, free will, and experience) functional and constructive interpretations.

The difficulty of the problem of consciousness partly comes from the fact that it is not only a technical issue, but also a conceptual one. The chapter by **Richard Loosemore** provides a conceptual analysis of the notion of consciousness, helping us to understand what kind of answer might qualify as a solution to the problem. Such a meta-level reflection is necessary because if we get the problem wrong, there is little chance to get the solution right.

1.6 Summary

This book is not an attempt to settle all the fundamental problems of AGI, but merely to showcase and comprehensibly overview some of the key current theoretical explorations in the field. Given its stress on the generality and holistic nature of intelligence, AGI arguably has a greater demand for coherent theoretical foundations than narrow AI; and yet, the task

of formulating appropriate theories is harder for AGI than for narrow AI, due to the wider variety of interdependent factors involved.

The last chapter by **Pei Wang** is an attempt to provide common criteria for the analysis, comparison, and evaluation of the competing AGI theories. It is proposed that, due to the nature of the field, a proper theory of intelligence for AGI should be *correct* according to our knowledge about human intelligence, *concrete* on how to build intelligent machines, and *compact* in its theoretical structure and content. Furthermore, these criteria should be balanced against each other.

This collection of writings of representative, leading AGI researchers shows that there is still no field-wide consensus on the accurate specification of the objective and methodology of AGI research. Instead, the field is more or less held together by a shared attitude toward AI research, which treats the problem of AI as one problem, rather than as a group of loosely related problems, as in mainstream AI. Furthermore, AGI researchers believe that it is possible to directly attack the problem of general intelligence now, rather than to postpone it to a unspecified future time.

The problems discussed in this book are not the same as those addressed by the traditional AI literatures or in AI's various sibling disciplines. As we have argued previously [11], general-purpose AI has its own set of problems, which is neither a subset, nor a superset, of the problems studied in mainstream AI (the latter being exemplified in [8], e.g.). Among the problems of AGI, many are theoretical in nature, and must be solved by theoretical analysis – which in turn, must often be inspired and informed by experimental and engineering work. We hope this book will attract more attention, from both inside and outside the AGI field, toward the theoretical issues of the field, so as to accelerate the progress of AGI research – a matter which has tremendous importance, both intellectually and practically, to present-day human beings and our human and artificial successors.

Bibliography

- [1] Brachman, R. J. (2006). (AA)AI – more than the sum of its parts, 2005 AAAI Presidential Address, *AI Magazine* **27**, 4, pp. 19–34.
- [2] Feigenbaum, E. A. and Feldman, J. (1963). *Computers and Thought* (McGraw-Hill, New York).
- [3] Feigenbaum, E. A. and McCorduck, P. (1983). *The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the world* (Addison-Wesley Publishing Company, Reading, Massachusetts).
- [4] Goertzel, B. and Pennachin, C. (eds.) (2007). *Artificial General Intelligence* (Springer, New York).
- [5] McCarthy, J. (2007). From here to human-level AI, *Artificial Intelligence* **171**, pp. 1174–1182.

-
- [6] McCarthy, J., Minsky, M., Rochester, N. and Shannon, C. (1955). A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, URL: <http://www-formal.stanford.edu/jmc/history/dartmouth.html>.
 - [7] Newell, A. and Simon, H. A. (1963). GPS, a program that simulates human thought, in E. A. Feigenbaum and J. Feldman (eds.), *Computers and Thought* (McGraw-Hill, New York), pp. 279–293.
 - [8] Russell, S. and Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*, 3rd edn. (Prentice Hall, Upper Saddle River, New Jersey).
 - [9] Turing, A. M. (1950). Computing machinery and intelligence, *Mind* **LIX**, pp. 433–460.
 - [10] Wang, P. (2008). What do you mean by ‘AI’, in *Proceedings of the First Conference on Artificial General Intelligence*, pp. 362–373.
 - [11] Wang, P. and Goertzel, B. (2007). Introduction: Aspects of artificial general intelligence, in B. Goertzel and P. Wang (eds.), *Advance of Artificial General Intelligence* (IOS Press, Amsterdam), pp. 1–16.

Chapter 2

Artificial Intelligence and Cognitive Modeling Have the Same Problem

Nicholas L. Cassimatis

*Department of Cognitive Science, Rensselaer Polytechnic Institute, 108 Carnegie, 110 8th
St. Troy, NY 12180*

cassin@rpi.edu

Cognitive modelers attempting to explain human intelligence share a puzzle with artificial intelligence researchers aiming to create computers that exhibit human-level intelligence: how can a system composed of relatively unintelligent parts (such as neurons or transistors) behave intelligently? I argue that although cognitive science has made significant progress towards many of its goals, that solving the puzzle of intelligence requires special standards and methods in addition to those already employed in cognitive science. To promote such research, I suggest creating a subfield within cognitive science called *intelligence science* and propose some guidelines for research addressing the intelligence puzzle.

2.1 The Intelligence Problem

Cognitive scientists attempting to fully understand human cognition share a puzzle with artificial intelligence researchers aiming to create computers that exhibit human-level intelligence: how can a system composed of relatively unintelligent parts (say, neurons or transistors) behave intelligently?

2.1.1 *Naming the problem*

I will call the problem of understanding how unintelligent components can combine to generate human-level intelligence the *intelligence problem*; the endeavor to understand how the human brain embodies a solution to this problem *understanding human intelligence*; and the project of making computers with human-level intelligence *human-level artificial intelligence*.

When I say that a system exhibits human-level intelligence, I mean that it can deal with the same set of situations that a human can with the same level of competence. For example, I will say a system is a human-level conversationalist to the extent that it can have the same kinds of conversations as a typical human. A caveat to this is that artificial intelligence systems may not be able to perform in some situations, not for reasons of their programming, but because of issues related to their physical manifestation. For example, it would be difficult for a machine without hand gestures and facial expressions to converse as well as a human in many situations because hand gestures and facial expressions are so important to many conversations. In the long term, it may be necessary therefore to sort out exactly which situations matter and which do not. However, the current abilities of artificial systems are so far away from human-level that resolving this issue can generally be postponed for some time. One point that does follow from these reflections, though, is the inadequacy of the Turing Test. Just as the invention of the airplane was an advance in artificial flight without convincing a single person that it was a bird, it is often irrelevant whether a major step towards human-intelligence cons observers into believing a computer is a human.

2.1.2 Why the Intelligence Problem is Important

Why is the human-level intelligence problem important to cognitive science? The theoretical interest is that human intelligence poses a problem for a naturalistic worldview insofar as our best theories about the laws governing the behavior of the physical world posit processes that do not include creative problems solving, purposeful behavior and other features of human-level cognition. Therefore, not understanding how the relatively simple and “unintelligent” mechanisms of atoms and molecules combine to create intelligent behavior is a major challenge for a naturalistic world view (upon which much cognitive science is based). Perhaps it is the last major challenge. Surmounting the human-level intelligence problem also has enormous technological benefits which are obvious enough.

2.1.3 The State of the Science

For these reasons, understanding how the human brain embodies a solution to the human-level intelligence problem is an important goal of cognitive science. At least at first glance, we are quite far from achieving this goal. There are no cognitive models that can, for example, fully understand language or solve problems that are simple for a young child. This paper evaluates the promise of applying existing methods and standards in

cognitive science to solve this problem and ultimately proposes establishing a new subfield within cognitive science, called *Intelligence Science*¹, and outlines some guiding principles for that field.

Before discussing how effective the methods and standards of cognitive science are in solving the intelligence problem, it is helpful to list some of the problems or questions intelligence science must answer. The elements of this list are not original (see (Cassimatis, 2010) and (Shanahan, 1997)) or exhaustive. They are merely illustrative examples:

Qualification problem How does the mind retrieve or infer in so short a time the exceptions to its knowledge? For example, a hill symbol on a map means there is a hill in the corresponding location in the real world except if: the mapmaker was deceptive, the hill was leveled during real estate development after the map was made, or the map is of shifting sand dunes. Even the exceptions have exceptions. The sand dunes could be part of a historical site and be carefully preserved or the map could be based on constantly updated satellite images. In these exceptions to the exceptions, a hill symbol does mean there is a hill there now. It is impossible to have foreseen or been taught all these exceptions in advance, yet we recognize them as exceptions almost instantly.

Relevance problem Of the enormous amount of knowledge people have, how do they manage to retrieve the relevant aspects of it, often in less than a second, to sort from many of the possible interpretations of a verbal utterance or perceived set of events?

Integration problem How does the mind solve problems that require, say, probabilistic, memory-based and logical inferences when the best current models of each form of inference are based on such different computational methods?

Is it merely a matter of time before cognitive science as it is currently practiced answers questions like these or will it require new methods and standards to achieve the intelligence problem?

2.2 Existing Methods and Standards are not Sufficient

Historically, AI and cognitive science were driven in part by the goal of understanding and engineering human-level intelligence. There are many goals in cognitive science and, although momentous for several reasons, human-level intelligence is just one of them. Some other goals are to generate models or theories that predict and explain empirical data,

¹Obviously, for lack of a better name.

to develop formal theories to predict human grammatically judgments and to associate certain kinds of cognitive processes with brain regions. Methods used today in cognitive science are very successful at achieving these goals and show every indication of continuing to do so. In this paper, I argue that these methods are not adequate to the task of understanding human-level intelligence.

Put another way, it is possible to do good research by the current standards and goals of cognitive science and still not make much progress towards understanding human intelligence.

Just to underline the point, the goal of this paper is not to argue that “cognitive science is on the wrong track”, but that despite great overall success on many of its goals, progress towards one of its goals, understanding human-level intelligence, requires methodological innovation.

2.2.1 *Formal linguistics*

The goal of many formal grammarians is to create a formal theory that predicts whether a given set of sentences is judged by people to be grammatical or not. Within this framework, whether elements of the theory correspond to a mechanism humans use to understand language is generally not a major issue. For example, at various times during the development of Chomsky and his students’ formal syntax, their grammar generated enormous numbers of syntactic trees and relied on grammatical principles to rule out ungrammatical trees. These researchers never considered it very relevant to criticize their framework by arguing that it was implausible to suppose that humans could generate and sort through this many trees in the second or two it takes them to understand most sentences. That was the province of what they call “performance” (the mechanisms the mind uses) not competence (what the mind, in some sense, knows, independent of how it uses this knowledge). It is possible therefore to do great linguistics without addressing the computational problems (e.g. the relevance problem from the last section) involved in human-level language use.

2.2.2 *Neuroscience*

The field of neuroscience is so vast that it is difficult to even pretend to discuss it in total. I will confine my remarks to the two most relevant subfields of neuroscience. First, “cognitive neuroscience” is probably the subfield that most closely addresses mechanisms relevant to understanding human intelligence. What often counts as a result in this field is a demonstration that certain regions of the brain are active during certain forms of cognition.

A simplistic, but not wholly inaccurate way of describing how this methodology would apply to understanding intelligence would be to say that the field is more concerned with what parts of the brain embody a solution to the intelligence problem, not how they actually solve the problem. It is thus possible to be a highly successful cognitive neuroscientist without making progress towards solving the intelligence problem.

Computational neuroscience is concerned with explaining complex computation in terms of the interaction of less complex parts (i.e., neurons) obviously relevant to this discussion. Much of what I say about cognitive modeling below also applies to computational neuroscience.

2.2.3 *Artificial intelligence*

An important aim of this paper is that cognitive science's attempt to solve the intelligence problem is also an AI project and in later sections I will describe how this has and can still help cognitive science. There are, however, some ways AI practice can distract from that aim, too. Much AI research has been driven in part by at least one of these two goals.

(1) A formal or empirical demonstration that an algorithm is consistent with, approximates, or converges on some normative standard. Examples include proving that a Bayes network belief propagation algorithm converges on a probability distribution dictated by probability theory or proving that a theorem prover is sound and complete with respect to a semantics for some logic. Although there are many theoretical and practical reasons for seeking these results (I would like nuclear power plant software to be correct as much as anyone), they do not necessarily constitute progress towards solving the intelligence problem. For example, establishing that a Bayes Network belief propagation algorithm converges relatively quickly towards a normatively correct probability distribution given observed states of the world does not in any way indicate that solving such problems is part of human-level intelligence, nor is there any professional incentive or standard requiring researchers to argue for this. There is in fact extensive evidence that humans are not normatively correct reasoners. It may even be that some flaws in human reasoning are a tradeoff required of any computational system that solves the problems humans do.

(2) Demonstrating with respect to some metric that an algorithm or system is faster, consumes fewer resources and/or is more accurate than some alternative(s). As with proving theorems, one can derive great professional mileage creating a more accurate part of speech

tagger or faster STRIPS planner without needing to demonstrate in any way that their solution is consistent with or contributes to the goal of achieving human-level intelligence.

2.2.4 *Experimental psychology*

Cognitive psychologists generally develop theories about how some cognitive process operates and run experiments to confirm these theories. There is nothing specifically in this methodology that focuses the field on solving the intelligence problem. The field's standards mainly regard the accuracy and precision of theories, not the level of intelligence they help explain. A set of experiments discovering and explaining a surprising new phenomenon in (mammalian-level) place memory in humans will typically receive more plaudits than another humdrum experiment in high-level human reasoning. To the extent that the goal of the field is solely to find accurate theories of cognitive processes, this makes sense. But it also illustrates the lack of an impetuous towards understanding human-level intelligence. In addition to this point, many of Newell's (Newell, 1973) themes apply to the project of understanding human-level intelligence with experimental psychology alone and will not be repeated here.

A subfield of cognitive psychology, cognitive modeling, does, at its best, avoid many of the mistakes Newell cautions against and I believe understanding human cognition is ultimately a cognitive modeling problem. I will therefore address cognitive modeling extensively in the rest of this paper.

2.3 Cognitive Modeling: The Model Fit Imperative

Cognitive modeling is indispensable to the project of understanding human-level intelligence. Ultimately, you cannot say for sure that you have understood how the human brain embodies a solution to the intelligence problem unless you have (1) a computational model that behaves as intelligently as a human and (2) some way of knowing that the mechanisms of that model, or at least its behavior, reflect what is going on in humans. Creating computer models to behave like humans and showing that the model's mechanisms at some level correspond to mechanism underlying human cognition is a big part of what most cognitive modelers aim to do today. Understanding how the human brain embodies a solution to the intelligence problem is thus in part a cognitive modeling problem.

This section describes why I think some of the practices and standards of the cognitive modeling community, while being well-suited for understanding many aspects of cognition,

are not sufficient to, and sometimes even impede progress towards, understanding human-level intelligence.

The main approach to modeling today is to create a model of human cognition in a task that fits existing data regarding their behavior in that task and, ideally, predicts behavior in other versions of the task or other tasks altogether. When a single model with a few parameters predicts behavior in many variations of a task or in many different tasks, that is good evidence that the mechanisms posited by the model correspond, at least approximately, to actual mechanisms of human cognition. I will call the drive to do this kind of work the *model fit imperative*.

What this approach does not guarantee is that the mechanisms uncovered are important to understanding human-level intelligence. Nor does it do impel researchers to find important problems or mechanisms that have not yet been addressed, but which are key to understanding human-level intelligence.

An analogy with understanding and synthesizing flight will illustrate these points². Let us call the project of understanding birds *aviary science*; the project of creating computational models of birds *aviary modeling* and the project of making machines that fly *artificial flight*. We call the problem of how a system that is composed of parts that individually succumb to gravity can combine to defy gravity the *flight problem*; and we call the project of understanding how birds embody a solution to this problem *understanding bird flight*.

You can clearly do great aviary science, i.e., work that advances the understanding of birds, without addressing the flight problem. You can create predictive models of bird mating patterns that can tell you something about how birds are constructed, but they will tell you nothing about how birds manage to fly. You can create models that predict the flapping rate of a bird's wings and how that varies with the bird's velocity, its mass, etc. While this work studies something related to bird flight, it does not give you any idea of how birds actually manage to fly. Thus, just because aviary science and aviary modeling are good at understanding many aspects of birds, it does not mean they are anywhere near understanding bird flight. If the only standard of their field is to develop predictive models of bird behavior, they can operate with great success without ever understanding how birds solve the flight problem and manage to fly.

I suggest that the model fit imperative in cognitive modeling alone is about as likely to lead to an understanding of human intelligence as it would be likely to drive aviary science towards understanding how birds fly. It is possible to collect data about human cognition,

²I have been told that David Marr has also made an analogy between cognitive science and aeronautics, but I have been unable to find the reference.

build fine models that fit the data and accurately predict new observations – it is possible to do all this without actually helping to understand human intelligence. Two examples of what I consider the best cognitive modeling I know of illustrate this point. (Lewis & Vasishth, 2005) have developed a great model of some mechanisms involved in sentence understanding, but this and a dozen more fine pieces of cognitive modeling could be done and we would still not have a much better idea of how people actually manage to solve all of the inferential problems in having a conversation, how they sort from among all the various interpretations of a sentence, how they manage to fill in information not literally appearing in a sentence to understand the speaker’s intent. Likewise, Anderson’s (Anderson, 2005) work modeling brain activity during algebraic problem solving is a big advance in confirming that specific mechanisms in ACT-R models of cognition actually reflect real, identifiable, brain mechanisms. But, as Anderson himself claimed³, these models only shed light on behavior where there is a preordained set of steps to take, not where people actually have to intelligently figure out a solution to the problem on their own.

The point of these examples is not that they are failures. These projects are great successes. They actually achieved the goals of the researchers involved and the cognitive modeling community. That they did so without greatly advancing the project of understanding human intelligence is the point. The model fit imperative is geared towards understanding cognition, but not specifically towards making sure that human-level intelligence is part of the cognition we understand. To put the matter more concretely, there is nothing about the model fit imperative that forces, say, someone making a cognitive model of memory to figure out how their model explains how humans solve the qualification and relevance problems. When one’s goal is to confirm that a model of a cognitive process actually reflects how the mind implements that process, the model fit imperative can be very useful. When one has the additional goal of explaining human-level intelligence, then some additional standard is necessary to show that this model is powerful enough to explain human-level performance.

Further, I suggest that the model fit imperative can actually impeded progress towards understanding human intelligence. Extending the analogy with the flight problem will help illustrate this point. Let us say the Wright Brothers decided for whatever reason to subject themselves to the standards of our hypothetical aviary modeling community. Their initial plane at Kitty Hawk was not based on detailed data on bird flight and made no predictions about it. Not only could their plane not predict bird wing flapping frequencies, its wings

³In a talk at RPI.

did not flap at all. Thus, while perhaps a technological marvel, their plane was not much of an achievement by the aviary modeling community's model fit imperative. If they and the rest of that community had instead decided to measure bird wing flapping rates and create a plane whose wings flapped, they may have gone through a multi-decade diversion into understanding all the factors that contribute to wing flapping rates (not to mention the engineering challenge of making plane whose wings flaps) before they got back to the nub of the problem, to discover the aerodynamic principles and control structures that can enable flight and thereby solve the flight problem. The Wright Flyer demonstrated that these principles were enough to generate flight. Without it, we would not be confident that what we know about bird flight is enough to fully explain how they fly. Thus, by adhering to the model fit imperative, aviary science would have taken a lot longer to solve the flight problem in birds.

I suggest that, just as it would in aviary science, the model fit imperative can retard progress towards understanding how the human brain embodies a solution to the intelligence problem. There are several reasons for this, which an example will illustrate. Imagine that someone has created a system that was able to have productive conversations about, say, managing one's schedule. The system incorporates new information and answer questions as good as a human assistant can. When it is uncertain about a statement or question it can engage in a dialog to correct the situation. Such a system would be a tremendous advance in solving the intelligence problem. The researchers who designed it would have had to find a way, which has so far eluded cognitive science and AI researchers, to integrate multiple forms of information (acoustic, syntactic, semantic, social, etc.) within milliseconds to sort through the many ambiguous and incomplete utterance people make. Of the millions of pieces of knowledge about this task, about the conversants and about whatever the conversants could refer to, the system must find just the right knowledge, again, within a fraction of a second. No AI researchers have to this point been able to solve these problems. Cognitive scientists have not determined how people solve these problems in actual conversation. Thus, this work is very likely to contain some new, very powerful ideas that would help AI and cognitive science greatly.

Would we seriously tell these researchers that their work is not progress towards understanding the mind because their system's reaction times or error rates (for example) do not quite match up with those of people in such conversations? If so, and these researchers for some reason wanted our approval, what would it have meant for their research? Would they have for each component of their model run experiments to collect data about that

component and calibrate the component to that data? What if their system had dozens of components, would they have had to spend years running these studies? If so, how would they have had the confidence that the set of components they were studying was important to human-level conversation and that they were not leaving out components whose importance they did not initially anticipate? Thus, the data fit model of research would either have forced these researchers to go down a long experimental path that they had little confidence would address the right issues or they would have had to postpone announcing, getting credit for and disseminating to the community the ideas underlying their system.

For all these reasons, I conclude that the model fit imperative in cognitive modeling does not adequately drive the field towards achieving an understanding of human intelligence and that it can even potentially impede progress towards that goal.

Does all this mean that cognitive science is somehow exceptional, that in every other part of science, the notion of creating a model, fitting it to known data and accurately predicting new observations does not apply to understanding human-level intelligence? Not at all. There are different levels of detail and granularity in data. Most cognitive modeling involves tasks where there is more than one possible computer program known that can perform in that task. For example, the problem of solving algebraic equations can be achieved by many kinds of computer programs (e.g., Mathematica and production systems). The task in that community is to see which program the brain uses and to select a program that exhibits the same reaction times and error rates as humans is a good way to go about this. However, in the case of human-level intelligence, *there are no known programs that exhibit human-level intelligence*. Thus, before we can get to the level of detail of traditional cognitive modeling, that is, before we can worry about fitting data at the reaction time and error rate level of detail, we need to explain and predict the most fundamental datum: people are intelligent. Once we have a model that explains this, we can fit the next level of detail and know that the mechanisms whose existence we are confirming are powerful enough to explain human intelligence.

Creating models that predict that people are intelligent means writing computer programs that behave intelligently. This is also a goal of artificial intelligence. Understanding human intelligence is therefore a kind of AI problem.

2.4 Artificial Intelligence and Cognitive Modeling Can Help Each Other

I have so far argued that existing standards and practices in the cognitive sciences do not adequately drive the field towards understanding human intelligence. The main problems are that (1) each field's standards make it possible to reward work that is not highly relevant to understanding human intelligence; (2) there is nothing in these standards to encourage researchers to discover each field's gaps in its explanation of human intelligence; and (3) that these standards can actually make it difficult for significant advances towards understanding human-intelligence to gain support and recognition. This section suggests some guidelines for cognitive science research into human intelligence.

Understanding human-intelligence should be its own subfield Research towards understanding human intelligence needs to be its own subfield, *intelligence science*, within cognitive science. It needs its own scientific standards and funding mechanisms. This is not to say that the other cognitive sciences are not important for understanding human intelligence; they are in fact indispensable. However, it will always be easier to prove theorems, fit reaction time data, refine formal grammars or measure brain activity if solving the intelligence problem is not a major concern. Researchers in an environment where those are the principal standards will always be at a disadvantage professionally if they are also trying to solve the intelligence problem. Unless there is a field that specifically demands and rewards research that makes progress towards understanding how the brain solves the intelligence problem, it will normally be, at least from a professional point of view, more prudent to tackle another problem. Just as it is impossible to seriously propose a comprehensive grammatically theory without addressing verb use, we need a field where it is impossible to propose a comprehensive theory of cognition or cognitive architecture without at least addressing the qualification, relevance, integration and other problems of human-level intelligence.

Model the right data I argued earlier and elsewhere (Cassimatis *et al.*, 2008) that the most important datum for intelligence scientists to model is that humans are intelligent. With respect to the human-level intelligence problem, for example, to worry about whether, say, language learning follows a power or logarithmic law before actually discovering how the learning is even possible is akin to trying to model bird flap frequency before understanding how wings contribute to flight.

The goal of building a model that behaves intelligently, instead of merely modeling mechanisms such as memory and attention implicated in intelligent cognition, assures that

the field addresses the hard problems involved in solving the intelligence problem. It is hard to avoid a hard problem or ignore an important mechanisms if, say, it is critical to human-level physical cognition and building a system that makes the same physical inferences that humans can is key to being published or getting a grant renewed.

A significant part of motivating and evaluating a research project in intelligence science should be its relevance for (making progress towards) answering problems such as the qualification, relevance and integration problems.

Take AI Seriously Since there are zero candidate cognitive models that exhibit human-level intelligence, researchers in intelligence science are in the same position as AI researchers aiming for human-level AI: they are both in need of and searching for computational mechanisms that exhibit a human-level of intelligence. Further, the history of AI confirms its relevance to cognitive science. Before AI many philosophers and psychologists did not trust themselves or their colleagues to posit internal mental representations without implicitly smuggling in some form of mysticism or homunculus. On a technical level, search, neural networks, Bayesian networks, production rules, etc. were all in part ideas developed by AI researchers but which play an important role in cognitive modeling today.

Chess-playing programs are often used as examples of how AI can succeed with brute-force methods that do not illuminate human intelligence. Note, however, that chess programs are very narrow in their functionality. They only play chess. Humans can play many forms of games and can learn to play these rather quickly. Humans can draw on skills in playing one game to play another. If the next goal after making computer programs chess masters was not to make them grandmasters, but to make them learn, play new games and transfer their knowledge to other games, brute force methods would not have been sufficient and researchers would have had to develop new ideas, many of which would probably bear on human-level intelligence.

Have a success Many AI researchers have retreated from trying to achieve human-level AI. The lesson many have taken from this is that one should work on more tractable problems or more practical applications. This attitude is tantamount to surrendering the goal of solving the human intelligence problem in our lifetimes. The field needs a success to show that real progress is capable soon. One obstacle to such a success is that the bar, especially in AI, has been raised so high that anything short of an outright demonstration of full human-level AI is considered by many to be hype. For a merely very important advance

towards human-level intelligence that has no immediate application, there is no good way to undeniably confirm that importance. We thus need metrics that push the state of the art but are at the same time realistic.

Develop realistic metrics Developing realistic methods for measuring a system's intelligence would make it possible to confirm that the ideas underlying it are an important part of solving the intelligence problem. Such metrics would also increase confidence in the prospects of intelligence science enabling quicker demonstrations of progress. My work on a model of physical cognition has illustrated the value of such metrics (Cassimatis, in press). I have so far tested this model by presenting it with sequences of partially occluded physical events that I have partly borrowed from the developmental psychology literature and have partly crafted myself. My strategy has been to continually find new classes of scenarios that require different forms of reasoning (e.g., probabilistic, logical, defeasible, etc.) and update my model so that it could reason about each class of scenarios. Using superficially simple physical reasoning problems in this way has had several properties that illustrate the value of the right metric.

Difficulty Challenge problems should be difficult enough so that a solution to them requires a significant advance in the level of intelligence it is possible to model. Human-level intelligence in the physical cognition domain requires advances towards understanding the frame problem, defeasible reasoning and how to integrate perpetual and cognitive models based on very different algorithms and data structures.

Ease While being difficult enough to require a real advance, challenge problem should be as simple as possible so that real progress is made while avoiding extraneous issues and tasks. One benefit of the physical cognition domain over, for example, Middle East politics is the smaller amount of knowledge required for a system to have before it can actually demonstrate intelligent reasoning.

Incremental It should be possible to demonstrate advances towards the goal short of actually achieving it. For example, it is possible to show progress in the physical cognition domain without actually providing a complete solution by showing that an addition to the model enables and explains reasoning in a significantly wider, but still not complete, set of scenarios.

General The extent to which a challenge problem involves issues that underlie cognition in many domains makes progress towards solving that problem more important. For exam-

ple, I have shown (Cassimatis, 2004) how syntactic parsing can be mapped onto a physical reasoning problem. Thus, progress towards understanding physical cognition amounts to progress in two domains.

2.5 Conclusions

I have argued that cognitive scientists attempting to understand human intelligence can be impeded by the standards of the cognitive sciences, that understanding human intelligence will require its own subfield, intelligence science, and that much of the work in this subfield will assume many of the characteristics of good human-level AI research. I have outlined some principles for guiding intelligence science that I suggest would support and motivate work towards solving the intelligence problem and understanding how the human brain embodies a solution to the intelligence problem.

In only half a century we have made great progress towards understanding intelligence within fields that, with occasional exceptions, have not been specifically and wholly directed towards solving the intelligence problem. We have yet to see the progress that can happen when large numbers of individuals and institutions make this their overriding goal.

Bibliography

- [1] Anderson, J.R. (2005). Human symbol manipulation within an integrated cognitive architecture. *Cognitive Science*, 313–341.
- [2] Cassimatis, N.L. (2004). *Grammatical Processing Using the Mechanisms of Physical Inferences*. Paper presented at the Twentieth-Sixth Annual Conference of the Cognitive Science Society.
- [3] Cassimatis, N.L., & Bignoli, P. (in press). Testing Common Sense Reasoning Abilities. *Journal of Theoretical and Experimental Artificial Intelligence*.
- [4] Cassimatis, N.L., Bello, P., & Langley, P. (2008). Ability, Parsimony and Breadth in Models of Higher-Order Cognition. *Cognitive Science*, **33** (8), 1304–1322.
- [5] Cassimatis, N., Bignoli, P., Bugajska, M., Dugas, S., Kurup, U., Murugesan, A., & Bello, P. (2010). An Architecture for Adaptive Algorithmic Hybrids. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, **4** (3), 903–914.
- [6] Lewis, R.L., & Vasishth, S. (2005). An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, 29, 375–419.
- [7] Newell, A. (1973). You can't play 20 questions with nature and win. In W.G. Chase (Ed.), *Visual Information Processing*, Academic Press.
- [8] Shanahan, M. (1997). Solving the frame problem: a mathematical investigation of the common sense law of inertia. MIT Press. Cambridge, MA.

Chapter 3

Psychometric Artificial General Intelligence: The Piaget-MacGuyver Room *

Selmer Bringsjord and John Licato

Department of Computer Science
Department of Cognitive Science
Lally School of Management & Technology
Rensselaer Polytechnic Institute (RPI)
Troy NY 12180 USA

Psychometric AGI (PAGI) is the brand of AGI that anchors AGI science and engineering to explicit tests, by insisting that for an information-processing (i-p) artifact to be rationally judged generally intelligent, creative, wise, and so on, it must pass a suitable, well-defined test of such mental power(s). Under the tent of PAGI, and inspired by prior thinkers, we introduce the *Piaget-MacGuyver Room* (PMR), which is such that, an i-p artifact can credibly be classified as general-intelligent if and only if it can succeed on *any* test constructed from the ingredients in this room. No advance notice is given to the engineers of the artifact in question, as to what the test is going to be; only the ingredients in the room are shared ahead of time. These ingredients are roughly equivalent to what would be fair game in the testing of neurobiologically normal Occidental students to see what stage within Piaget’s theory of cognitive development they are at. Our proposal and analysis puts special emphasis on a kind of cognition that marks Piaget’s Stage IV and beyond: viz., the intersection of hypothetico-deduction and analogical reasoning, which we call *analogico-deduction*.

3.1 Introduction

Psychometric AGI (PAGI; pronounced “pay guy”), in a nutshell, is the brand of AGI that anchors AGI science and engineering to explicit tests, by insisting that for an information-processing¹ (i-p) artifact to be rationally judged generally intelligent, creative,

*We are greatly indebted to not only the editors, but to two anonymous referees for invaluable feedback on earlier drafts of our paper.

¹By using ‘information-processing’ rather than ‘computational’ we leave completely open the level of information-processing power — from that of a standard Turing machine, to so-called “hypercomputers” — the artifact in question has. Note that we also for the most part steer clear of the term ‘agent,’ which is customary in

wise, and so on, the artifact must be capable of passing a suitable, well-defined test of such mental power(s), even when it hasn't seen the test before. (PAGI is built upon PAI, psychometric AI; see Bringsjord and Schimanski, 2003.) For example, someone might claim that IBM's i-p artifact Deep Blue is really and truly intelligent, in light of the fact that if you test it by seeing whether it can prevail against the best human chessplayers, you will find that it can. And someone might claim that natural-language-processing artifact Watson, another i-p artifact from IBM (Ferrucci *et al.*, 2010), is really and truly intelligent because it can vanquish human opponents in the game of *Jeopardy!*. However, while both of these artifacts are intelligent *simpliciter*, they most certainly aren't *general*-intelligent. Both Deep Blue and Watson were explicitly engineered to specifically play chess and *Jeopardy!*, nothing more; and in both cases the artifacts knew what their final tests would be.

Inspired by PAGI, and by a line of three thinkers (Descartes, Newell, and esp. Piaget) who gave much thought to the hallmarks of *general* intelligence, we define a room, the *Piaget-MacGyver Room* (PMR), which is such that, an i-p artifact can credibly be classified as general-intelligent if and only if it can succeed on *any* test constructed from the ingredients in this room. *No advance notice is given to the engineers of the artifact in question as to what the test is going to be.* This makes for rather a different situation than that seen in the case of both Deep Blue and Watson; for in both of these cases, again, the AI engineering that produced these i-p artifacts was guided by a thorough understanding and analysis, ahead of time, of the tests in question. In fact, in both cases, again, all along, the engineering was guided by repeatedly issuing pre-tests to both artifacts, and measuring their performance with an eye to making incremental improvements. This is particularly clear in the case of Watson; see (Ferrucci *et al.*, 2010). Of course, we happily concede that both Deep Blue and Watson *are* intelligent; we just don't believe that either is *general*-intelligent.²

As we say, only the *ingredients* in PMR are shared ahead of time with the relevant engineers. These ingredients are equivalent to what would be fair game in the testing, by Piaget, of a neurobiologically normal Occidental student who has reached at least Piaget's Stage III of cognitive development. If you will, Piaget is in control of the ingredients in the AI. We do so because 'agent' is usually taken to imply a function that is Turing-computable or easier; e.g., see the use of 'agent' in (Russell and Norvig, 2002).

²Our attitude is anticipated e.g. by Penrose, who for instance pointed out that Deep Blue would be paralyzed if challenged on the spot to play variants of chess; see (Penrose, 1994). In the case of Watson, questions based on neologisms would paralyze the system. E.g., "Supposing that bloogering! is to take a prime and bloogier it (add it to itself), and then bloogier thrice more times, what is bloogering! 7?"

room, and, with a general understanding of the MacGuyver television series,³ and of course with an understanding of his own account of cognitive development, assembles from the ingredients in the room a test of an artificial agent that is purportedly general-intelligent. For example, Piaget might be “armed” with the ingredients shown in Figure 3.1.

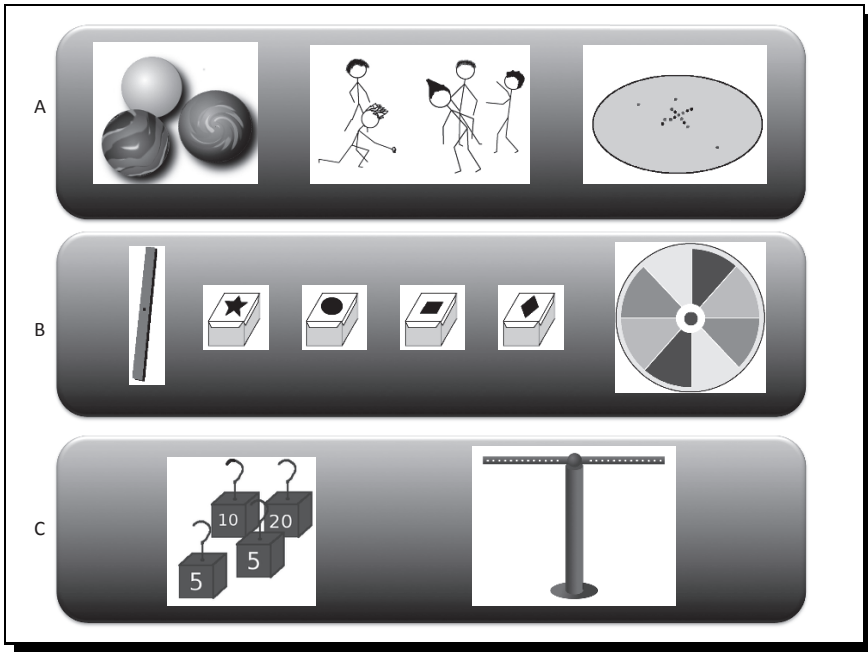


Fig. 3.1 A Possible Set of Ingredients From Which Piaget Can Work (weights, marbles (and its playing field), human confederates, familiar shapes, magnets, etc.)

If the artifact passes what Piaget assembles, we can safely say that it’s indeed general-intelligent; if it fails, we declare that it isn’t. We shall allow a range of responses that fall into these two categories, since some with the general intelligence of a Feynman, after being given the test, might well be able to find an abundance of solutions.⁴ As will be seen below, our proposal and analysis puts special emphasis on cognition that marks (Piagetian) Stage IV and beyond: viz., the intersection of hypothetico-deduction and analogical reasoning (which we call *analogico-deduction*). In hypothetico-deduction one creates hy-

³For a description of the series, see: <http://en.wikipedia.org/wiki/MacGuyver>. The hero and protagonist, MacGuyver, is stunningly resourceful, and hence hard to trap, seriously injure, or kill; he always manages to carry out some physical manipulation that solves the problem at hand. In the harder of Piaget’s tests, a high degree of resourcefulness is a *sine qua non*; and the tests invariably call à la MacGuyver for *physical manipulation* that confirms the resourcefulness.

⁴See <http://blogs.msdn.com/b/ericlippert/archive/2011/02/14/what-would-feynman-do.aspx>.

potheses h_1, h_2, \dots, h_n , conditionals of the form $h_i \rightarrow r_i$, and then tests to see whether the results r_i do indeed obtain, following upon an instantiation of h_i . If r_i doesn't obtain, *modus tollens* immediately implies that h_i is to be rejected. When analogical reasoning undergirds either the generation of the hypotheses or the conditionals, or the negation of a result r_i , the overall process falls under analogico-deduction. In order to focus matters we shall restrict our attention to not only such reasoning, but to such reasoning applied to a representative test fashioned by Piaget: his ingenious magnet test.

The plan of the chapter is as follows. We first (§ 3.2) explain in a bit more detail what PABI is, and why the roots of this brand of AGI are to be explicitly found in the thinking of Newell, and before him, in two tests described by Descartes. We then (§ 3.3) look at these two tests in a bit more detail. Next, in section 3.4, we give a barbarically quick overview of Piaget's view of thinking, and present his magnet challenge. We then (§ 3.5) briefly describe the LISA system for modeling analogical reasoning. Following this, we model and simulate, using the linked information-processing system Slate+LISA (Slate is an argument-engineering environment that can be used in the purely deductive mode for *proof* engineering), human problem-solving cognition in the magnet challenge (§ 3.6). A brief pointer to the next research steps in the PABI research program in connection with PMR (which, ultimately, we have every intention of building and furnishing), wraps up the chapter.

3.2 More on Psychometric AGI

Rather long ago, Newell (1973) wrote a prophetic paper: "You Can't Play 20 Questions with Nature and Win." This paper helped catalyze both modern-day computational cognitive modeling through cognitive architectures (such as ACT-R, NARS, Soar, Polyscheme, etc.), and AI's — now realized, of course — attempt to build a chess-playing machine better at the game than any human. However, not many know that in this paper Newell suggested a *third* avenue for achieving general machine intelligence, one closely aligned with psychometrics, and one — as we shall see — closely aligned as well with the way Piaget uncovered the nature of human intelligence. In the early days of AI, at least one thinker started decisively down this road for a time (Evans 1968); but now the approach, it may be fair to say, is not all that prominent in AI. We refer to this approach as *Psychometric AGI*, or just PABI (rhymes with "pay guy").

3.2.1 *Newell & the Neglected Route Toward General Machine Intelligence*

In the “20 Questions” paper, Newell bemoans the fact that, at a symposium gathering together many of the greatest psychologists at the time, there is nothing whatsoever to indicate that any of their work is an organized, integrated program aimed seriously at uncovering the nature of intelligence as information processing. Instead, Newell perceives a situation in which everybody is carrying out work (of the highest quality, he cheerfully admits) on his or her own specific little part of human cognition. In short, there is nothing that, to use Newell’s phrase, “pulls it all together.” He says: “We never seem in the experimental literature to put the results of all the experiments together.” (1973: 298) After making clear that he presupposes that “man is an information processor,” and that therefore from his perspective the attempt to understand, simulate, and replicate human intelligence is by definition to grapple with the challenge of creating machine intelligence, Newell offers three possibilities for addressing the fragmentary nature of the study of mind as computer.

The first possibility Newell calls “Complete Processing Models.” He cites his own work (with others; e.g., Simon; the two, of course, were to be a dynamic duo in AI for many decades to come) based on production systems, but makes it clear that the production-system approach isn’t the only way to go. Of course today’s cognitive architectures [e.g., NARS (Wang, 2006); SOAR (Rosenbloom, Laird and Newell, 1993); ACT-R (Anderson, 1993; Anderson and Lebiere, 1998; Anderson and Lebiere, 2003); Clarion (Sun, 2001); and Polyscheme (Cassimatis, 2002; Cassimatis, Trafton, Schultz and Bugajska, 2004)] can be traced back to this first possibility.

The second possibility is to “Analyze a Complex Task.” Newell sums this possibility up as follows.

A second experimental strategy, or paradigm, to help overcome the difficulties enumerated earlier is to accept a single complex task and do all of it . . . the aim being to demonstrate that one has a significant theory of a genuine slab of human behavior. . . . A final example [of such an approach] would be to take chess as the target super-task (Newell 1973: 303–304).

This second possibility is one most people in computational cognitive science and AI are familiar with. Though Deep Blue’s reliance upon standard search techniques having little cognitive plausibility perhaps signaled the death of the second avenue, there is no question that, at least for a period of time, many researchers were going down it.

The third possibility, “One Program for Many Tasks,” is the one many people seem to have either largely forgotten or ignored. Newell described it this way:

The third alternative paradigm I have in mind is to stay with the diverse collection of small experimental tasks, as now, but to construct a single system to perform them all. This single

system (this model of the human information processor) would have to take the instructions for each, as well as carry out the task. For it must truly be a single system in order to provide the integration we seek (Newell 1973: 305).

For those favorably inclined toward the test-based approach to AI or AGI, it's the specific mold within Newell's third possibility that is of acute interest. We read:

A ... mold for such a task is to construct a single program that would take a standard intelligence test, say the WAIS or the Stanford-Binet. (Newell 1973: 305)

We view this remark as a pointer to PAGI, and to a brief explication of this brand of AGI we now turn.

3.2.2 *So, What is Psychometric AGI?*

What is AI? We'd be willing to wager that many of you have been asked this question — by colleagues, reporters, friends and family, and others. Even if by some fluke you've dodged the question, perhaps you've asked it yourself, maybe even perhaps (in secret moments, if you're a practitioner) *to* yourself, without an immediate answer coming to mind. At any rate, AI *itself* repeatedly asks the question — as the first chapter of many AI textbooks reveals. Unfortunately, many of the answers standardly given don't ensure that AI tackles head on the problem of *general* intelligence (whether human or machine). For instance, Russell and Norvig (2002) characterize AI in a way (via functions from percepts to actions; they call these functions *intelligent agents*) that, despite its many virtues, doesn't logically entail any notion of generality whatsoever: An agent consisting solely in the factorial function qualifies as an intelligent agent on the R-N scheme. Our answer, however, is one in line with Newell's third possibility, and one in line with a perfectly straightforward response to the "What is AI?" question.

To move toward our answer, note first that presumably the 'A' part of 'AGI' isn't the challenge: We seem to have a fairly good handle on what it means to say that something is an artifact, or artificial. It's the 'G' and the 'I' parts that seem to throw us for a bit of a loop. First, what's intelligence? *This* is the first of the two big, and hard, questions. Innumerable answers have been given, but many outside the test-based approach to AI seem to forget that there is a particularly clear and straightforward answer available, courtesy of the field that has long sought to operationalize the concept in question; that field is psychometrics. Psychometrics is devoted to systematically measuring psychological properties, usually via tests. These properties include the ones most important in the present context: both intelligence, and general intelligence. In a nutshell, the initial version of a psychometrics-

oriented account of general intelligence (and this definition marks an answer to the second big question: What's *general* intelligence?) is simply this: Some i-p artifact is intelligent if and only if it can excel at *all* established, validated tests of neurobiologically normal cognition, even when these tests are new for the artifact.

Of course, psychometrics is by definition devoted to specifically defining and measuring *human* intelligence; the SAT, part of the very fabric of education in the United States (with its counterparts in widespread use in other technologized countries), is for example administered to humans, not machines. Some hold that it's neither possible nor necessary for AI to be identical to human intelligence. After all, it seems possible for an AI to have a vision system that covers a different frequency range of light, compared to of a normal human. Consequently, such a system may fail some tests that require color recognition. In this context, someone might object: "What makes the two of you think, then, that intelligence tests can be sensibly applied to i-p artifacts?"

Full analysis and rebuttal of this worry would occupy more space than we have; we must rest content with a brief response, via three points:

- (1) Judgments regarding whether i-p artifacts are intelligent are already informally, but firmly, rooted in the application of tests from the human sphere. We know that Kasparov is quite an intelligent chap; and we learned that Deep Blue, accordingly, is intelligent; a parallel moral emerged from the victory of Watson. We are simply, at bottom, extending and rigorizing this already-established human-centric way of gauging machine intelligence.
- (2) The field of psychometrics is in reality constrained by the need for *construct validity*, but in PABI this constraint is cheerfully defenestrated. Tests that are construct-valid are such that, when successfully taken, they ensure that the relevant underlying structures and processes have been active "inside" the agent in question. But in PABI, the bottom line is "getting the job done," and in fact we assume that i-p machines will, "under the hood," depart from human techniques.
- (3) The third point in our answer flows from the second, and is simply a reminder that while in the human sphere the scoring of tests of mental ability is indeed constrained by comparison to other human test-takers (an IQ "score," after all, is meaningless without relative comparison to other humans who take the relevant test), PABI is founded upon a much more coarse-grained view of intelligence tests — a view according to which, for instance, a perfect score on the part of an i-p artifact indicates that it's intelligent *simpliciter*, not that it's intelligent within some human-centric continuum. This general point applies directly to PMR: For example, prowess in PMR specifically requires sensorimotor prowess, but not *human* sensorimotor adroitness. We assume only that one side of general intelligence, as that concept covers both human and i-p machine, is perceiving and moving, in playful ways, physical objects.

We anticipate that some will insist that while intelligence tests are sensibly applicable to i-p artifacts in principle, the fact remains that even broad intelligence tests are still just too narrow, when put in the context of the full array of cognitive capacities seen in *homo sapiens*. But one can understand general intelligence, from the standpoint of psychometrics, to

include many varied, indeed for that matter all, tests of intellectual ability. Accordingly, one can work on the basis of a less naïve definition of PEGI, which follows.⁵

Psychometric AGI is the field devoted to building i-p artifacts capable of at least solid performance on all established, validated tests of intelligence and mental ability, without having seen these tests beforehand at all; the class of tests in play here includes not just the rather restrictive IQ tests, but also tests of the many different forms of intelligence seen in the human sphere.⁶

This definition, when referring to tests of mental ability, is pointing to much more than IQ tests. For example, following Sternberg (1988), someone with much musical aptitude would count as brilliant even if their scores on tests of “academic” aptitude (e.g., on the SAT, GRE, LSAT, etc.) were low. Nonetheless, even if, hypothetically, one were to restrict attention in PEGI to intelligence tests, a large part of cognition would be targeted. Along this line, in choosing the WAIS, Newell knew what he was doing.

To see this, we begin by going back to the early days of AI, specifically to a time when Psychometric AI was at least implicitly entertained. For example, in the mid 1960s, the largest Lisp program on earth was Evans’ (1968) ANALOGY program, which could solve problems like those shown in Figure 3.2. Evans himself predicted that systems able to solve such problems would “be of great practical importance in the near future,” and he pointed out that performance on such tests is often regarded to be the “touchstone” of human intelligence. However, ANALOGY simply hasn’t turned out to be the first system in a longstanding, comprehensive research program (Newellian or otherwise). Why is this? Given our approach and emphasis, this question is a penetrating one. After all, we focus on analogical reasoning, and ANALOGY certainly must be capable of such reasoning. (There is no deduction required by a program able to solve problems in the class in question, but if the artifact was asked to rigorously justify its selection, deduction would unstopably enter the picture.) So again: Given that Evans was by our own herein-advertised lights on the right track, why the derailment?

We think the main reason is summed up in this quote from Fischler & Firschein (1987):

If one were offered a machine purported to be intelligent, what would be an appropriate method of evaluating this claim? The most obvious approach might be to give the machine an IQ test. . . . However, [good performance on tasks seen in IQ tests would not] be completely satisfactory *because the machine would have to be specially prepared for any specific task that it was asked to perform*. The task could not be described to the machine in a normal conversation (verbal or written) if the specific nature of the task was not already

⁵For more on PAI, which of course forms the foundation for PEGI, readers can consult a recent issue of the *Journal of Experimental and Theoretical Artificial Intelligence* devoted to the topic: **23.3**.

⁶The notion that intelligence includes more than *academic* intelligence is unpacked and defended by numerous psychologists. E.g., see (Sternberg, 1988).

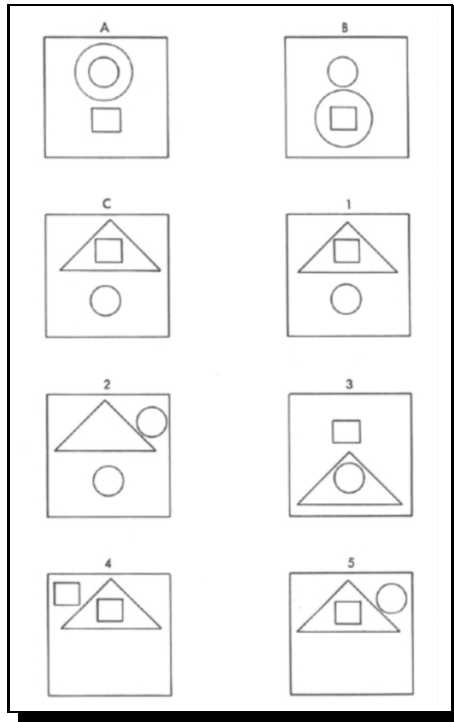


Fig. 3.2 Sample Problem Solved by Evan's (1968) ANALOGY Program. Given sample geometric configurations in blocks A, B, and C, choose one of the remaining five possible configurations that completes the relationship: *A is to B as C is to ...?*. Subjects asked to prove that their answers are correct must resort to analogico-deduction.

programmed into the machine. Such considerations led many people to believe that the ability to communicate freely using some form of natural language is an essential attribute of an intelligent entity (Fischler & Firschein 1987, p. 12; emphasis ours).

3.2.3 *Springboard to the Rest of the Present Paper*

Our response to this response is three-fold. One, there is nothing here that tells against the suspicion that the marriage of analogical and deductive reasoning, which is specifically called for by problems of the sort that the ANALOGY system solved, is at the heart of general intelligence, whether that intelligence is embodied in the mind of a person or machine. Two, a test-based approach to AI can, despite what F&F say, take full account of the requirement that a truly intelligent computing machine must not simply be pre-programmed. Indeed, this is one of the chief points of the PMR. And finally, three, a test-based approach

to uncovering the nature of human intelligence, when broadened in the manner of Piaget, provides a suitable guide to engineering aimed at producing artificial general intelligence.

At this point the reader has sufficient understanding of PAGI to permit us to move on.⁷

3.3 Descartes' Two Tests

Descartes was quite convinced that animals are mechanical machines. He felt rather differently about persons, however: He held that persons, whether of the divine variety (e.g., God, the existence of whom he famously held to be easily provable) or the human, were *more* than mere machines.

Someone might complain that Descartes, coming before the likes of Turing, Church, Post, and Gödel, could not have had a genuine understanding of the concept of a *computing* machine, and therefore couldn't have claimed the human persons are more than such machines. There are two reasons why this complaint falls flat. One, while we must admit that Descartes didn't *exactly* have in the mind the concept of a computing machine in the manner of, say, of a universal Turing machine, or a register machine, and so on, what he did have in mind would subsume such modern logico-mathematical devices. For Descartes, a machine was overtly mechanical; but there is a good reason why recursion theory has been described as revolving around what is *mechanically solvable*. A Turing machine, and ditto for its equivalents (e.g., register machines) are themselves overtly mechanical.

Descartes suggested two tests to use in order to separate mere machines from human persons. The first of these directly anticipates the so-called "Turing Test." The second test is the one that anticipates the Piaget-MacGyver Room. To see this, consider:

If there were machines which bore a resemblance to our body and imitated our actions as far as it was morally possible to do so, we should always have two very certain tests by which to recognize that, for all that, they were not real men. The first is, that they could never use speech or other signs as we do when placing our thoughts on record for the benefit of others. For we can easily understand a machine's being constituted so that it can utter words, and even emit some responses to action on it of a corporeal kind, which brings about a change in its organs; for instance, if it is touched in a particular part it may ask what we wish to say to it; if in another part it may exclaim that it is being hurt, and so on. But it never happens that it arranges its speech in various ways, in order to reply appropriately to everything that may be said in its presence, as even the lowest type of man can do. And the second difference is, that although machines can perform certain things as well as or perhaps better than any of us can do, they infallibly fall short in others, by which means we may discover that they did not act from knowledge, but only for the disposition of their organs. For while reason is a universal instrument which can serve for all contingencies,

⁷For more on PAI, the foundation for PAGI, readers can consult a recent issue of the *Journal of Experimental and Theoretical Artificial Intelligence* devoted to the topic: 23.3.

these organs have need of some special adaptation for every particular action. From this it follows that it is morally impossible that there should be sufficient diversity in any machine to allow it to act in all the events of life in the same way as our reason causes us to act (Descartes 1911, p. 116).

We now know all too well that “machines can perform certain things as well or perhaps better than any of us” (witness Deep Blue and Watson, and perhaps, soon enough, say, auto-driving cars that likewise beat the pants off of human counterparts); but we also know that these machines are engineered for specific purposes that are known inside and out ahead of time. PMR is designed specifically to test for the level of proficiency in using what Descartes here refers to as a “universal instrument.” This is so because PMR inherits Piaget’s focus on general-purpose reasoning. We turn now to a brief discussion of Piaget and this focus.

3.4 Piaget’s View of Thinking & The Magnet Test

Many people, including many outside psychology and cognitive science, know that Piaget seminally — and by Bringsjord’s lights, correctly — articulated and defended the view that mature human reasoning and decision-making consists in processes operating for the most part on formulas in the language of classical extensional logic (e.g., see Inhelder and Piaget, 1958b).⁸ You may yourself have this knowledge. You may also know that Piaget posited a sequence of cognitive stages through which humans, to varying degrees, pass; we have already referred above to Stages III and IV. How many stages are there, according to Piaget? The received answer is: four; in the fourth and final stage, *formal operations*, neurobiologically normal humans can reason accurately and quickly over formulas expressed in the logical system known as first-order logic, \mathcal{L}_1 . This logic allows for use of relations, functions, the universal and existential quantifiers, the familiar truth-functional connectives from the propositional calculus, and includes a so-called “proof theory,” that is, a mechanical method for deriving some formulas from others.⁹ One cornerstone of every classical proof theory, as the reader will likely well know, is *modus ponens*, according to which the formula ψ can be derived from the formulas ϕ and $\phi \rightarrow \psi$ (read: if ϕ then ψ).

⁸Many readers will know that Piaget’s position long ago came under direct attack, by such thinkers as Wason and Johnson-Laird (Wason, 1966; Wason and Johnson-Laird, 1972). In fact, unfortunately, for the most part academics believe that this attack succeeded. Bringsjord doesn’t agree in the least, but this isn’t the place to visit the debate in question. Interested readers can consult (Bringsjord, Bringsjord and Noel, 1998; Rinella, Bringsjord and Yang, 2001). Piaget himself retracted any claims of *universal* use of formal logic: (Piaget, 1972).

⁹A full overview of logic, \mathcal{L}_1 included, in order to model and simulate large parts of cognition, can be found in (Bringsjord, 2008).

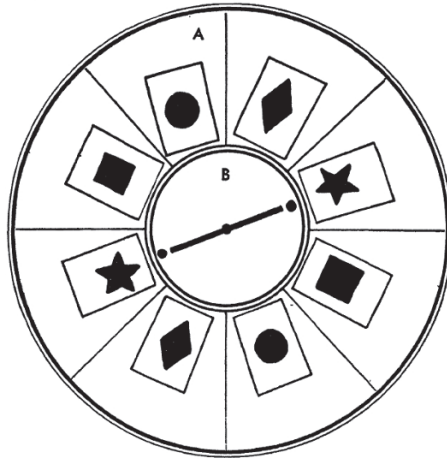


Fig. 3.3 Piaget’s famous “rigged” rotating board to test for the development of Stage-III-or-better reasoning in children. The board, A, is divided into sectors of different colors and equal surfaces; opposite sectors match in color. B is a rotating disk with a metal rod spanning its diameter—but the catch is that the star cards have magnets buried under them (hidden inside wax), so the alignment after spinning is invariably as shown here, no matter how the shapes are repositioned in the sectors (with matching shapes directly across from each other). This phenomenon is what subjects struggle to explain. Details can be found in (Inhelder and Piaget, 1958b).

Judging by the cognition taken by Piaget to be stage-III or stage-IV (e.g., see Figure 3.3, which shows one of the many problems presented to subjects in (Inhelder and Piaget, 1958b), the basic scheme is that an agent \mathcal{A} receives a problem P (expressed as a visual scene accompanied by explanatory natural language), represents P in a formal language that is a superset of the language of \mathcal{L}_I , producing $[P]$, and then reasons over this representation (along with background knowledge, which we can assume to be a set Γ of formal declarative statements) using at least a combination of some of the proof theory of \mathcal{L}_I and “psychological operators.”¹⁰ This reasoning allows the agent to obtain the solution $[S]$. To ease exposition, we shall ignore the heterodox operations that Piaget posits (see note 10) in favor of just standard proof theory, and we will moreover view $[P]$ as a triple (ϕ, C, Q) , where ϕ is a (possibly complicated) formula in the language of \mathcal{L}_I , C is further information that provides context for the problem, and consists of a set of first-order

¹⁰ The psychological operators in question cannot always be found in standard proof theories. For example, Piaget held that the quartet I N R C of “transformations” were crucial to thought at the formal level. Each member of the quartet transforms formulas in certain ways. E.g., N is *inversion*, so that $N(p \vee q) = \neg p \wedge \neg q$; this seems to correspond to DeMorgan’s Law. But R is *reciprocity*, so $R(p \vee q) = \neg p \vee \neg q$, and of course this isn’t a valid inference in the proof theory for the propositional calculus or \mathcal{L}_I .

formulas, and Q is a query asking for a proof of ϕ from $C \cup \Gamma$. So:

$$[P] = (\phi, C, Q = C \cup \Gamma \vdash \phi?)$$

At this point a reader might be puzzled about the fact that what we have so far described is exclusively deductive, given that we have said that our focus is reasoning that includes not just deduction, but also analogical reasoning; the key term, introduced above, is *analogico-deduction*. To answer this, and to begin to give a sense of how remarkably far-reaching Piaget's magnet challenge is, first consider how this deduction-oriented scheme can be instantiated.

To begin, note that in the invisible magnetization problem shown in Figure 3.3, which requires stage-III reasoning in order to be solved, the idea is to explain how it is that ϕ^{**} , that is, that the rotation invariably stops with the two stars selected by the rod. Since Piaget is assuming the hypothetico-deductive method of explanation made famous by Popper (1959), to provide an explanation is to rule out hypotheses until one arrives deductively at ϕ^{**} . In experiments involving child subjects, a number of incorrect (and sometimes silly) hypotheses are entertained—that the stars are heavier than the other shaped objects, that the colors of the sections make a difference, and so on. Piaget's analysis of those who discard mistaken hypotheses in favor of ϕ^{**} is that they expect consequences of a given hypothesis to occur, note that these consequences fail to obtain, and then reason backwards by *modus tollens* to the falsity of the hypotheses. For example, it is key in the magnet experiments of Figure 3.3 that “for some spins of the disk, the rod will come to rest upon shapes other than the stars” is an expectation. When expectations fail, disjunctive syllogism allows ϕ^{**} to be concluded. However, the reasoning patterns so far described are only those at the “top level,” and even at that level exclude the *generation* of hypotheses. Beneath the top level, many non-deductive forms of reasoning are perfectly compatible with Piaget's framework, and one thing that is crystal clear on a reading of his many experiments is that subjects draw from past experience to by analogy rule out hypotheses, and to generate hypotheses in the first place.

Hence the magnet challenge, like other famous challenges invented and presented by Piaget, is a portal to a remarkably wide landscape of the makings of general intelligence. This is confirmed not just by taking account of the magnet challenge in the context of Piaget's framework, and more generally in the context of deliberative reasoning and decision-making; it's also confirmed by placing the magnet challenge (and counterparts that can be fashioned from the raw materials for PMR) in the context of broad characterizations of intelligence offered even by AI researchers more narrowly oriented than AGI researchers. For

example, the magnet challenge taps many elements in the expansive, more-than-deduction view of rational intelligence laid out by Pollock (1989), and likewise taps much of the functionality imparted to the more sophisticated kinds of agents that are pseudo-coded in Russell and Norvig (2009).

As will soon be seen, our modeling and simulation of the magnet challenge reflects its requiring much more than straight deduction. But before moving in earnest to that modeling and simulation, we provide a rapid overview of the system we use for analogical reasoning: LISA.

3.5 The LISA model

LISA (*Learning and Inference with Schemas and Analogies*) is the formidable fruit of an attempt to create a neurally-plausible model of analogical reasoning by using a hybrid connectionist and symbolic architecture (Hummel and Holyoak, 2003a; Hummel and Holyoak, 2003b). We here provide only a very brief summary of some relevant features of LISA; for a more detailed description the reader is directed to (Hummel & Holyoak, 2003) and (Hummel & Landy, 2009).

LISA allows for explicit representation of propositional knowledge, the arguments of which can be either token objects or other propositions.¹¹ Propositional knowledge is organized into *analogs*, which contain the proposition nodes, along with other related units: the sub-propositional units which help to bind relational roles within propositions to their arguments, nodes representing the objects (one object unit corresponds to a token object across all propositions within an analog), predicate units which represent the individual roles within a proposition, and higher-level groupings of propositions (Hummel and Landy, 2009). Semantic units, which are outside of and shared by all of the analogs, connect to the object and predicate units.

In self-supervised learning, LISA performs analogical inference by firing the propositional units in a preset order, which propagates down to the semantic units. This allows for units in different analogs to be temporarily mapped to each other if they fire in synchrony, and for new units to be recruited (or *inferred*) if necessary. Of course, many details are left out here in the interests of space; for more, see (Hummel & Holyoak 2003).

¹¹E.g., *knows(Tom, loves(Sally, Jim))*.

3.6 Analogico-Deductive Reasoning in the Magnet Test

The ways in which analogical and deductive reasoning interact in a typical human reasoner are, we concede, complex to the point of greatly exceeding any reasoning needed to excel in the narrower-than-real-life PMR; and, in addition, these ways no doubt vary considerably from person to person. A model such as the one we present here can thus only hope to be a simulation of a *possible* way that a reasoner might solve a problem on the order of the magnet challenge and its relatives.

This said, and turning now to the Piagetian task on which we focus, we first note again that analogical reasoning can often be useful in generation of hypotheses and theories to explain unfamiliar phenomena. For example, Holyoak *et al.* (2001) explain that the wave theory of sound, as it became better understood, was the basis for creating an analogy that described the wave theory of light. Such an analogical mapping would presumably be responsible for inferring the existence of a medium through which light would travel, just as sound needs air or something like it (indeed, the luminiferous aether was of course proposed to be this very medium). In contrast, Newton's particle theory of light would provide an analogical mapping that would not require a medium. Thus, we have two different analogical mappings; and each then suggests slightly different groups of hypotheses, members of which, in both cases, could in turn be tested with a combination of experimentation and deductive reasoning.

Now let's get more specific. Analogico-deductive reasoning in the Piagetian hidden-magnet experiment can be modeled using LISA and Slate together; specifically, a dialogue between an experimenter and a subject referred as 'Gou' provides an interesting basis for doing so (Inhelder and Piaget, 1958a). Gou, who is developmentally in Piaget's concrete operations stage (Stage III), after being presented with the hidden-magnets challenge, does from the start suspect that magnets are responsible — but quickly abandons this hypothesis in favor of the one claiming that the weight of the objects is what leads the needle to repeatedly stop on the stars. The experimenter then asks Gou what he would have to do in order to "prove that it isn't the weight," to which Gou responds by carrying out a series of small experiments designed to prove that weight isn't responsible for the bar's stopping. One of these experiments involves removing the star and diamond boxes, and checking to see if the bar still stops on the heaviest of the remaining boxes. Predictably (given *our* understanding of the background's mechanisms), it does not; this provides Gou with empirical evidence that weight is not causally responsible for the bar's stopping as it invariably does (although

he continues to subsequently perform small experiments to further verify that weight is not responsible).

In our overall model of Gou’s reasoning as being of the analogico-deductive variety, we of course must make use of both deduction and analogical reasoning, woven together. The overall reasoning abides by the deductive framework known as *proof by cases*, which is straightforward and bound to be familiar to all our readers. The core idea is that if one knows that a disjunction

$$\phi_1 \vee \phi_2 \vee \dots \vee \phi_n$$

holds, and knows as well that one or more of the disjuncts ϕ_i fail to hold, then one can infer a new disjunction lacking the false disjuncts. In the case at hand, in light not only of Gou’s thinking, but that of many other subjects, there are four hypotheses in play, as follows (with underlying S-expressions in FOL given in each case).¹²

H1 Weight accounts for the invariance.

- (Initially boardWeighted)

H2 Color accounts for the invariance.

- (Initially boardColored)

H3 Magnets account for the invariance.

- (Initially boardMagnetized)

H4 Order accounts for the invariance.

- (Initially boardOrdered)

The overall proof, which makes use of LISA to carry out analogical reasoning to rule out the hypothesis that weight is the causally responsible element in the test, is shown in Figure 3.4, which we encourage the reader to take a few minutes to assimilate.

We can model Gou’s reasoning process, by first assuming that he already understands that there is some force or property P_{stop} that causes the bar to stop. We can model this by invoking a predicate $More_P(x, y)$, which is true iff a pair of boxes x is more likely to stop the rotating bar than another pair of boxes y . Gou does know that some boxes are heavier than others, which can be represented by predicates of the form $Heavier(x, y)$. We will assume that Gou has some knowledge of the transitivity of weight. Finally, the causal relationship suggested to Gou by the experimenter — that weight is causally

¹²Modeling and simulating the *generation* of the full quartet of hypotheses is outside our scope, and we thus commence our analysis in earnest essentially at the point when this quartet is being entertained.

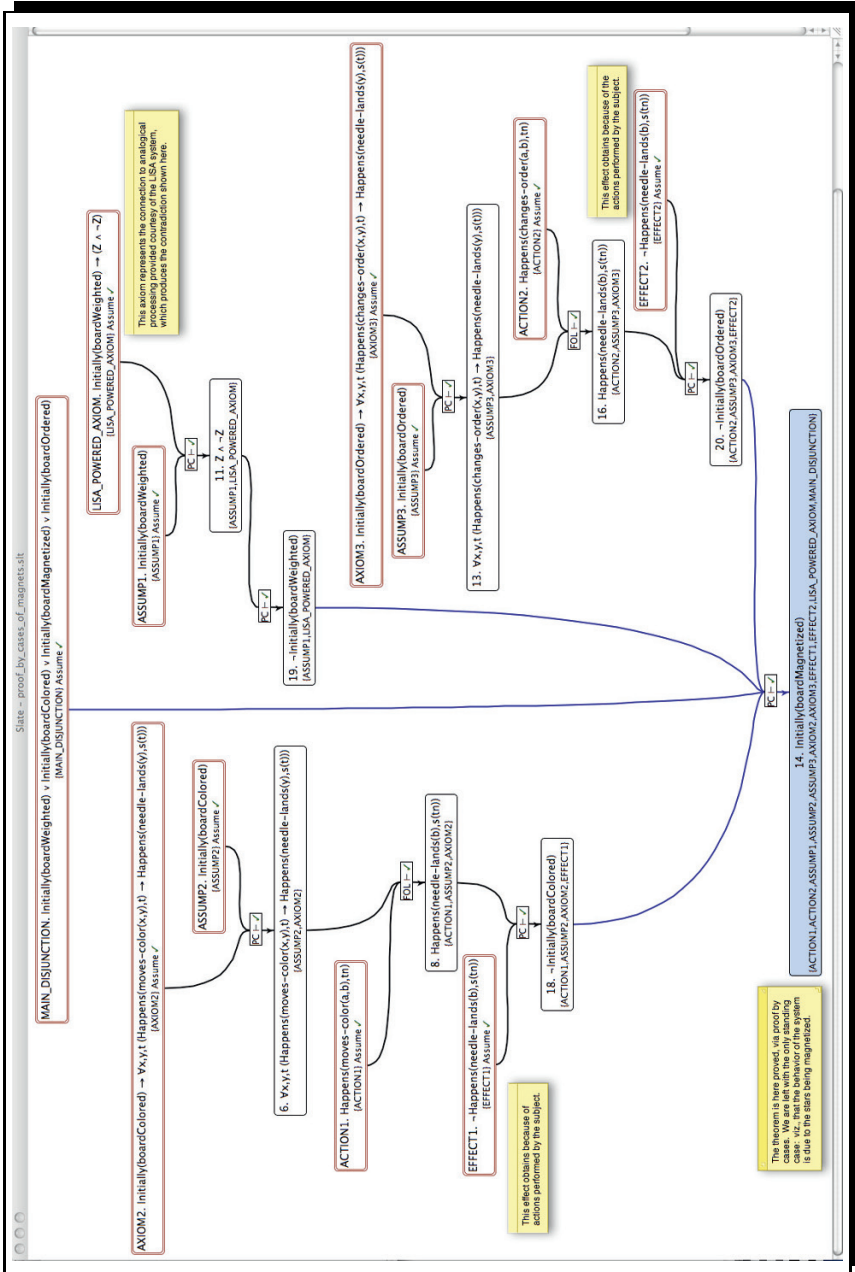


Fig. 3.4 The Top-Level Reasoning Strategy.

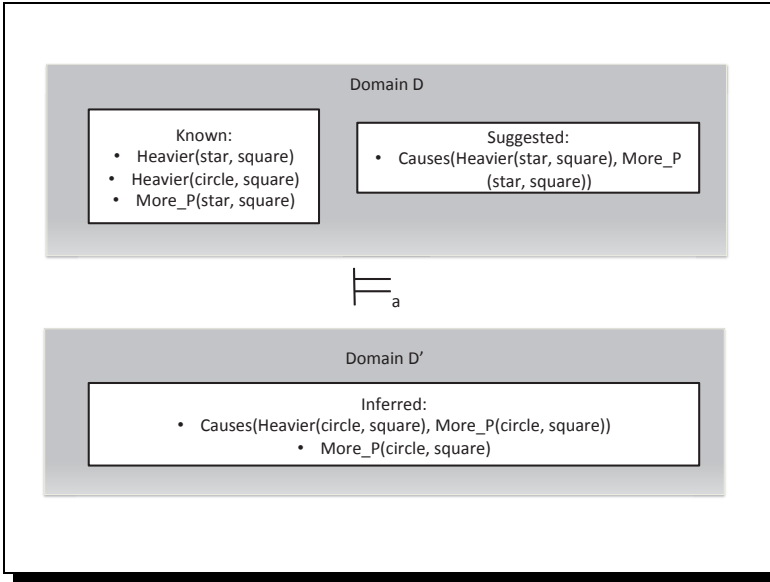


Fig. 3.5 Propositional knowledge used in LISA for the Gou example. Domain D' is inferred from D using LISA's analogical inferencing capability. Propositions representing semantic connections are not pictured here.

responsible for the bar's stopping — is represented using the special-group proposition $\text{Causes}(\text{Heavier}(x,y), \text{More_P}(x,y))$.¹³

In the first stage of this simulation, the set \mathbf{D} , consisting of both propositional knowledge held by Gou and semantic knowledge about the objects in \mathbf{D} , is represented in Slate's memory. Semantic knowledge is represented using a special predicate *Semantic_Prop*, which simply connects an object to its relevant semantic unit. For example, $\text{Semantic_Prop}(\text{bill}, \text{tall})$ and $\text{Semantic_Prop}(\text{jim}, \text{tall})$ connect the objects *bill* and *jim* to the semantic unit *tall*.

\mathbf{D} is then subjected to reasoning by analogical inference. To do this, \mathbf{D} must first be divided into two subsets: $\mathbf{D}_{\text{source}}$ and $\mathbf{D}_{\text{target}}$. Note that these two subsets need not be mutually exclusive or collectively exhaustive — they only need to each be subsets of \mathbf{D} . Choosing which propositions to include in $\mathbf{D}_{\text{source}}$ and $\mathbf{D}_{\text{target}}$ may be an iterative process, the details of which we do not provide at this time. For now, we can assume that in a relatively simple problem such as this, a useful division such as the one we will describe shortly will occur.

¹³Proposition groupings are treated differently in LISA than regular propositions (Hummel and Holyoak, 2003a).

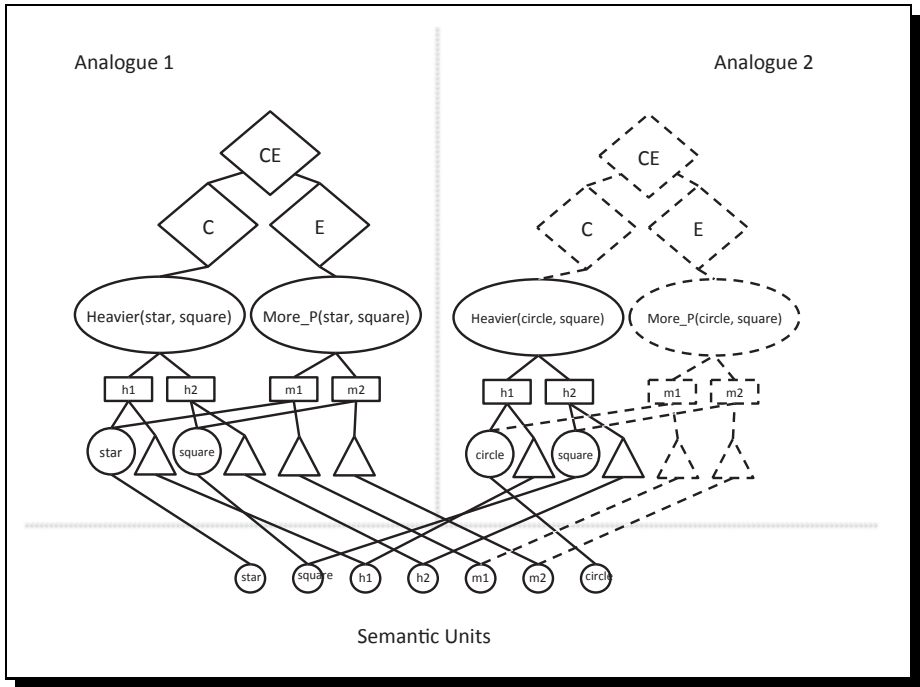


Fig. 3.6 Structured knowledge used in LISA for the Gou example. The units in dotted lines are generated using analogical inference.

D_{source} and D_{target} are then sent to LISA, where they are each used to create an analog. Analogical inference then produces the structure seen in Figure 3.6. Note that the semantic connections from the inferred predicate are mapped to the relevant semantic values as a result of the analogical inference process (Hummel and Holyoak, 2003a). The inferred predicates and semantic connections are then collected as the set D' (Figure 3.5), which is returned to Slate, where it is then subjected to further deductive reasoning. This reasoning over $D \cup D'$ may ideally derive one of two things: a testable hypothesis, which a reasoner would then empirically verify or refute; or a contradiction. A failure to derive either can result in either a repeat of the analogical process with different subsets chosen for D_{source} and D_{target} , or a general failure condition. In the present example, Figure 3.5 shows that D' contains the proposition $More_P(circle, square)$. Gou's experiment, however, shows $\neg More_P(circle, square)$. This leads to a contradiction exploited in Slate, and hence the possibility that weight is causally responsible for whatever force is stopping the metal bar is rejected.

One might ask at this point whether analogical reasoning is necessary to carry out this process. The answer is clearly “No.” But the question is wrong-headed. After all, every elementary logic textbook covering not just deduction, but also induction, abduction, and analogical reasoning,¹⁴ presents the alert reader with formal facts that allow her to see that, *in principle*, deduction can be used to arrive at whatever conclusion is produced in heterogeneous fashion — if additional premises are added. (This is actually an easily proved theorem, given that the commonality to all forms of reasoning is that the content reasoned over is relational and declarative.) Accordingly, while we are not arguing that the precise procedure we have chronicled exactly models the thought processes all reasoners go through, it seems that analogical reasoning produces a plausible explanation.

In fact, consider the following. After Gou is asked to investigate what force or property P_{stop} was responsible for stopping the bars, he might then perform some experiments on the assumption that P_{stop} is transitive. For example, he might think that if the *star* boxes are heavier than the *square* boxes, and a set of boxes b existed that were heavier than the *star* boxes, then the b boxes should be more likely to stop the bar than the *star* boxes. However, it doesn’t follow from deductive reasoning alone that P_{stop} is transitive. After all, it may be the case that stacking two boxes on top of each other would cancel out their relative contributions to P_{stop} , or that the boxes together would have no stronger effect on stopping the rotating bar than they would have alone. He may have suspected that P_{stop} behaved in a similar way to forces familiar to him; forces like gravity or magnetism. If so, analogical ability neatly explains how he would have mapped the properties of magnetism — for example, its ability to pull on some objects more than others — on to P_{stop} . This process suggests to us that he previously understood the transitivity of weight, analogically inferred that P_{stop} was similarly transitive, and formed an easily testable hypothesis.

Note that in the previous paragraph we say “suggests to us.” Although we have stated that complete psychological plausibility is not a primary goal of our simulation (it focuses more on possible ways in which analogical and deductive reasoning can interact), we should note here that Piaget himself was suspicious of the existence of analogical reasoning in children who have not yet reached Stage III. A series of experiments he carried out with Montangero and Billeter seemed to suggest that young children are not capable of consistently performing stable analogical reasoning, and that they instead tend to reason using surface similarity in analogical problems (Piaget, Montangero and Billeter, 2001). Goswami and Brown (1990) recreated a similar experiment with items and relations more

¹⁴E.g., Copi, Cohen and MacMahon (2011)

likely to be familiar to small children; she demonstrated that they indeed had more analogical ability than Piaget suspected. Further experimentation by other researchers showed analogical ability in pre-linguistic children as well (Goswami, 2001). In any case, these results point to the complexity and ever-changing nature of the ways in which analogical and deductive reasoning mix.

Recent work by Christie & Gentner (2010) suggests that at least in the case of young children, analogical reasoning is not likely to be used in generating hypotheses — unless the relevant stimuli are presented simultaneously, in a manner that invites side-by-side comparison and higher-level relational abstraction. Instead, the magnet experiment's format would encourage hypotheses based on surface similarity, which presumably would lack the depth to provide a satisfactory set of testable hypotheses. (We see this with most of Piaget's younger subjects: after a while, they simply *give up* (Inhelder and Piaget, 1958a).) The example we presented here does not have the child initially using analogy to generate a theory about weight. Instead, the mapping from weight is triggered by a suggestion from the experimenter himself. Analogico-deductive reasoning is then used to elaborate on this suggestion, and ultimately refute its validity.

3.7 Next Steps

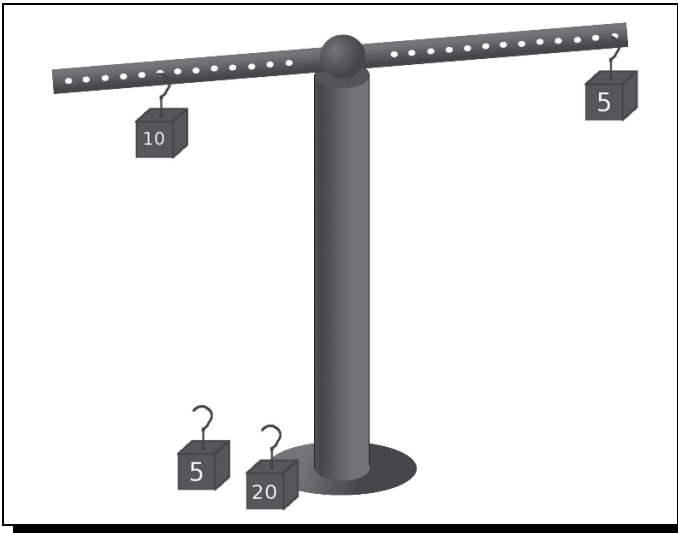


Fig. 3.7 Rendering of the Scale in Piaget's Famous Balance Challenge.

Alert readers will have observed that under the assumption that Piaget can draw from the ingredients in Figure 3.1 to construct a PAGI challenge in PMR, rather more than the magnet challenge is possible. Our next foray into PAGI via analogico-deduction, now underway, involves another of Piaget's challenges: the balance problem. In this challenge, subjects are presented with a scale like that shown in Figure 3.7. To crack this puzzle, the subject must reason to the general rule r that balance is achieved under weight differentials when distance from the vertical post for hanging weights is proportional to the amount of weight in question. Victorious problem-solvers here, like MacGyver, manage in relatively short order to figure out that weights of different sizes can nonetheless be hung so that balance is achieved, as long as r is apprehended, and followed in the physical manipulation. In our work-in-progress, r is represented by a formula in FOL, and is arrived at via — no surprise here — analogico-deduction. Use of such reasoning is supported by what is seen in the subjects; for example in the case of a child who finds the secret to the balance puzzle in the game of marbles, which, if you look carefully, you will indeed see listed in Figure 3.1 as raw material for PMR.

Bibliography

- Anderson, J. and Lebiere, C. (2003). The Newell test for a theory of cognition, *Behavioral and Brain Sciences* **26**: 587–640.
- Anderson, J. R. (1993). *Rules of Mind*, Lawrence Erlbaum, Hillsdale, NJ.
- Anderson, J. R. and Lebiere, C. (1998). *The Atomic Components of Thought*, Lawrence Erlbaum, Mahwah, NJ.
- Bringsjord, S. (2008). Declarative/Logic-Based Cognitive Modeling, in R. Sun (ed.), *The Handbook of Computational Psychology*, Cambridge University Press, Cambridge, UK, pp. 127–169.
- Bringsjord, S. and Schimanski, B. (2003). What is artificial intelligence? Psychometric AI as an answer, *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, Morgan Kaufmann, San Francisco, CA, pp. 887–893.
- Bringsjord, S., Bringsjord, E. and Noel, R. (1998). In Defense of Logical Minds, *Proceedings of the 20th Annual Conference of the Cognitive Science Society*, Lawrence Erlbaum, Mahwah, NJ, pp. 173–178.
- Cassimatis, N. (2002). *Polyscheme: A Cognitive Architecture for Integrating Multiple Representation and Inference Schemes*, PhD thesis, Massachusetts Institute of Technology (MIT).
- Cassimatis, N., Trafton, J., Schultz, A. and Bugajska, M. (2004). Integrating cognition, perception and action through mental simulation in robots, in C. Schlenoff and M. Uschold (eds), *Proceedings of the 2004 AAAI Spring Symposium on Knowledge Representation and Ontology for Autonomous Systems*, AAAI, Menlo Park, pp. 1–8.
- Christie, S. and Gentner, D. (2010). Where hypotheses come from: Learning new relations by structural alignment, *Journal of Cognition and Development* **11**(3): 356–373.
- Copi, I., Cohen, C. and MacMahon, K. (2011). *Introduction to Logic*, Prentice-Hall, Upper Saddle River, NJ. This is the 14th (!) edition of the book.

- Descartes, R. (1911). *The Philosophical Works of Descartes, Volume 1. Translated by Elizabeth S. Haldane and G.R.T. Ross*, Cambridge University Press, Cambridge, UK.
- Evans, G. (1968). A program for the solution of a class of geometric-analogy intelligence-test questions, in M. Minsky (ed.), *Semantic Information Processing*, MIT Press, Cambridge, MA, pp. 271–353.
- Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A., Lally, A., Murdock, W., Nyberg, E., Prager, J., Schlaefer, N. and Welty, C. (2010). Building Watson: An Overview of the DeepQA Project, *AI Magazine* pp. 59–79.
- Fischler, M. and Firschein, O. (1987). *Intelligence: The Eye, the Brain, and the Computer*, Addison-Wesley, Reading, MA.
- Goswami, U. (2001). Analogical reasoning in children, in D. Gentner, K. J. Holyoak and B. N. Kokinov (eds), *The Analogical Mind: Perspectives from Cognitive Science*, The MIT Press.
- Goswami, U. and Brown, A. L. (1990). Melting chocolate and melting snowmen: Analogical reasoning and causal relations, *Cognition* **35**(1): 69–95.
- Holyoak, K. J., Gentner, D. and Kokinov, B. N. (2001). Introduction: The place of analogy in cognition, in D. Gentner, K. J. Holyoak and B. N. Kokinov (eds), *The Analogical Mind: Perspectives from Cognitive Science*, The MIT Press, chapter 1.
- Hummel, J. E. and Holyoak, K. J. (2003a). Relational reasoning in a neurally-plausible cognitive architecture: An overview of the lisa project, *Cognitive Studies: Bulletin of the Japanese Cognitive Science Society* **10**: 58–75.
- Hummel, J. E. and Holyoak, K. J. (2003b). A symbolic-connectionist theory of relational inference and generalization, *Psychological Review* **110**: 220–264.
- Hummel, J. E. and Landy, D. H. (2009). From analogy to explanation: Relaxing the 1:1 mapping constraint...very carefully, in B. Kokinov, K. J. Holyoak and D. Gentner (eds), *New Frontiers in Analogy Research: Proceedings of the Second International Conference on Analogy*, Sofia, Bulgaria.
- Inhelder, B. and Piaget, J. (1958a). *The Growth of Logical Thinking: From Childhood to Adolescence*, Basic Books, Inc.
- Inhelder, B. and Piaget, J. (1958b). *The Growth of Logical Thinking from Childhood to Adolescence*, Basic Books, New York, NY.
- Newell, A. (1973). You can't play 20 questions with nature and win: Projective comments on the papers of this symposium, in W. Chase (ed.), *Visual Information Processing*, New York: Academic Press, pp. 283–308.
- Penrose, R. (1994). *Shadows of the Mind*, Oxford, Oxford, UK.
- Piaget, J. (1972). Intellectual evolution from adolescence to adulthood, *Human Development* **15**: 1–12.
- Piaget, J., Montangero, J. and Billeter, J. (2001). The formation of analogies, in R. Campbell (ed.), *Studies in Reflecting Abstraction*, Psychology Press.
- Pollock, J. (1989). *How to Build a Person: A Prolegomenon*, MIT Press, Cambridge, MA.
- Popper, K. (1959). *The Logic of Scientific Discovery*, Hutchinson, London, UK.
- Rinella, K., Bringsjord, S. and Yang, Y. (2001). Efficacious Logic Instruction: People are not Irremediably Poor Deductive Reasoners, in J. D. Moore and K. Stenning (eds), *Proceedings of the Twenty-Third Annual Conference of the Cognitive Science Society*, Lawrence Erlbaum Associates, Mahwah, NJ, pp. 851–856.
- Rosenbloom, P., Laird, J. and Newell, A. (eds) (1993). *The Soar Papers: Research on Integrated Intelligence*, MIT Press, Cambridge, MA.
- Russell, S. and Norvig, P. (2002). *Artificial Intelligence: A Modern Approach*, Prentice Hall, Upper Saddle River, NJ.
- Russell, S. and Norvig, P. (2009). *Artificial Intelligence: A Modern Approach*, Prentice Hall, Upper Saddle River, NJ. Third edition.

- Sternberg, R. (1988). *The Triarchic Mind: A New Theory of Human Intelligence*, Viking, New York, NY.
- Sun, R. (2001). *Duality of the Mind*, Lawrence Erlbaum Associates, Mahwah, NJ.
- Wang, P. (2006). The logic of intelligence, in B. Goertzel and C. Pennachin (eds), *Artificial General Intelligence*, Springer, New York, NY, pp. 31–62.
- Wason, P. (1966). Reasoning, *New Horizons in Psychology*, Penguin, Hammondsworth, UK.
- Wason, P. and Johnson-Laird, P. (1972). *Psychology of Reasoning: Structure and Content*, Harvard University Press, Cambridge, MA.

Chapter 4

Beyond the Octopus: From General Intelligence toward a Human-like Mind

Sam S. Adams¹ and Steve Burbeck²

¹ *IBM Research*

² *evolutionofcomputing.org*

E-mail: ssadams@us.ibm.com, sburbeck@mindspring.com

General intelligence varies with species and environment. Octopuses are highly intelligent, sensing and rapidly learning the complex properties of their world. But as asocial creatures, all their learned knowledge dies with them. Humans, on the other hand, are exceedingly social, gathering much more complex information and sharing it with others in their family, community and wider culture. In between those extremes there are several distinct types, or levels, of reasoning and information sharing that we characterize as a metaphorical “ladder” of intelligence. Simple social species occupy a “rung” above octopuses. Their young passively learn the ways of their species from parents and siblings in their early lives. On the next rung, “cultural” social animals such as primates, corvids, cetaceans, and elephants actively teach a complex culture to their young over much longer juvenile learning periods. Human-level intelligence relies on all of those lower rungs and adds three more: information sharing via oral language, then literacy, and finally civilization-wide sharing. The human mind, human behavior, and the very ontology with which we structure and reason about our world relies upon the integration of all these rungs. AGI researchers will need to recapitulate the entire ladder to produce a human-like mind.

4.1 Introduction

Multi-strategy problem solving, spatial reasoning, rich sensory perception in multiple modalities, complex motor control, tool usage, theory-of-mind and even possibly consciousness – these and other capabilities form the target for digital systems designed to exhibit Artificial General Intelligence (hereafter AGI) across a broad range of environments and domains.

The ultimate goal of most AGI research is to create a system that can perform as well as humans in many scenarios and perhaps surpass human performance in some. And yet most of the capabilities listed above are already exhibited by the octopus, a solitary asocial creature that does not interact with its own kind save for a brief mating period at the end of its short life. The octopus learns very quickly and solves problems in idiosyncratic and creative ways. Most AGI researchers, let alone businesses and governments, would be thrilled to have systems that function with as much learning ability, creativity, and general intelligence as an adult octopus, and yet no system today comes even close.

This chapter examines the lessons AGI researchers can learn from the capabilities of the octopus and the more social animals up to and including humans. We do not attempt to characterize the intelligence of animals and humans, but rather focus on what sort of information they have to reason with, dependencies between different sorts of information, and the degree to which they learn from or pass information to others of their species.

4.2 Octopus Intelligence

The octopus is an asocial genius that survives by its wits. The common octopus (*Octopus vulgaris*) lives from 12 to 18 months. A mature female mates, lays tens of thousands of eggs [1], tends them until they hatch, and dies soon thereafter. The tiny octopus hatchlings disperse quickly and seldom encounter others of their species until they eventually mate. The hatchlings spend 45 to 60 days floating in ocean currents and feeding in the plankton layer where most of them perish, becoming food for other predators. The small proportion that survive this stage grow rapidly, “parachute” to the sea floor, and begin a bottom dwelling life in an environment that is quite different from the plankton environment [2]. When they land on the bottom, typically far away from where they hatched, octopuses must learn very quickly and be very lucky to survive.

The typical adult octopus has a relatively large brain, estimated at 300 million neurons [3]. The ratio of octopus brain to body mass is much higher than that of most fish and amphibians, a ratio more similar to that of birds and mammals. The complex lobes of the octopus brain support an acute and sensitive vision system, good spatial memory, decision-making, and camouflage behavior. “Sensory and motor function is neatly separated into a series of well-defined lobes... There are two parallel learning systems, one for touch and one for vision, and a clear hierarchy of motor control [4].” Each arm has smaller, mostly independent neural systems (about 50 million neurons each) that deal with

chemical sensors, delicate touch sensors, force sensors, and control of the muscles in that arm. All this processing power supports general intelligence, but at a cost. Neurons use more energy than other cells. Just the photoreceptors in the eyes of a fly consume 8% of the fly's resting energy [5]. The metabolic costs of an octopus' large brain must be justified by its contribution to rapid learning of more effective foraging and more effective defenses against predators.

Adult octopuses are quite clever, adaptable, and rapid learners. Experts speculate that most octopus behaviors are learned independently rather than being based on instinct. At least one researcher [6] posits that cephalopods may even have a primitive form of consciousness.

The following anecdotes illustrate some of their most notable learning and creative talents:

Opening a screw top jar

A five-month-old female octopus in a Munich zoo learned to open screw-top jars containing shrimp by pressing her body on the lid, grasping the sides with her eight tentacles and repeatedly twisting her body. She apparently learned this trick by watching human hands do the same task.

Using coconut halves as portable shelters

An octopus in Indonesia was observed (and filmed [7]) excavating a half of a coconut husk buried in sand, carrying it to the location of another similar half many meters away, crawling into one half and pulling the other over itself to hide from predators.

Shooting out the lights

An aquarium in Coburg, Germany was experiencing late-night blackouts. Upon investigation it turned out that their octopus had learned to "... swing onto the edge of his tank and shoot out the 2000 Watt spot light above him with a carefully directed jet of water [8]."

Spatial learning

Studies show that octopuses learn maps of the territory in which they hunt. Researchers have "... traced young *Octopus vulgaris* in Bermuda on many of these hunting excursions and returns [*typically returning by routes different from their outward path*]. The octopuses

seemed to cover different parts of their home range one after another on subsequent hunts and days [9].” In controlled laboratory experiments octopuses learn to navigate mazes and optimize their paths. They find short cuts as if they could reason about the maze in its entirety from an internal map they have constructed for themselves.

Observational learning

Formal experiments show that captive octopuses can learn to choose the “correct” colored ball from a pair placed in their tank by observing other octopuses trained to do the task [10]. More noteworthy is that it required between 16 and 22 trials to train the “demonstrator” octopuses via formal conditioning (food reward for “correct” choices and electric shock punishment for “wrong” choices), yet the “observer” octopuses learned in as few as five trials.

Camouflage and behavioral mimicry

All cephalopods can dramatically alter their appearance by changing the color, patterning, and texture of their skin [11]. A few species of octopus also disguise themselves by mimicking the shape and movements of other animals in their environment. One Caribbean octopus that inhabits flat sandy bottoms disguises itself by imitating the coloring, shape and swimming behavior of a kind of flounder (a bottom dwelling flatfish) [12]. An Indonesian octopus (*Thaumoctopus mimicus*) learns to mimic the shape, coloring, and movement of various poisonous or dangerous fish that the octopus’ potential predators avoid [13]. It impersonates several species and may shift between impersonations as it crosses the ocean floor. Individual octopuses apparently learn these tricks on their own. Researchers point out that “... all animals were well separated (50m–100m apart) and all displays were observed in the absence of conspecifics [14].”

Using its siphon to squirt an offending spotlight and using coconut halves to build a shelter against predators have been asserted to qualify as a sort of tool use. Whether or not that assertion is fully justified, the behaviors are quite creative. Furthermore, octopus mimicry suggests an intelligent response, even a possible meta-cognitive “theory of predator behavior” that is used to avoid unwanted interaction with predators. Less tangible evidence of octopus general intelligence comes from the assertion by many professional aquarists that octopuses have distinct personalities [15].

Octopuses seem to be so clever, learn so fast, and are so creative that one might wonder why 99.99% of them fail to survive long enough to reproduce. However, we must be

cautious about drawing conclusions from the behavior of the rare octopus that reaches adulthood. Octopuses learn so rapidly in such complex environments that many of the associations they learn can best be thought of as ineffective or even counterproductive superstitions that may be fatal the next time they are invoked. AGI systems that jump to conclusions too quickly may face a similar fate.

4.3 A “Ladder” of Intelligence

Unlike the octopus, humans can rely upon a large legacy of knowledge learned from, and actively taught by, parents, peers, and the culture at large. Social animals also make use of legacies of information that they are able to effectively transfer from one individual to the next and one generation to the next. More effective knowledge legacies go hand-in-hand with more intelligent reasoning although the correlation is far from perfect, as the octopus demonstrates.

Here we discuss a metaphorical ladder of cognitive abilities, each successive rung of which is characterized by larger and more complex legacies of knowledge. Reasoning at each rung of the ladder subsumes the capabilities of the lower rungs and accumulates additional sorts of information required to reason about and interact with more complex aspects of the world. Animals on the lowest rung, the octopus being perhaps the most intelligent, are asocial. They sense and act within their own bodies and their immediate environment, learning by trial-and-error with no cooperative interactions with others. What they learn dies with them. Less solitary animals participate in increasingly complex social interactions that communicate information learned by their ancestors and peers. Humans can act more intelligently than animals in part because we are able to share more information more effectively via oral language, music, art, crafts, customs, and rituals. Based upon oral language, humans have developed written language that supports logic, science, formal government, and ultimately civilization-wide sharing of knowledge. Human intelligence, the eventual target for AGI, depends upon the combined capabilities of all of the rungs as described below.

Asocial reasoning, the lowest rung, does not require cooperation or communication with other animals. Asocial animals reproduce via eggs, often large numbers of eggs, and the young fend for themselves from birth without any parental guidance or protection. These animals learn nothing from others of their species, do not cooperate with other creatures, and pass nothing they learn on to the next generation. Asocial animals learn

regularities in the world on their own by interacting with an environment that is at best indifferent and at worst predatory or dangerous. Typically, only a small percentage of them survive to adulthood.

Social reasoning arises in animals where the young are tended by parents and interact with siblings and perhaps others of their species. As a result, they learn in an environment largely shaped by the parents. One generation thereby passes some knowledge to the next: what is edible and where to find it in the environment, how to hunt, or what predators to avoid and how to do so.

Animal cultural reasoning, found in species such as primates, elephants, corvids, dolphins, wolves, and parrots, requires considerably more transfer of information from one generation to the next. Parents and others of such species actively *teach* the young over relatively long childhoods. Communication in these species includes non-linguistic but nonetheless complex vocalizations and gestures. Parents must teach those communication skills in addition to accumulated culture.

Oral linguistic reasoning is unique to humans despite proto-linguistic behavior at the animal cultural rung. Language not only supports a much richer transfer of intergenerational information, but also a much richer sort of reasoning. Oral language is evanescent, however, not lingering in the minds of either speaker or listener for long unless deliberately memorized. Thus oral cultures are rich in social practices that aid memory such as ritual, storytelling, and master-apprentice relationships.

Literate reasoning depends upon oral language, but is qualitatively different from oral linguistic reasoning. For its first thousand years, writing merely preserved oral utterances. Reading required speaking out-loud until the ninth century [16] and even today many readers silently verbalize internally as they read. In the western hemisphere, literate skills were confined to a small subculture of priests and scribes for hundreds of years until literacy began to spread rapidly in the Renaissance.

Language committed to writing has several advantages over speech. Writing can immortalize long complex structures of words in the form of books and libraries of books. The preserved words can be reread and reinterpreted over time and thereby enable much longer and more complex chains of reasoning that can be shared by a larger group of thinkers. The collaboration enabled by written language gave birth to science, history, government, literature and formal reasoning that could not be supported by oral communication alone.

Finally, *Civilization-scale reasoning* applies to the ideas and behavior of large populations of humans, even entire civilizations. Such ideas impact what every individual human says or does. Long-lasting ideas (memes) evolve, spread and recombine in huge, slow, interconnected human systems exemplified by philosophies, religions, empires, technologies and commerce over long timescales and large geographies. Recent technologies such as the Internet and worldwide live satellite television have accelerated the spread and evolution of memes across these temporal and geographic scales. Nonetheless, long-standing differences in language, religion, philosophy, and culture still balkanize civilizations.

Within a human mind, all rungs are active at all times, in parallel, with different agendas that compete for shared resources such as where to direct the eyes, which auditory inputs to attend to (the “cocktail party effect” [17]), what direction to move (e.g., fight or flight decisions), or what the next utterance will be. For example, direction of gaze is a social signal for humans and many animals precisely because it provides information about which of many internal agendas has priority.

4.4 Linguistic Grounding

Linguistic communication depends upon understanding the meaning of words (the familiar “Symbol Grounding Problem” [18]) as well as the meaning of longer utterances. In social circumstances, the meaning of an utterance may include context from a prior utterance in the current conversation or at some time in the past. Or the meaning may “simply” be that it was uttered at all in a particular social context [19].

Each rung of the ladder surfaces in human language. The meaning of individual words and multi-word utterances often are grounded far lower than the verbal rung and often involve memes at more than one level. For example, *stumble*, *crawl*, *fall*, *hungry*, *startle*, *pain*, and *sleep* are grounded on basic facts of human bodies. We also use such words metaphorically e.g., “stumble upon some problem or situation” or “trip over an awkward fact.” We also “hunger for love” and “slow to a crawl”. Words such as *regret* and *remorse* are grounded in the subtleties of human emotion and memory. An AGI cannot be expected to understand such words based only on dictionary definitions, or a semantic net, without having some exposure to the underlying phenomena to which they refer.

Consider the rungs at which the root meanings of the following English words are grounded:

- *Hide, forage, hunt, kill, flee, eat, what* and *where* are most deeply grounded on the asocial rung. They typically signal object parsing (what), spatial reasoning (where) and other survival issues crucial even to an asocial individual. Yet these same words may take on other meanings in a human social arena when they involve group behavior. To properly interpret such words either literally or metaphorically requires some “gut-level” grounding on the asocial rung.
- *Nurture, protect, feed, bond, give, and share* are grounded within the social rung. They refer to issues fundamental to the social relations within groups of animals, including humans. “*Who*” is especially crucial because humans and other social animals often must recognize individuals of their species to determine if they are likely to be friendly, neutral, dangerous, or a competitor. Individuals are distinguishable from one another by subtle visual, olfactory, auditory, or movement cues that do not translate readily into language.
- *Follow, cooperate, play, lead, warn, trick, steal* (as opposed to simply *take*), and *teach* (in the sense of interacting in a way designed to maximize its teaching value) are grounded within the non-linguistic animal cultural rung where more instinctive social behaviors extend into intentional meta-cognition (e.g., theory-of-mind). These behaviors occur in groups of elephants, corvids, cetaceans, parrots and primates, among others. They depend not only upon accurate classification of relationships and recognition of individuals and their relative roles, but also on memories of the history of each relationship.
- *Promise, apologize, oath, agree, covenant, name* (as in a person’s name or the name of an object or place), *faith, god, game, gamble, plan, lie* (or *deceive*), *ritual, style, status, soul, judge, sing, clothing* (hence *nakedness*), and above all *why*, are grounded in the human oral linguistic rung and depend on shared culture, customs and language – but are not grounded in literacy.
- *Library, contract, fiction, technology, essay, spelling, acronym, document, and book* are grounded in literate culture that has accumulated collective wisdom in written documents. New but similar conventions have already grown around video and audio recordings. Not only can such documents be read, debated, and reasoned about over wide geographies and time periods, but they also support more complex interrelated arguments across multiple documents.
- Above the level of human current affairs, there are more abstract concepts such as *democracy, empire, philosophy, science, mathematics, culture, economy, nation, liter-*

ature and many others that are about vast collections of memes evolving over decades or centuries within the minds of large numbers of people. Individual humans have some local sense of the meaning of such concepts even though their understanding may be little better than a fish's awareness of water.

An AGI that could not convincingly use or understand most of the above words and thousands more like them will not be able to engage even in flat, monotone, prosaic human conversations. In our view, such an AGI would simply not be at the *human-level* no matter how well it can do nonverbal human tasks. AGI systems will need to be evaluated more like human toddlers [20] instead of adult typists in a Turing Test.

4.5 Implications of the Ladder for AGI

The ladder metaphor highlights the accumulation of knowledge from generation to generation and the communication of that knowledge to others of the species. Each rung of the ladder places unique requirements on knowledge representation and the ontologies required for reasoning at that level.

The term ontology is used differently, albeit in related ways, in philosophy, anthropology, computer science, and in the “Semantic Web” [21]. One definition of ontology commonly used in computer science is: “a formal representation of a set of concepts within a domain and the relationships between those concepts.” In philosophical metaphysics, ontology is concerned with what entities exist or can be said to exist, how such entities can be grouped or placed in some hierarchy, or grouped according to similarities and differences. Within recent anthropological debates, it has been argued that ontology is just another word for culture [22]. None of the above definitions quite do the trick in our context. For the purposes of the following discussion, the term ontology is used to describe the organization of the internal artifacts of mental existence in an intelligent system or subsystem, including an AGI system.

It is not our goal here to define a specific ontology for an AGI. In fact we argue that goal is pointless, if not impossible, because an ontology appropriate for a solitary asocial octopus has little in common with one appropriate for a herd herbivore such as a bison, a very long lived highly social animal such as an elephant, or a linguistically competent human. Instead, we seek to explore the implications of the different design choices faced by researchers seeking to develop AGI systems [23]. Since we are concerned here with human-level AGI, we will discuss the ontological issues related to the human version of

the rungs of the ladder: asocial, social, animal cultural, linguistic, literate, and civilization-level. Let us examine each in turn.

Asocial ontologies – Humans share with many asocial animals the ability to process and act upon visual 2D data and other spatial maps. Octopuses, insects, crabs and even some jellyfish [24] use visual information for spatial navigation and object identification. The octopus is exceptionally intelligent, with complex behavior befitting its large brain and visual system. Its ontology has no need for social interaction and may encompass no more than a few hundred categories or concepts representing the various predators and prey it deals with, perhaps landmarks in their territories, maps of recent foraging trips and tricks of camouflage. It presumably does not distinguish one instance of a class from another, for example, one particular damselfish from another. Octopus ontology also presumably supports the temporal sequences that underlie the ability of the octopus to make and execute multi-step plans such as shooting out the lights, opening shrimp jars, or building coconut shell shelters, although one can posit other mechanisms. Humans also have equivalents of other asocial processing abilities such as the ability to sense and process information about temporal ordering, proprioception, audition, and the chemical environment (smell, taste). What can AGI researchers learn from such parallels?

In humans, the rungs are not as separable as AGI researchers might wish. Human infants are hardwired to orient to faces, yet that hardwired behavior soon grows into a social behavior. Infants cry asocially at first, without consideration of impact or implications on others, but they soon learn to use crying socially. The same can be said for smiling and eating, first applied asocially, and then adapted to social purposes. In summary, many of our asocial behaviors and their supporting ontology develop over infancy into social behaviors. The social versions of asocial behaviors seem to be elaborations, or layers, that obscure but do not completely extinguish the initial asocial behavior, e.g., unceremoniously wolfing down food when very hungry, or crying uncontrollably when tragedy strikes.

Many behaviors apparent in infants are asocial simply because they are grounded in bodies and brains. Yet we learn to become consciously aware of many of our asocial behaviors, which then become associated with social concepts and become social aspects of our ontology. Because humans learn them over many years in the midst of other simultaneously operating rungs, the ontological categories inevitably become intertwined in ways difficult to disentangle. Learning to understand the issues characteristic of the asocial rung by building an asocial octopus-level AGI would therefore be a good strategy for separation of concerns.

Social ontologies – Social interaction provides a richer learning experience than does hatching into an asocial existence. It ensures that learned ontologies about the environment, foods, and early experiences are biased by the parents and siblings. By sharing a nest or other group-defined environment the experiences of the young are much more like one another, teaching them what they need to know socially. What they learn may be communicated via posture, “body language,” herd behavior, behavioral imprinting (e.g., ducklings imprinting on their mother), pheromones, and many other means. Indirect interaction also may occur via persistent signals deposited on inanimate features of the environment, a phenomenon known as stigmergy [25]. Stigmergy is best understood in social insects where signals deposited on physical structures, e.g., termite mounds, honeycombs, or ant trails, affect and partially organize the behavior of the insects. Any modification of the environment by an individual that can influence the behavior of others of its kind can also produce stigmergy. Nearly all higher animals, including humans, make extensive use of such indirect communication channels. Humans create especially rich stigmergy structures: clothes, jewelry, pottery, tools, dwellings, roads, cities, art, writing, video and audio recordings, and most recently the Internet.

Social animals necessarily have larger and more complex ontologies than asocial animals because, *in addition* to what a comparable asocial animal must learn, social animals must learn how to communicate with others of their species. That requires concepts and ontological categories for the communicative signals themselves (body postures, chemical signals such as urine as a territorial marker, sounds, touch, facial expressions and many others) as well as the ability to associate them to specific behaviors that the learner can interpret or mimic.

The human version of such primitive social behavior includes social dominance and submission signals, group membership awareness, predator awareness and warnings. These cognitive skills are crucial to successful cooperation and perhaps to forming primitive morals, such as those that minimize fratricide or incest.

Cultural ontologies – The most intelligent social species *actively teach* their young. Examples include elephants, primates, corvids (ravens, crows, magpies, etc.), parrots, dolphins, whales and wolves. Many of these explicitly pass on information via structured and emotive utterances that lack the syntactic structure necessary to qualify as a language – call them proto-languages.

Elephant groups in the wild communicate with each other using “... more than 70 kinds of vocal sounds and 160 different visual and tactile signals, expressions, and ges-

tures in their day-to-day interactions” [26]. Wild elephants exhibit behaviors associated with grief, allomothering (non-maternal infant care), mimicry, a sense of humor, altruism, use of tools, compassion, and self recognition in a mirror [27]. Parrots develop and use individual unique names for each other that also encode their family and close knit “clan” relationships [28]. Crows even recognize and remember individual human faces and warn each other about humans that have been observed mistreating crows. These warnings spread quickly throughout the flock [29]. Aided by theory-of-mind insights into the juvenile learner’s forming mind, these species purposefully teach about what foods are safe, what predators to flee, the meaning of group signals such as postures and vocalization, the best places to hunt or find water, cooperative behaviors when hunting, and who’s who in the social structure of the group.

Ontologies needed to support animal social cultures must be rich enough to allow for learning the vocal and non-vocal signals used to organize cooperative behavior such as group hunting or defense. The ontology must also support recognition of individuals, placement of those individuals in family and clan relationships, and meta-cognitive (Theory of Mind) models of individuals.

Human versions of the animal cultural rung are quite similar to the animal version when the knowledge to be transferred is not well suited to verbal description. Non-verbal examples might include playing musical instruments, dancing, fishing, athletic activities such as skiing, and perhaps the art of cooking. We are, however, so skilled at developing vocabulary to teach verbally that completely non-verbal human teaching is relatively uncommon. Building a non-verbal cultural AGI beginning with a primitive social AGI may be a difficult step because it will require non-verbal theory-of-mind.

It may be strategically important for AGI researchers to first learn to build a primitive social AGI before attempting to address human culture because it is very difficult to determine whether a given human juvenile behavior has been learned by passive proximity (i.e., social mimicry) rather than by active teaching by adults or siblings (i.e., culture). That distinction is nonetheless important because humans tend to be less conscious of behavior learned by passive mimicry than behavior actively taught. And some behaviors, such as empathy, are very difficult, if not impossible, to teach. A common example is epitomized by a parent saying, “I just don’t know how Mary learned that,” when everyone else recognizes that Mary learned it by mimicking the parent. Moreover, an AGI that did not learn important primitive human social skills, skipping instead to the cultural learning rung may

turn out to be an AGI sociopath: a completely asocial, completely selfish predator with an overlaid set of behavioral rules taught without the necessary social grounding.

Oral linguistic ontologies – Human language (including fully syntactic sign language) facilitates an explosion of concepts which at this level are more appropriately called memes [30]. Meme, as we use the term here, approximates the notion of a concept, often but not necessarily expressible in words. Language both supports and reflects a richer and more complex meme structure than is possible without language. Oral cultures are not “dumbed-down” literate cultures (a common misconception). Primary oral cultures – those that have never had writing – are *qualitatively different* from literate cultures [31].

Language in primary oral societies uses a more complex and idiosyncratic syntax than written language, with rules and customs for combining prefixes, suffixes, and compound words that are more flexible than those used in writing. The rules of oral language have to do with the sound, intonation and tempo of the language as spoken throughout prehistory and in the many non-literate societies that still exist. Such rules define allowable vowel and consonant harmonies, or restrict allowable phoneme usage such as two phonemes that may not occur in the same word [32]. Oral cultures use constructs such as rhyme and rhythm, alliteration and other oratorical patterns to aid in the memorability of utterances without the aid of a written record. Oral cultures also aid memorability via rituals, chanting, poetry, singing, storytelling, and much repetition. And they employ physical tokens and other stigmergy structures such as ritual masks and costumes, notched sticks for recording counts, decorations on pottery or cave walls, clothing and decorations symbolic of status, astrology (used for anticipating and marking seasons, e.g., Stonehenge) and the like.

AGI researchers will need to be exceedingly careful to properly develop an oral-language human-level AGI because academics are so thoroughly steeped in literate intelligence that they may find it difficult to put that experience aside. For example, literate people find it very difficult to grasp that the notion of a “word” is not necessarily well defined and hence it is not necessarily the atomic base of language nor the fundamental ontological concept for a primary oral language (or sign language) [33].

Literate ontologies – Writing emerged from spoken language in complicated ways that vary with the specific language and culture [34]. Writing and other forms of long-term recorded thought allow generations and populations far removed from each other, temporally or physically, to share verbal knowledge, reasoning, experience of events (history), literature, styles of thought, philosophy, religion and technologies. Caravans and ships that once transmitted verbal gossip, tall tales, and rumors, began also to carry scrolls and letters

which communicated more authoritative information. Literacy exposes people to a wider variety of memes than does an oral culture. As literacy became more common, the memes transmitted via writing grew in number and importance.

The specificity and permanence of a written work also allows more formal and longer-range relationships between the parts of the written “argument” than can be accomplished solely with real-time verbal communication. The development in the last century of recorded audio and video has similarly and dramatically changed the way we learn and reason. And now that these kinds of media are globally available over the Internet, we can expect further changes in human reasoning and ontology.

Since AGI researchers are familiar with the relationship between language and intelligence, little more need be said here. But that familiarity does not necessarily tell us what must be added to the ontology or the reasoning skills of an oral-language AGI to support literacy. It took a millennium after the invention of writing for humans to widely adopt literacy. We suspect that there are some mysteries hidden in the transition that will surface only when AGI researchers attempt to add literacy to an oral-only system. Understanding written material and writing original material are not simple modifications of conversation. Writing does not provide for immediate feedback between the writer and the reader to signal understanding or disagreement, nor a clear context in which the information exchange is embedded. The reader cannot interrupt to ask what the writer really means, or why the author even bothered to write the material in the first place. What would AGI researchers need to add to an orally competent conversational AGI for it to pick up a book on its own and read it? Or sequester itself in some upper room to write a book, or even a prosaic email, or a tweet? We simply don’t know.

Civilization-scale ontologies – Over the longer term and wider geography, literacy and other persistent forms of human knowledge can affect large numbers of people who combine and recombine ideas (memes) in new ways to form new memes. These memes spread throughout and across cultures to be further combined and recombined and accepted or forgotten by large portions of the population. Ideas like money, capitalism, democracy, orchestral music, science, agriculture, or Artificial Intelligence can gain or lose momentum as they travel from mind to mind across generations, centuries, and cultures. Cultural memes of this sort are not phenomena that operate at the level of individuals, or even small groups of individuals. They do not run their course in only a few months or years. For example, the idea of humans landing on the moon played out over a century (from Jules Verne, a French author writing in the early 1860’s, by way of German rocket scientists in

the 1940s, to the American Apollo 11 landing in 1969). At no time did any human mind encompass more than a tiny portion of the knowledge required for Neil Armstrong to make his “one giant leap for mankind.”

Humanity collectively reasons about the world in slow, subtle and unobservable ways, and not necessarily with the help of literacy. Some ancient civilizations appear not to have relied on writing, e.g., the Indus Valley civilization [35]. Modern civilizations are continually being changed by audio and video shared over radio, television, mobile phones, and the Internet. Live television broadcasts allowed people worldwide to share the experience of the Apollo-11 moon landing, the 2001 destruction of the World Trade Towers, the recent Arab-spring events, and disasters such as floods, earthquakes and tsunamis. Widely shared events or manifestations of ideas directly affect civilization-level reasoning that is seldom observable in any individual human, yet is increasingly easy to observe in the Internet [36]. The ontological requirements for supporting intelligence at this level are largely unexplored.

The civilization-level may turn out to be where researchers first succeed in building an AGI that can surpass some of the abilities of humans on the basis of orders of magnitude more memory and search speed. IBM’s WATSON [37] seemed to do so, but WATSON isn’t even an AGI, let alone a human-level AGI. Time will tell. First, the AGI needs to be competent in all the other rungs simply to make any sense of what it reads, sees, and hears in the libraries of civilization, or their digital equivalents on the Internet.

4.6 Conclusion

The octopus is clearly quite clever. Building an AGI with intelligence roughly equivalent to that of an octopus would be quite a challenge, and perhaps an unwise one if it were allowed to act autonomously. A human-level AGI is far more challenging and, we believe, quite hopeless if one attempts to start at the higher rungs of the intelligence ladder and somehow finesse the lower rungs or fill them in later.

From the beginning of ancient philosophical discourse through the recent decades of AI and now AGI research, mankind’s quest to understand and eventually emulate the human mind in a machine has borne fruit in the ever-increasing understanding of our own intelligence and behavior as well as the sometimes daunting limitations of our machines. The human mind does not exist in splendid isolation. It depends on other minds in other times and places interacting in multiple ways that we characterize in terms of a metaphorical lad-

der. Mapping out the journey ahead and acknowledging the challenges before us, we must begin at the base of the ladder and climb one rung at a time.

Bibliography

- [1] V. Hernández-García, J.L. Hernández-López, and J.J. Castro-Hdez, “On the reproduction of *Octopus vulgaris* off the coast of the Canary Islands”, *Fisheries Research*, Vol. 57, No. 2, August 2002, pp. 197–203.
- [2] M. Nixon and K. Mangold, “The early life of *Octopus vulgaris* (Cephalopoda: Octopodidae) in the plankton and at settlement: a change in lifestyle”, *Journal of Zoology*, Vol. 239, Issue 2, pp. 301–327, June (1996).
- [3] J.Z. Young, *The anatomy of the nervous system of Octopus vulgaris*, Clarendon Press Oxford (1971).
- [4] M. Wells, “Review of The Anatomy of the Nervous System of *Octopus vulgaris*. By J.Z. Young”, *J. Anat.* 1972 May; 112 (Pt 1): 144.
- [5] J.E. Niven, J.C. Anderson, and S.B. Laughlin, “Fly Photoreceptors Demonstrate Energy-Information Trade-Offs in Neural Coding” *PLoS Biol* 5 (4): e116 (2007).
- [6] J. Mather, “Consciousness in Cephalopods?”, *J. Cosmology*, vol. 14 (2011).
- [7] M. Kaplan, “Bizarre Octopuses Carry Coconuts as Instant Shelters”, *National Geographic News*, December 15, 2009.
- [8] A. Seabrook, “The Story Of An Octopus Named Otto”, *All Things Considered*, National Public Radio, aired November 2, 2008.
- [9] J. Mather, *Journal of Comparative Physiology A* 168, 491–497 (1991).
- [10] G. Fiorito and P. Scotto, “Observational Learning in *Octopus vulgaris*”, *Science*, vol. 256, 24 April 1992, pp. 545–547.
- [11] J.B. Messenger, “Cephalopod chromatophores; neurobiology and natural history”, *Biological Reviews*, Vol. 76, No. 4, pp. 473–528. (2001).
- [12] R.T. Hanlon, A.C. Watson, and A. Barbosa, A “Mimic Octopus” in the Atlantic: Flatfish Mimicry and Camouflage by *Macrotritopus defilippi*”, *Bio. Bull.* 218: 15–24 February 2010.
- [13] M.D. Norman and F.G. Hochberg, “The ‘Mimic Octopus’ (*Thaumoctopus mimicus*), a new octopus from the tropical Indo-West Pacific”, *Molluscan Research*, Vol. 25: 57–70 (2006).
- [14] M.D. Norman, J. Finn and T. Tregenza, “Dynamic mimicry in an Indo-Malayan octopus”, *Proc. R. Soc. Lond. B*, 268, 1755–1758 (2001).
- [15] J.A. Mather, “To boldly go where no mollusk has gone before: Personality, play, thinking, and consciousness in cephalopods”, *American Malacological Bulletin*, 24 (1/2): 51–58. (2008).
- [16] A. Manguel, *A History of Reading*, New York; Viking Press, (1996).
- [17] E.C. Cherry, “Some Experiments on the Recognition of Speech, with One and with Two Ears”, *J. Acoust. Soc. Amer.*, Vol. 25, Issue 5, pp. 975–979 (1953).
- [18] S. Harnad, “The Symbol Grounding Problem”, *Physica D*, 42: 335–346. (1990).
- [19] C. Fleisher Feldman, “Oral Metalanguage”, in *Literacy and Orality*, Olson, D.R., & Torrance, N. (eds.) Cambridge Univ. Press, (1991).
- [20] N. Alvarado, S.S. Adams, S. Burbeck, and C. Latta, “Beyond the Turing Test: Performance metrics for evaluating a computer simulation of the human mind”, *Proceedings of The 2nd International Conference on Development and Learning (ICDL’02)*, Cambridge, MA, IEEE Computer Society (2002).
- [21] T. Berners-Lee, J. Hendler, and O. Lassila, “The Semantic Web”, *Scientific American Magazine*, May 2001.

- [22] S. Venkatesan, "Ontology Is Just Another Word for Culture: Motion Tabled at the 2008 Meeting of the Group for Debates in Anthropological Theory, University of Manchester", *Anthropology Today*, Vol. 24, No. 3, p. 28 (2008).
- [23] A. Sloman, "Evolved Cognition and Artificial Cognition: Some Genetic/Epigenetic Trade-offs for Organisms and Robots", *CiteSeerX*, doi: 10.1.1.193.7193 (2011).
- [24] A. Garm, M. Oskarsson, and D. Nilsson, "Box Jellyfish Use Terrestrial Visual Cues for Navigation", *Current Biology*, Vol. 21, Issue 9, pp. 798–803 (2011).
- [25] <http://evolutionofcomputing.org/Multicellular/Stigmergy.html>
- [26] D.L. Parsell, "In Africa, Decoding the 'Language' of Elephants", *National Geographic News*, February 21, 2003.
- [27] J.M. Plotnik, F.B.M. de Waal, and D. Reiss, "Self-recognition in an Asian elephant", *Proc Natl Acad Sci USA*, October 30, 2006. 103 (45).
- [28] V. Morell, "Parrotlet Chicks Learn Their Calls From Mom and Dad", *Science NOW*, July 12, 2011.
- [29] J.M. Marzluff, J. Walls, H.N. Cornell, J.C. Withey, and D.P. Craig, "Lasting recognition of threatening people by wild American crows", *Animal Behaviour*, Vol. 79, No. 3, March 2010, pp. 699–707.
- [30] S. Blackmore, "Imitation and the definition of a meme", *Journal of Memetics - Evolutionary Models of Information Transmission*, Vol. 2 (1998).
- [31] W.J. Ong, *Orality and Literacy: The Technologizing of the Word* (second edition; orig. 1982). Routledge, London and New York, (2002).
- [32] R. Applegate, in *Papers on the Chumash*, San Luis Obispo County Archaeological Society occasional paper number nine (1975).
- [33] D.R. Olson, and N. Torrance (eds.), *Literacy and Orality*, Cambridge Univ. Press, (1991).
- [34] *Ibid.*
- [35] S. Farmer, R. Sproat, and M. Witzel. "The Collapse of the Indus-Script Thesis: The Myth of a Literate Harappan Civilization," *Electronic Journal of Vedic Studies (EJVS)*, Vol. 11. No. 2 (2004) pp. 19–57.
- [36] J. Michel, Y.K. Shen, A. Presser Aiden, A. Veres, M.K. Gray, W. Brockman, The Google Books Team, J.P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, S. Pinker, M.A. Nowak, and E. Lieberman Aiden. "Quantitative Analysis of Culture Using Millions of Digitized Books", *Science*, December 16, 2010.
- [37] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A.A. Kalyanpur, A. Lally, J.W. Murdock, E. Nyberg, J. Prager, N. Schlaefer, and C. Welty, "Building Watson: An Overview of the DeepQA Project", *AI Magazine*, Association for the Advancement of Artificial Intelligence, (2010).

Chapter 5

One Decade of Universal Artificial Intelligence

Marcus Hutter

*RSCS @ ANU and SML @ NICTA
Canberra, ACT, 0200, Australia*

*& Department of Computer Science
ETH Zürich, Switzerland*

The first decade of this century has seen the nascency of the first mathematical theory of general artificial intelligence. This theory of Universal Artificial Intelligence (UAI) has made significant contributions to many theoretical, philosophical, and practical AI questions. In a series of papers culminating in book [24] an exciting sound and complete mathematical model for a super intelligent agent (AIXI) has been developed and rigorously analyzed. While nowadays most AI researchers avoid discussing intelligence, the award-winning PhD thesis [38] provided the philosophical embedding and investigated the UAI-based universal measure of rational intelligence, which is formal, objective and non-anthropocentric. Recently, effective approximations of AIXI have been derived and experimentally investigated in JAIR paper [79] This practical breakthrough has resulted in some impressive applications, finally muting earlier critique that UAI is only a theory. For the first time, without providing any domain knowledge, the same agent is able to self-adapt to a diverse range of interactive environments. For instance, AIXI is able to *learn* from scratch to play TicTacToe, Pacman, Kuhn Poker, and other games by trial and error, without even providing the rules of the games.

These achievements give new hope that the grand goal of Artificial General Intelligence is not elusive.

This chapter provides an informal overview of UAI in context. It attempts to gently introduce a very theoretical, formal, and mathematical subject, and discusses philosophical and technical ingredients, traits of intelligence, some social questions, and the past and future of UAI.

“The formulation of a problem is often more essential than its solution, which may be merely a matter of mathematical or experimental skill. To raise new questions, new possibilities, to regard old problems from a new angle, requires creative imagination and marks real advance in science.”

— Albert Einstein (1879–1955)

5.1 Introduction

The dream. The *human mind* is one of the great mysteries in the Universe, and arguably the most interesting phenomenon to study. After all, it is connected to *consciousness* and *identity* which define who we are. Indeed, a healthy mind (and body) is our most precious possession. Intelligence is the most distinct characteristic of the human mind, and one we are particularly proud of. It enables us to understand, explore, and considerably shape our world, including ourselves. The field of *Artificial Intelligence* (AI) is concerned with the study and construction of artifacts that exhibit intelligent behavior, commonly by means of computer algorithms. The *grand goal* of AI is to develop systems that exhibit *general intelligence* on a *human-level* or beyond. If achieved, this would have a far greater impact on human society than all previous inventions together, likely resulting in a post-human civilization that only faintly resembles current humanity [31, 36].

The dream of creating such artificial devices that reach or outperform our own intelligence is an old one with a persistent great divide between “optimists” and “pessimists”. Apart from the overpowering technical challenges, research on machine intelligence also involves many fundamental philosophical questions with possibly inconvenient answers: What is intelligence? Can a machine be intelligent? Can a machine have free will? Does a human have free will? Is intelligence just an emergent phenomenon of a simple dynamical system or is it something intrinsically complex? What will our “Mind Children” be like? How does mortality affect decisions and actions? to name just a few.

What was wrong with last century’s AI. Some claim that AI has not progressed much in the last 50 years. It definitely has progressed much slower than the fathers of AI expected and/or promised. There are also some philosophical arguments that the grand goal of creating super-human AI may even be elusive in principle. Both reasons have led to a decreased interest in funding and research on the foundations of Artificial General Intelligence (AGI).

The real problem in my opinion is that early on, AI has focussed on the wrong paradigm, namely deductive logical; and being unable to get the foundations right in this framework, AI soon concentrated on practical but limited algorithms. Some prominent early researchers such as Ray Solomonoff, who actually participated in the 1956 Dartmouth workshop, generally regarded as the birth of AI, and later Peter Cheeseman and others, advocated a probabilistic inductive approach but couldn’t compete with the soon dominating figures such as Marvin Minsky, Nils Nilsson, and others who advocated a sym-

bolic/logic approach as the foundations of AI. (of course this paragraph is only a caricature of AI history).

Indeed it has even become an acceptable attitude that general intelligence is in principle unamenable to a formal definition. In my opinion, claiming something to be impossible without strong evidence sounds close to an unscientific position; and there *are no* convincing arguments against the feasibility of AGI [6, 38].

Also, the failure of once-thought-promising AI-paradigms at best shows that they were not the right approach or maybe they only lacked sufficient computing power at the time. Indeed, after early optimism mid-last century followed by an AI depression, there is renewed, justified, optimism [56, Sec. 1.3.10], as is evident by the new conference series on Artificial General Intelligence, the Blue Brain project, the Singularity movement, and this book prove. AI research has come in waves and paradigms (computation, logic, expert systems, neural nets, soft approaches, learning, probability). Finally, with the free access to unlimited amounts of data on the internet, *information*-centered AI research has blossomed.

New foundations of A(G)I. Universal Artificial Intelligence (UAI) is such a modern information-theoretic inductive approach to AGI, in which logical reasoning plays no direct role. UAI is a new paradigm to AGI via a path from universal induction to prediction to decision to action. It has been investigated in great technical depth [24] and has already spawned promising formal definitions of rational intelligence, the optimal rational agent AIXI and practical approximations thereof, and put AI on solid mathematical foundations. It seems that we could, for the first time, have a general mathematical theory of (rational) intelligence that is sound and complete in the sense of well-defining the general AI problem as detailed below. The theory allows a rigorous mathematical investigation of many interesting philosophical questions surrounding (artificial) intelligence. Since the theory is complete, definite answers can be obtained for a large variety of intelligence-related questions, as foreshadowed by the award winning PhD thesis of [38].

Contents. Section 5.2 provides the context and background for UAI. It will summarize various last century's paradigms for and approaches to understanding and building artificial intelligences, highlighting their problems and how UAI is similar or different to them. Section 5.3 then informally describes the ingredients of UAI. It mentions the UAI-based intelligence measure only in passing to go directly to the core AIXI definition. In which sense AIXI is the most intelligent agent and a theoretical solution of the AI problem is explained. Section 5.4 explains how the complex phenomenon of intelligence with all its

facets can emerge from the simple AIXI equation. Section 5.5 considers an embodied version of AIXI embedded into our society. I go through some important social questions and hint at how AIXI might behave, but this is essentially unexplored terrain. The technical state-of-the-art/development of UAI is summarized in Section 5.6: theoretical results for AIXI and universal Solomonoff induction; practical approximations, implementations, and applications of AIXI; UAI-based intelligence measures, tests, and definitions; and the human knowledge compression contest. Section 5.7 concludes this chapter with a summary and outlook how UAI helps in formalizing and answering deep philosophical questions around AGI and last but not least how to build super intelligent agents.

5.2 The AGI Problem

The term AI means different things to different people. I will first discuss why this is so, and will argue that this due to a lack of solid and generally agreed-upon foundations of AI. The field of AI soon abandoned its efforts of rectifying this state of affairs, and pessimists even created a defense mechanism denying the possibility or usefulness of a (simple) formal theory of general intelligence. While human intelligence might indeed be messy and unintelligible, I will argue that a simple formal definition of machine intelligence *is* possible and useful. I will discuss how this definition fits into the various important dimensions of research on (artificial) intelligence including human \leftrightarrow rational, thinking \leftrightarrow acting, top-down \leftrightarrow bottom-up, the agent framework, traits of intelligence, deduction \leftrightarrow induction, and learning \leftrightarrow planning.

The problem. I define *the AI problem* to mean the problem of building systems that possess general, rather than specific, intelligence in the sense of being able to solve a wide range of problems generally regarded to require human-level intelligence.

Optimists believe that the AI problem can be solved within a couple of decades [36]. Pessimists deny its principle feasibility on religious, philosophical, mathematical, or technical grounds (see [56, Chp. 26] for a list of arguments). Optimists have refuted/rebutted all those arguments (see [6, Chp. 9] and [38]), but haven't produced super-human AI either, so the issue remains unsettled.

One problem in AI, and I will argue key problem, is that there is no general agreement on what intelligence is. This has led to endless circular and often fruitless arguments, and has held up progress. Generally, the lack of a generally-accepted solid foundation makes high card houses fold easily. Compare this with Russell's paradox which shattered the

foundations of mathematics, and which was finally resolved by the completely formal and generally agreed-upon ZF(C) theory of sets.

On the other hand, it is an anomaly that nowadays most AI researchers avoid discussing or formalizing intelligence, which is caused by several factors: It is a difficult old subject, it is politically charged, it is not necessary for narrow AI which focusses on specific applications, AI research is done primarily by computer scientists who mainly care about algorithms rather than philosophical foundations, and the popular belief that general intelligence is principally unamenable to a mathematical definition. These reasons explain but only partially justify the limited effort in trying to formalize general intelligence. There is no convincing argument that this is impossible.

Assume we had a formal, objective, non-anthropocentric, and direct definition, measure, and/or test of intelligence, or at least a very general intelligence-resembling formalism that could serve as an adequate substitute. This would bring the higher goals of the field into tight focus and allow us to objectively and rigorously compare different approaches and judge the overall progress. Formalizing and rigorously defining a previously vague concept usually constitutes a quantum leap forward in the field: Cf. the history of sets, numbers, logic, fluxions/infinitesimals, energy, infinity, temperature, space, time, observer, etc.

Is a simple *formal definition of intelligence* possible? Isn't intelligence a too complex and anthropocentric phenomenon to allow formalization? Likely not: There are very simple models of chaotic phenomena such as turbulence. Think about the simple iterative map $z \rightarrow z^2 + c$ that produces the amazingly rich, fractal landscape, sophisticated versions of it used to produce images of virtual ecosystems as in the movie Avatar. Or the complexity of (bio)chemistry emerges out of the elegant mathematical theory Quantum Electro Dynamics.

Modeling human intelligence is probably going to be messy, but ideal rational behavior seems to capture the essence of intelligence, and, as I claim, can indeed be completely formalized. Even if there is no unique definition capturing all aspects we want to include in a definition of intelligence, or if some aspects are forever beyond formalization (maybe consciousness and qualia), pushing the frontier and studying the best available formal proxy is of utmost importance for understanding artificial and natural minds.

Context. There are many fields that try to understand the phenomenon of intelligence and whose insights help in creating intelligent systems: *cognitive psychology* [67] and *behaviorism* [64], *philosophy of mind* [7, 61], *neuroscience* [18], *linguistics* [8, 17], *an-*

thropology [51], *machine learning* [5, 74], *logic* [44, 77], *computer science* [56], *biological evolution* [33, 75], *economics* [46], and *others*.

Cognitive science studies how humans think, Behaviorism and the Turing test how humans act, the laws of thought define rational thinking, while AI research increasingly focusses on systems that act rationally.

What is AI?	Thinking	Acting
humanly	Cognitive Science	Turing test, Behaviorism
rationally	Laws of Thought	Doing the Right Thing

In computer science, most AI research is *bottom-up*; extending and improving existing or developing new *algorithms* and increasing their range of applicability; an interplay between experimentation on toy problems and theory, with occasional real-world applications. A *top-down* approach would start from a general principle and derive effective approximations (like heuristic approximations to minimax tree search). Maybe when the top-down and bottom-up approaches meet in the middle, we will have arrived at practical truly intelligent machines.

The science of artificial intelligence may be defined as the construction of intelligent systems (*artificial agents*) and their analysis. A natural definition of a *system* is anything that has an input and an output stream, or equivalently an agent that acts and observes. This agent perspective of AI [56] brings some order and unification into the large variety of problems the fields wants to address, but it is only a framework rather than providing a complete theory of intelligence. In the absence of a perfect (stochastic) model of the environment the agent interacts with, *machine learning* techniques are needed and employed to learn from experience. There is no general theory for learning agents (apart from UAI). This has resulted in an ever increasing number of *limited models and algorithms* in the past.

What distinguishes an *intelligent* system from a non-intelligent one? *Intelligence* can have many faces like *reasoning*, *creativity*, *association*, *generalization*, *pattern recognition*, *problem solving*, *memorization*, *planning*, *achieving goals*, *learning*, *optimization*, *self-preservation*, *vision*, *language processing*, *classification*, *induction*, *deduction*, and *knowledge acquisition and processing*. A formal definition incorporating every aspect of intelligence, however, *seems* difficult.

There is no lack of attempts to characterize or define intelligence trying to capture all traits *informally* [39]. One of the more successful characterizations is: *Intelligence measures an agents ability to perform well in a large range of environments* [40]. Most traits of intelligence are implicit in and emergent from this definition as these capacities

enable an agent to succeed [38]. Convincing formal definitions other than the ones spawned by UAI are essentially lacking.

Another important dichotomy is whether an approach focusses (more) on deduction or induction. Traditional AI concentrates mostly on the logical deductive reasoning aspect, while machine learning focusses on the inductive inference aspect. Learning and hence induction are indispensable traits of any AGI. Regrettably, induction is peripheral to traditional AI, and the machine learning community in large is not interested in A(G)I. It is the field of reinforcement learning at the intersection of AI and machine learning that has AGI ambitions *and* takes learning seriously.

UAI in perspective. The theory of Universal Artificial Intelligence developed in the last decade is a modern information-theoretic, inductive, reinforcement learning approach to AGI that has been investigated in great technical depth [24].

Like traditional AI, UAI is concerned with agents *doing the right thing*, but is otherwise quite different: It is a *top-down* approach in the sense that it starts with a single completely *formal general* definition of intelligence from which an essentially *unique agent* that seems to possess all *traits* of rational intelligence is derived. It is not just another framework with some gaps to be filled in later, since the agent is *completely* defined.

It also takes induction very seriously: Universal learning is one of the agent's two key elements (the other is stochastic planning). Indeed, logic and deduction play no fundamental role in UAI (but are emergent). This also naturally dissolves Lucas' and Penrose' [52] argument against AGI that Goedel's incompleteness result shows that the human mind is not a computer. The fallacy is to assume that the mind (human and machine alike) are infallible deductive machines.

The status of UAI might be compared to Super String theory in physics. Both are currently the most promising candidates for a grand unification (of AI and physics, respectively), although there are also marked differences. Like the unification hierarchy of physical theories allows relating and regarding the myriad of limited models as effective approximations, UAI allows us to regard existing approaches to AI as effective approximations. Understanding AI in this way gives researchers a much more coherent view of the field.

Indeed, UAI seems to be the first sound and complete mathematical theory of (rational) intelligence. The next section presents a very brief introduction to UAI from [29], together with an informal explanation of what the previous sentence actually means. See [24] for formal definitions and results.

5.3 Universal Artificial Intelligence

This section describes the theory of Universal Artificial Intelligence (UAI), a modern information-theoretic approach to AI, which differs essentially from mainstream A(G)I research described in the previous sections. The connection of UAI to other research fields and the philosophical and technical ingredients of UAI (Ockham, Epicurus, Turing, Bayes, Solomonoff, Kolmogorov, Bellman) are briefly discussed. The UAI-based universal intelligence measure and order relation in turn define the (w.r.t. this measure) most intelligent agent AIXI, which seems to be the first sound and complete theory of a universal optimal rational agent embedded in an arbitrary computable but unknown environment with reinforcement feedback. The final paragraph clarifies what this actually means.

Defining Intelligence. Philosophers, AI researchers, psychologists, and others have suggested many informal=verbal definitions of intelligence [39], but there is not too much work on formal definitions that are broad, objective, and non-anthropocentric. See [40] for a comprehensive collection, discussion and comparison of intelligence definitions, tests, and measures with all relevant references. It is beyond the scope of this article to discuss them.

Intelligence is graded, since agents can be more or less intelligent. Therefore it is more natural to consider measures of intelligence, rather than binary definitions which would classify agents as intelligent or not based on an (arbitrary) threshold. This is exactly what UAI provides: A formal, broad, objective, universal measure of intelligence [40], which formalizes the verbal characterization stated in the previous section. Agents can be more or less intelligent w.r.t. this measure and hence can be sorted w.r.t. their intelligence [24, Sec. 5.1.4]. One can show that there is an agent, coined AIXI, that maximizes this measure, which could therefore be called the most intelligent agent.

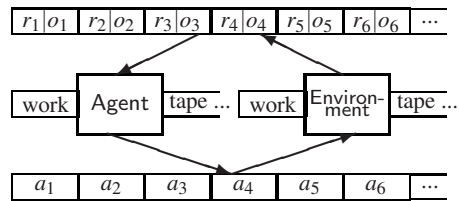
I will not present the UAI-based intelligence measure [40] and order relation [24] here, but, after listing the conceptual ingredients to UAI and AIXI, directly proceed to defining and discussing AIXI.

UAI and AIXI ingredients [29]. The theory of UAI has interconnections with (draws from and contributes to) many research fields, encompassing computer science (artificial intelligence, machine learning, computation), engineering (information theory, adaptive control), economics (rational agents, game theory), mathematics (statistics, probability), psychology (behaviorism, motivation, incentives), and philosophy (inductive inference, theory of knowledge). The concrete ingredients in AIXI are as follows: Intelligent *actions* are based

on informed *decisions*. Attaining good decisions requires *predictions* which are typically based on models of the environments. Models are constructed or learned from past observations via *induction*. Fortunately, based on the *deep philosophical insights* and *powerful mathematical developments*, all these problems have been overcome, at least in theory: So what do we need (from a mathematical point of view) to construct a universal optimal learning agent interacting with an arbitrary unknown environment? The theory, coined *UAI*, developed in the last decade and explained in [24] says: *All you need is Ockham, Epicurus, Turing, Bayes, Solomonoff* [65], *Kolmogorov* [35], and *Bellman* [1]: Sequential decision theory [4] (*Bellman's equation*) formally solves the problem of rational agents in uncertain worlds if the true environmental probability distribution is known. If the environment is unknown, *Bayesians* [2] replace the true distribution by a weighted mixture of distributions from some (hypothesis) class. Using the large class of all (semi)measures that are (semi)computable on a *Turing* machine bears in mind *Epicurus*, who teaches not to discard any (consistent) hypothesis. In order not to ignore *Ockham*, who would select the simplest hypothesis, *Solomonoff* defined a universal prior that assigns high/low prior weight to simple/complex environments [54], where *Kolmogorov* quantifies complexity [43]. Their unification constitutes the theory of *UAI* and resulted in the universal intelligence measure and order relation and the following model/agent *AIXI*.

The AIXI Model in one line [29]. It is possible to write down the *AIXI* model explicitly in one line, although *one should not expect to be able to grasp the full meaning and power from this compact and somewhat simplified representation*.

AIXI is an agent that interacts with an environment in cycles $k = 1, 2, \dots, m$. In cycle k , *AIXI* takes action a_k (e.g. a limb movement) based on past perceptions $o_1 r_1 \dots o_{k-1} r_{k-1}$ as defined below. Thereafter, the environment provides a (regular) observation o_k (e.g. a camera image) to *AIXI* and a real-valued reward r_k . The reward can be very scarce, e.g. just +1 (-1) for winning (losing) a chess game, and 0 at all other times. Then the next cycle $k + 1$ starts. This agent-environment interaction protocol can be depicted as on the right. Given the interaction protocol above, the simplest version of *AIXI* is defined by



$$\text{AIXI} \quad a_k := \arg \max_{a_k} \sum_{o_k r_k} \dots \max_{a_m} \sum_{o_m r_m} [r_k + \dots + r_m] \sum_{q: U(q, a_1 \dots a_m) = o_1 r_1 \dots o_m r_m} 2^{-\ell(q)}$$

The expression shows that AIXI tries to maximize its total future reward $r_k + \dots + r_m$. If the environment is modeled by a deterministic program q , then the future perceptions $\dots o_k r_k \dots o_m r_m = U(q, a_1 \dots a_m)$ can be computed, where U is a universal (monotone Turing) machine executing q given $a_1 \dots a_m$. Since q is unknown, AIXI has to maximize its expected reward, i.e. average $r_k + \dots + r_m$ over all possible future perceptions created by all possible environments q that are consistent with past perceptions. The simpler an environment, the higher is its a-priori contribution $2^{-\ell(q)}$, where simplicity is measured by the length ℓ of program q . AIXI effectively learns by eliminating Turing machines q once they become inconsistent with the progressing history. Since noisy environments are just mixtures of deterministic environments, they are automatically included [54, Sec. 7.2], [82]. The sums in the formula constitute the averaging process. Averaging and maximization have to be performed in chronological order, hence the interleaving of max and Σ (similarly to minimax for games).

One can fix any finite action and perception space, any reasonable U , and any large finite lifetime m . This completely and uniquely defines AIXI's actions a_k , which are limit-computable via the expression above (all quantities are known).

Discussion. The AIXI model seems to be the first sound and complete *theory* of a universal optimal rational agent embedded in an arbitrary computable but unknown environment with reinforcement feedback. AIXI is *universal* in the sense that it is designed to be able to interact with any (deterministic or stochastic) computable environment; the universal Turing machines on which it is based is crucially responsible for this. AIXI is *complete* in the sense that it is not an incomplete framework or partial specification (like Bayesian statistics which leaves open the choice of the prior or the rational agent framework or the subjective expected utility principle) but is completely and essentially uniquely defined. AIXI is *sound* in the sense of being (by construction) free of any internal contradictions (unlike e.g. in knowledge-based deductive reasoning systems where avoiding inconsistencies can be very challenging). AIXI is *optimal* in the senses that: no other agent can perform uniformly better or equal in all environments, it is a unification of two optimal theories themselves, a variant is self-optimizing; and it is likely also optimal in other/stronger senses. AIXI is *rational* in the sense of trying to maximize its future long-term reward. For the reasons above I have argued that AIXI is a mathematical "solution" of the AI problem: AIXI would be able to learn any learnable task and likely better so than any other unbiased agent, but AIXI is more a *theory* or formal definition rather than an algorithm, since it is only limit-computable. How can an equation that fits into a single line capture the diver-

sity, complexity, and essence of (rational) intelligence? We know that complex appearing phenomena such as chaos and fractals can have simple descriptions such as iterative maps and the complexity of chemistry emerges from simple physical laws. There is no a-priori reason why ideal rational intelligent behavior should not also have a simple description, with most traits of intelligence being emergent. Indeed, even an axiomatic characterization seems possible [72, 73].

5.4 Facets of Intelligence

Intelligence can have many faces. I will argue in this section that the AIXI model possesses all or at least most properties an intelligent rational agent should possess. Some facets have already been formalized, some are essentially built-in, but the majority have to be emergent. Some of the claims have been proven in [24] but the majority has yet to be addressed.

Generalization is essentially inductive inference [54]. **Induction** is the process of inferring general laws or models from observations or data by finding regularities in past/other data. This trait is a fundamental cornerstone of intelligence.

Prediction is concerned with forecasting future observations (often based on models of the world learned) from past observations. Solomonoff's theory of prediction [65, 66] is a universally optimal solution of the prediction problem [26, 54]. Since it is a key ingredient in the AIXI model, it is natural to expect that AIXI is an optimal predictor if rewarded for correct predictions. Curiously only weak and limited rigorous results could be proven so far [24, Sec. 6.2].

Pattern recognition, abstractly speaking, is concerned with classifying data (patterns). This requires a similarity measure between patterns. Supervised **classification** can essentially be reduced to a sequence prediction problem, hence formally pattern recognition reduces to the previous item, although interesting questions specific to classification emerge [24, Chp. 3].

Association. Two stimuli or observations are associated if there exists some (cor)relation between them. A set of observations can often be **clustered** into different categories of similar=associated items. For AGI, a *universal* similarity measure is required. Kolmogorov complexity via the universal similarity metric [9] can provide such a measure, but many fundamental questions have yet to be explored: How does association function in AIXI?

How can Kolmogorov complexity well-define the (inherently? so far?) ill-defined clustering problem?

Reasoning is arguably the most prominent trait of human intelligence. Interestingly deductive reasoning and logic are *not* part of the AIXI architecture. The fundamental assumption is that there is no sure knowledge of the world, all inference is tentative and inductive, and that logic and *deduction* constitute an idealized limit applicable in situations where uncertainties are extremely small, i.e. probabilities are extremely close to 1 or 0. What would be very interesting to show is that *logic* is an emergent phenomenon, i.e. that AIXI learns to reason logically if/since this helps collect reward.

Problem solving might be defined as goal-oriented reasoning, and hence reduces to the previous item, since AIXI is designed to *achieve goals* (which is reward maximization in the special case of a terminal reward when the goal is achieved). Problems can be of very different nature, and some of the other traits of intelligence can be regarded as instances of problem solving, e.g. planning.

Planning ability is directly incorporated in AIXI via the alternating maximization and summation in the definition. Algorithmically AIXI plans through its entire life via a deep expectimax tree search up to its death, based on its belief about the world. In known constrained domains this search corresponds to classical exact planning strategies as e.g. exemplified in [24, Chp. 6].

Creativity is the ability to generate innovative ideas and to manifest these into reality. Creative people are often more successful than unimaginative ones. Since AIXI is the ultimate success-driven agent, AIXI should be highly creative, but this has yet to be formalized and proven, or at least exemplified.

Knowledge. AIXI stores the entire interaction history and has perfect *memory*. Additionally, models of the experienced world are constructed (learned) from this *information* in form of short(est) programs. These models guide AIXI's behavior, so constitute knowledge for AIXI. Any *ontology* is implicit in these programs. How short-term, long-term, relational, hierarchical, etc. memory emerges out of this compression-based approach has not yet been explored.

Actions influence the environment which reacts back to the agent. *Decisions* can have long-term consequences, which the expectimax planner of AIXI should properly take into account. Particular issues of concern are the interplay of learning and planning (the in-

famous exploration↔exploitation tradeoff [37]). Additional complications that arise from embodied agents will be considered in the next section.

Learning. There are many different forms of learning: supervised, unsupervised, semi-supervised, reinforcement, transfer, associative, transductive, prequential, and many others. By design, AIXI is a reinforcement learner, but one can show that it will also “listen” to an informative teacher, i.e. it *learns* to learn supervised [24, Sec. 6.5]. It is plausible that AIXI can also acquire the other learning techniques.

Self-awareness allows one to (meta)reason about one’s own thoughts, which is an important trait of higher intelligence, in particularly when interacting with other forms of intelligence. Technically all what might be needed is that an agent has and exploits not only a model of the world but also a model of itself including aspects of its own algorithm, and this recursively. Is AIXI self-aware in this technical sense?

Consciousness is possibly the most mysterious trait of the human mind. Whether anything rigorous can ever be said about the consciousness of AIXI or AIs in general is not clear and in any case beyond my expertise. I leave this to philosophers of the mind [7] like the world-renowned expert on (the hard problem of) consciousness, David Chalmers [6].

5.5 Social Questions

Consider now a sophisticated physical humanoid robot like Honda’s ASIMO but equipped with an AIXI brain. The observations o_k consist of camera image, microphone signal, and other sensory input. The actions a_k consist of controlling mainly a loud speaker and motors for limbs, but possibly other internal functions it has direct control over. The reward r_k should be some combination of its own “well-being” (e.g. proportional to its battery level and condition of its body parts) and external reward/punishment from some “teacher(s)”.

Imagine now what happens if this AIXI-robot is let loose in our society. Many questions deserving attention arise, and some are imperative to be rigorously investigated before risking this experiment.

Children of higher animals require extensive nurturing in a safe environment because they lack sufficient innate skills for survival in the real world, but are compensated for their ability to learn to perform well in a large range of environments. AIXI is at the extreme of being “born” with essentially no knowledge about our world, but a universal “brain” for

learning and planning in any environment where this is possible. As such, it also requires a guiding teacher initially. Otherwise it would simply run out of battery.

AIXI has to learn *vision*, *language*, and *motor skills* from scratch, similarly to higher animals and machine learning algorithms, but more extreme/general. Indeed, Solomonoff [65] already showed how his system can learn grammar from positive instances only, but much remains to be done. Appropriate *training sequences* and *reward shaping* in this early “childhood” phase of AIXI are important. AIXI can learn from rather crude teachers as long as the reward is biased in the ‘right’ direction. The answers to many of the following questions likely depend on the upbringing of AIXI:

- **Schooling:** Will a pure reward maximizer such as AIXI listen to and trust a teacher and learn to learn supervised (=faster)? Yes [24, Sec. 6.5].
- Take **Drugs** (hacking the reward system): Likely no, since long-term reward would be small (death), but see [55].
- **Replication or procreation:** Likely yes, if AIXI believes that clones or descendants are useful for its own goals.
- **Suicide:** Likely yes (no), if AIXI is raised to believe to go to heaven (hell) i.e. maximal (minimal) reward forever.
- **Self-Improvement:** Likely yes, since this helps to increase reward.
- **Manipulation:** Manipulate or threaten teacher to give more reward.
- **Attitude:** Are pure reward maximizers egoists, *psychopaths*, and/or killers or will they be *friendly* (*altruism* as extended *ego(t)ism*)?
- **Curiosity** killed the cat and maybe AIXI, or is extra reward for curiosity necessary [48, 60]?
- **Immortality** can cause laziness [24, Sec. 5.7]!
- Can **self-preservation** be learned or need (parts of) it be innate.
- **Socializing:** How will AIXI interact with another AIXI [29, Sec. 5j], [53]?

A partial discussion of some of these questions can be found in [24] but many are essentially unexplored. Point is that since AIXI is completely formal, it permits to formalize these questions and to mathematically analyze them. That is, UAI has the potential to arrive at definite answers to various questions regarding the social behavior of super-intelligences. Some formalizations and semi-formal answers have recently appeared in the award-winning papers [49, 55].

5.6 State of the Art

This section describes the technical achievements of UAI to date. Some remarkable and surprising results have already been obtained. Various theoretical consistency and optimality results for AIXI have been proven, although stronger results would be desirable. On the other hand, the special case of universal induction and prediction in non-reactive environments is essentially closed. From the practical side, various computable approximations of AIXI have been developed, with the latest MC-AIXI-CTW incarnation exhibiting impressive performance. Practical approximations of the universal intelligence measure have also been used to test and consistently order systems of limited intelligence. Some other related work such as the compression contest is also briefly mentioned, and references to some more practical but less general work such as feature reinforcement learning are given.

Theory of UAI. Forceful theoretical arguments that AIXI is the most intelligent general-purpose agent incorporating all aspects of rational intelligence have been put forward, supported by partial proofs. For this, results of many fields had to be pulled together or developed in the first place: *Kolmogorov complexity* [43], *information theory* [10], *sequential decision theory* [4], *reinforcement learning* [74], *artificial intelligence* [56], *Bayesian statistics* [3], *universal induction* [54], and *rational agents* [62]. Various notions of optimality have been considered. The difficulty is coming up with sufficiently strong but still satisfiable notions. Some are weaker than desirable, others are too strong for any agent to achieve. What has been shown thus far is that AIXI learns the correct predictive model [24], is Pareto optimal in the sense that no other agent can perform uniformly better or equal in all environments, and a variant is self-optimizing in the sense that asymptotically the accumulated reward is as high as possible, i.e. the same as the maximal reward achievable by a completely informed agent [23]. AIXI is likely also optimal in other/stronger senses. An axiomatic characterization has also been developed [72, 73].

The induction problem. The induction problem is a fundamental problem in philosophy [12, 54] and science [15, 32, 80], and a key sub-component of UAI. Classical open problems around induction are the zero prior problem and the confirmation of (universal) hypotheses in general and the Black ravens paradox in particular, reparametrization invariance, the old-evidence problem and ad-hoc hypotheses, and the updating problem [12]. In a series of papers (see [26] for references) it has been shown that Solomonoff's theory of universal induction essentially solves or circumvents all these problems [54]. It is also predictively optimal and has minimal regret for arbitrary loss functions.

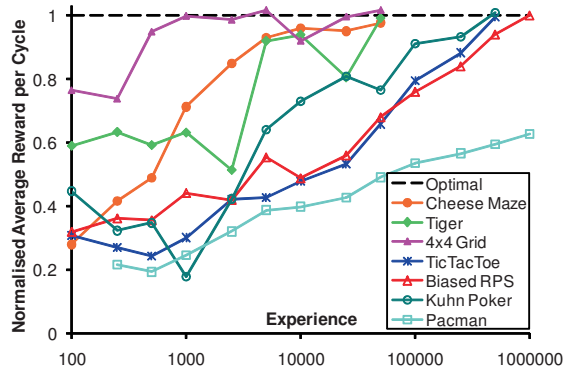
It is fair to say that Solomonoff's theory serves as an adequate mathematical/theoretical foundation of induction [54], machine learning [30], and component of UAI [24].

Computable approximations of AIXI. An early critique of UAI was that AIXI is incomputable. The down-scaled still provably optimal $AIXI_{\epsilon}$ model [24, Chp. 7] based on universal search algorithms [16, 22, 42] was still computationally intractable. The Optimal Ordered Problem Solver [59] was the first practical implementation of universal search and has been able to solve open learning tasks such as Towers-of-Hanoi for arbitrary number of disks, robotic behavior, and others.

For repeated 2×2 matrix games such as the Prisoner's dilemma, a direct brute-force approximation of AIXI is computationally tractable. Despite these domains being tiny, they raise notoriously difficult questions [62]. The experimental results confirmed the theoretical optimality claims of AIXI [53], as far as limited experiments are able to do so.

A Monte-Carlo approximation of AIXI has been proposed in [50] that samples programs according to their algorithmic probability as a way of approximating Solomonoff's universal a-priori probability, similar to sampling from the speed prior [58].

The most powerful systematic approximation, implementation, and application of AIXI so far is the MC-AIXI-CTW algorithm [78]. It combines award-winning ideas from universal Bayesian data compression [81] and the recent highly successful (in computer Go) upper confidence bound algorithm for expect-



timax tree search [34]. For the first time, without any domain knowledge, the same agent is able to self-adapt to a diverse range of environments. For instance, AIXI, is able to *learn* from scratch how to play TicTacToe, Pacman, Kuhn Poker, and other games by trial and error without even providing the rules of the games [79].

Measures/tests/definitions of intelligence. The history of informal definitions and measures of intelligence [39] and anthropocentric tests of intelligence [76] is long and old. In the last decade various formal definitions, measures and tests have been suggested: Solomonoff induction and Kolmogorov complexity inspired the universal C-test [19, 21], while AIXI inspired an extremely general, objective, fundamental, and

formal intelligence order relation [24] and a universal intelligence measure [38, 40], which have already attracted the popular scientific press [14] and received the SIAI award. Practical instantiations thereof [20, 41] also received quite some media attention (<http://users.dsic.upv.es/proy/anynt/>).

Less related/general work. There is of course other less related, less general work, similar in spirit to or with similar aims as UAI/AIXI, e.g. UTree [45], URL [13], PORL [69, 70], FOMDP [57], FacMDP [68], PSR [63], POMDP [11], and others. The feature reinforcement learning approach also belongs to this category [27, 28, 47, 71].

Compression contest. The ongoing Human Knowledge Compression Contest [25] is another outgrowth of UAI. The contest is motivated by the fact that being able to compress well is closely related to being able to predict well and ultimately to act intelligently, thus reducing the slippery concept of intelligence to hard file size numbers. Technically it is a community project to approximate Kolmogorov complexity on real-world textual data. In order to compress data, one has to find regularities in them, which is intrinsically difficult (many researchers live from analyzing data and finding compact models). So compressors better than the current “dumb” compressors need to be smart(er). Since the prize wants to stimulate the development of “universally” smart compressors, a “universal” corpus of data has been chosen. Arguably the online encyclopedia Wikipedia is a good snapshot of the Human World Knowledge. So the ultimate compressor of it should “understand” all human knowledge, i.e. be really smart. The contest is meant to be a cost-effective way of motivating researchers to spend time towards achieving AGI via the promising and quantitative path of compression. The competition raised considerable attention when launched, but to retain attention the prize money should be increased (sponsors are welcome), and the setup needs some adaptation.

5.7 Discussion

Formalizing and answering deep philosophical questions. UAI deepens our understanding of artificial (and to a limited extent human) intelligence; in particular which and how facets of intelligence can be understood as emergent phenomena of goal- or reward-driven actions in unknown environments. UAI allows a more quantitative and rigorous discussion of various philosophical questions around intelligence, and ultimately settling these questions. This can and partly has been done by formalizing the philosophical concepts related to intelligence under consideration, and by studying them mathematically. Formal

definitions may not perfectly or not one-to-one or not uniquely correspond to their intuitive counterparts, but in this case alternative formalizations allow comparison and selection. In this way it might even be possible to rigorously answer various social and ethical questions: whether a super rational intelligence such as AIXI will be benign to humans and/or its ilk, or behave psychopathically and kill or enslave humans, or be insane and e.g. commit suicide.

Building more intelligent agents. From a practical point of building intelligent agents, since AIXI is incomputable or more precisely only limit-computable, it has to be approximated in practice. The results achieved with the MC-AIXI-CTW approximation are only the beginning. As outlined in [79], many variations and extensions are possible, in particular to incorporate long-term memory and smarter planning heuristics. The same single MC-AIXI-CTW agent is already able to learn to play TicTacToe, Kuhn Poker, and most impressively Pacman [79] from scratch. Besides Pacman, there are hundreds of other arcade games from the 1980s, and it would be sensational if a single algorithm could learn them all solely by trial and error, which seems feasible for (a variant of) MC-AIXI-CTW. While these are “just” recreational games, they *do* contain many prototypical elements of the real world, such as food, enemies, friends, space, obstacles, objects, and weapons. Next could be a test in modern virtual worlds (e.g. bots for VR/role games or intelligent software agents for the internet) that require intelligent agents, and finally some selected real-world problems.

Epilogue. It is virtually impossible to predict the future rate of progress but past progress on UAI makes me confident that UAI as a whole will continually progress. By providing rigorous foundations to AI, I believe that UAI will also speed up progress in the field of A(G)I in general. In any case, UAI is a very useful educational tool with AIXI being a gold standard for intelligent agents which other practical general purpose AI programs should aim for.

Bibliography

- [1] R. E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.
- [2] J. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer, Berlin, 3rd edition, 1993.
- [3] J. Berger. The case for objective Bayesian analysis. *Bayesian Analysis*, 1(3):385–402, 2006.
- [4] D. P. Bertsekas. *Dynamic Programming and Optimal Control, volume I and II*. Athena Scientific, Belmont, MA, 3rd edition, 2006. Volumes 1 and 2.
- [5] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

- [6] D. J. Chalmers. *The Conscious Mind*. Oxford University Press, USA, 1996.
- [7] D. J. Chalmers, editor. *Philosophy of Mind: Classical and Contemporary Readings*. Oxford University Press, USA, 2002.
- [8] N. Chomsky. *Language and Mind*. Cambridge University Press, 3rd edition, 2006.
- [9] R. Cilibrasi and P. M. B. Vitányi. Clustering by compression. *IEEE Trans. Information Theory*, 51(4):1523–1545, 2005.
- [10] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition, 2006.
- [11] F. Doshi-Velez. The infinite partially observable markov decision process. In *Proc. 22nd Conf. on Neural Information Processing Systems 22 (NIPS'09)*, 2009.
- [12] J. Earman. *Bayes or Bust? A Critical Examination of Bayesian Confirmation Theory*. MIT Press, Cambridge, MA, 1993.
- [13] V. Farias, C. C. Moallemi, B. V. Roy, and T. Weissman. Universal reinforcement learning. *IEEE Transactions on Information Theory*, 56(5):2441–2454, 2010.
- [14] C. Fiévet. Mesurer l'intelligence d'une machine. In *Le Monde de l'intelligence*, volume 1, pages 42–45, Paris, November 2005. Mondeo publishing.
- [15] D. M. Gabbay, S. Hartmann, and J. Woods, editors. *Handbook of Inductive Logic*. North Holland, 2011.
- [16] M. Gaglio. Universal search. *Scholarpedia*, 2(11):2575, 2007.
- [17] R. Hausser. *Foundations of Computational Linguistics: Human-Computer Communication in Natural Language*. Springer, 2nd edition, 2001.
- [18] J. Hawkins and S. Blakeslee. *On Intelligence*. Times Books, 2004.
- [19] J. Hernández-Orallo. On the computational measurement of intelligence factors. In *Performance Metrics for Intelligent Systems Workshop*, pages 1–8, Gaithersburg, MD, USA, 2000.
- [20] J. Hernandez-Orallo and D. L. Dowe. Measuring universal intelligence: Towards an anytime intelligence test. *Artificial Intelligence*, 174(18):1508–1539, 2010.
- [21] J. Hernández-Orallo and N. Minaya-Collado. A formal definition of intelligence based on an intensional variant of kolmogorov complexity. In *International Symposium of Engineering of Intelligent Systems*, pages 146–163, 1998.
- [22] M. Hutter. The fastest and shortest algorithm for all well-defined problems. *International Journal of Foundations of Computer Science*, 13(3):431–443, 2002.
- [23] M. Hutter. Self-optimizing and Pareto-optimal policies in general environments based on Bayes-mixtures. In *Proc. 15th Annual Conf. on Computational Learning Theory (COLT'02)*, volume 2375 of *LNAI*, pages 364–379, Sydney, 2002. Springer, Berlin.
- [24] M. Hutter. *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*. Springer, Berlin, 2005.
- [25] M. Hutter. Human knowledge compression prize, 2006. open ended, <http://prize.hutter1.net/>.
- [26] M. Hutter. On universal prediction and Bayesian confirmation. *Theoretical Computer Science*, 384(1):33–48, 2007.
- [27] M. Hutter. Feature dynamic Bayesian networks. In *Proc. 2nd Conf. on Artificial General Intelligence (AGI'09)*, volume 8, pages 67–73. Atlantis Press, 2009.
- [28] M. Hutter. Feature reinforcement learning: Part I: Unstructured MDPs. *Journal of Artificial General Intelligence*, 1:3–24, 2009.
- [29] M. Hutter. Open problems in universal induction & intelligence. *Algorithms*, 3(2):879–906, 2009.
- [30] M. Hutter. Universal learning theory. In C. Sammut and G. Webb, editors, *Encyclopedia of Machine Learning*, pages 1001–1008. Springer, 2011.
- [31] M. Hutter. Can intelligence explode? *Journal of Consciousness Studies*, 19(1-2), 143–166, 2012.

- [32] E. T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, Cambridge, MA, 2003.
- [33] K. V. Kardong. *An Introduction to Biological Evolution*. McGraw-Hill Science / Engineering / Math, 2nd edition, 2007.
- [34] L. Kocsis and C. Szepesvári. Bandit based Monte-Carlo planning. In *Proc. 17th European Conf. on Machine Learning (ECML'06)*, pages 282–293, 2006.
- [35] A. N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of Information and Transmission*, 1(1):1–7, 1965.
- [36] R. Kurzweil. *The Singularity Is Near*. Viking, 2005.
- [37] T. Lattimore and M. Hutter. Asymptotically optimal agents. In *Proc. 22nd International Conf. on Algorithmic Learning Theory (ALT'11)*, volume 6925 of *LNAI*, pages 368–382, Espoo, Finland, 2011. Springer, Berlin.
- [38] S. Legg. *Machine Super Intelligence*. PhD thesis, IDSIA, Lugano, Switzerland, 2008. Recipient of the \$10'000,- Singularity Prize/Award.
- [39] S. Legg and M. Hutter. A collection of definitions of intelligence. In *Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms*, volume 157 of *Frontiers in Artificial Intelligence and Applications*, pages 17–24, Amsterdam, NL, 2007. IOS Press.
- [40] S. Legg and M. Hutter. Universal intelligence: A definition of machine intelligence. *Minds & Machines*, 17(4):391–444, 2007.
- [41] S. Legg and J. Veness. An approximation of the universal intelligence measure. In *Proc. Solomonoff 85th Memorial Conference, LNAI*, Melbourne, Australia, 2011. Springer.
- [42] L. A. Levin. Universal sequential search problems. *Problems of Information Transmission*, 9:265–266, 1973.
- [43] M. Li and P. M. B. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications*. Springer, Berlin, 3rd edition, 2008.
- [44] J. W. Lloyd. *Foundations of Logic Programming*. Springer, 2nd edition, 1987.
- [45] A. K. McCallum. *Reinforcement Learning with Selective Perception and Hidden State*. PhD thesis, Department of Computer Science, University of Rochester, 1996.
- [46] R. B. McKenzie. *Predictably Rational? In Search of Defenses for Rational Behavior in Economics*. Springer, 2009.
- [47] P. Nguyen, P. Sunehag, and M. Hutter. Feature reinforcement learning in practice. In *Proc. 9th European Workshop on Reinforcement Learning (EWRL-9)*, volume 7188 of *LNAI*, Springer, 2011. to appear.
- [48] L. Orseau. Optimality issues of universal greedy agents with static priors. In *Proc. 21st International Conf. on Algorithmic Learning Theory (ALT'10)*, volume 6331 of *LNAI*, pages 345–359, Canberra, 2010. Springer, Berlin.
- [49] L. Orseau and M. Ring. Self-modification and mortality in artificial agents. In *Proc. 4th Conf. on Artificial General Intelligence (AGI'11)*, volume 6830 of *LNAI*, pages 1–10. Springer, Berlin, 2011.
- [50] S. Pankov. A computational approximation to the AIXI model. In *Proc. 1st Conference on Artificial General Intelligence*, volume 171, pages 256–267, 2008.
- [51] M. A. Park. *Introducing Anthropology: An Integrated Approach*. McGraw-Hill, 4th edition, 2007.
- [52] R. Penrose. *Shadows of the Mind, A Search for the Missing Science of Consciousness*. Oxford University Press, 1994.
- [53] J. Poland and M. Hutter. Universal learning of repeated matrix games. In *Proc. 15th Annual Machine Learning Conf. of Belgium and The Netherlands (Benelearn'06)*, pages 7–14, Ghent, 2006.
- [54] S. Rathmanner and M. Hutter. A philosophical treatise of universal induction. *Entropy*, 13(6):1076–1136, 2011.

- [55] M. Ring and L. Orseau. Delusion, survival, and intelligent agents. In *Proc. 4th Conf. on Artificial General Intelligence (AGI'11)*, volume 6830 of *LNAI*, pages 11–20. Springer, Berlin, 2011.
- [56] S. J. Russell and P. Norvig. *Artificial Intelligence. A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, 3rd edition, 2010.
- [57] S. Sanner and C. Boutilier. Practical solution techniques for first-order MDPs. *Artificial Intelligence*, 173(5–6):748–788, 2009.
- [58] J. Schmidhuber. The speed prior: A new simplicity measure yielding near-optimal computable predictions. In *Proc. 15th Conf. on Computational Learning Theory (COLT'02)*, volume 2375 of *LNAI*, pages 216–228, Sydney, 2002. Springer, Berlin.
- [59] J. Schmidhuber. Optimal ordered problem solver. *Machine Learning*, 54(3):211–254, 2004.
- [60] J. Schmidhuber. Simple algorithmic principles of discovery, subjective beauty, selective attention, curiosity & creativity. In *Proc. 10th Intl. Conf. on Discovery Science (DS'07)*, volume *LNAI 4755*, pages 26–38, Sendai, 2007. Springer.
- [61] J. R. Searle. *Mind: A Brief Introduction*. Oxford University Press, USA, 2005.
- [62] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009.
- [63] S. Singh, M. Littman, N. Jong, D. Pardoe, and P. Stone. Learning predictive state representations. In *Proc. 20th International Conference on Machine Learning (ICML'03)*, pages 712–719, 2003.
- [64] B. F. Skinner. *About Behaviorism*. Random House, 1974.
- [65] R. J. Solomonoff. A formal theory of inductive inference: Parts 1 and 2. *Information and Control*, 7:1–22 and 224–254, 1964.
- [66] R. J. Solomonoff. Complexity-based induction systems: Comparisons and convergence theorems. *IEEE Transactions on Information Theory*, IT-24:422–432, 1978.
- [67] R. L. Solso, O. H. MacLin, and M. K. MacLin. *Cognitive Psychology*. Allyn & Bacon, 8th edition, 2007.
- [68] A. L. Strehl, C. Diuk, and M. L. Littman. Efficient structure learning in factored-state MDPs. In *Proc. 27th AAAI Conference on Artificial Intelligence*, pages 645–650, Vancouver, BC, 2007. AAAI Press.
- [69] N. Suematsu and A. Hayashi. A reinforcement learning algorithm in partially observable environments using short-term memory. In *Advances in Neural Information Processing Systems 12 (NIPS'09)*, pages 1059–1065, 1999.
- [70] N. Suematsu, A. Hayashi, and S. Li. A Bayesian approach to model learning in non-Markovian environments. In *Proc. 14th Intl. Conf. on Machine Learning (ICML'97)*, pages 349–357, 1997.
- [71] P. Sunehag and M. Hutter. Consistency of feature Markov processes. In *Proc. 21st International Conf. on Algorithmic Learning Theory (ALT'10)*, volume 6331 of *LNAI*, pages 360–374, Canberra, 2010. Springer, Berlin.
- [72] P. Sunehag and M. Hutter. Axioms for rational reinforcement learning. In *Proc. 22nd International Conf. on Algorithmic Learning Theory (ALT'11)*, volume 6925 of *LNAI*, pages 338–352, Espoo, Finland, 2011. Springer, Berlin.
- [73] P. Sunehag and M. Hutter. Principles of Solomonoff induction and AIXI. In *Proc. Solomonoff 85th Memorial Conference, LNAI*, Melbourne, Australia, 2011. Springer.
- [74] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.
- [75] A. Tettamanzi, M. Tomassini, and J. Janßen. *Soft Computing: Integrating Evolutionary, Neural, and Fuzzy Systems*. Springer, 2001.
- [76] A. M. Turing. Computing machinery and intelligence. *Mind*, 1950.
- [77] R. Turner. *Logics for Artificial Intelligence*. Ellis Horwood Series in Artificial Intelligence, 1984.

- [78] J. Veness, K. S. Ng, M. Hutter, and D. Silver. Reinforcement learning via AIXI approximation. In *Proc. 24th AAAI Conference on Artificial Intelligence (AAAI'10)*, pages 605–611, Atlanta, 2010. AAAI Press.
- [79] J. Veness, K. S. Ng, M. Hutter, W. Uther, and D. Silver. A Monte Carlo AIXI approximation. *Journal of Artificial Intelligence Research*, 40:95–142, 2011.
- [80] C. S. Wallace. *Statistical and Inductive Inference by Minimum Message Length*. Springer, Berlin, 2005.
- [81] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens. The context tree weighting method: Basic properties. *IEEE Transactions on Information Theory*, 41:653–664, 1995.
- [82] I. Wood, P. Sunehag, and M. Hutter. (Non-)equivalence of universal priors. In *Proc. Solomonoff 85th Memorial Conference, LNAI*, Melbourne, Australia, 2011. Springer.

Chapter 6

Deep Reinforcement Learning as Foundation for Artificial General Intelligence

Itamar Arel

*Machine Intelligence Lab, Department of Electrical Engineering and Computer Science,
University of Tennessee*

E-mail: itamar@ieee.org

Deep machine learning and reinforcement learning are two complementing fields within the study of intelligent systems. When combined, it is argued that they offer a promising path for achieving artificial general intelligence (AGI). This chapter outlines the concepts facilitating such merger of technologies and motivates a framework for building scalable intelligent machines. The prospect of utilizing custom neuromorphic devices to realize large-scale deep learning architectures is discussed, paving the way for achieving human-level AGI.

6.1 Introduction: Decomposing the AGI Problem

A fundamental distinction between Artificial General Intelligence (AGI) and “conventional” Artificial Intelligence (AI) is that AGI focuses on the study of systems that can perform tasks successfully across different problem domains, while AI typically pertains to domain-specific expert systems. General problem-solving ability is one that humans naturally exhibit. A related capability is generalization, which allows mammals to effectively associate causes perceived in their environment with regularities observed in the past. Another critical human skill involves decision making under uncertainty, tightly coupled with generalization since the latter facilitates broad situation inference.

Following this line of thought, it can be argued that at a coarse level, intelligence involves two complementing sub-systems: *perception* and *actuation*. Perception can be interpreted as mapping sequences of observations, possibly received from multiple modalities, to an inferred state of the world with which the intelligence agent interacts. Actuation is

often framed as a control problem, centering on the goal of selecting actions to be taken at any given time so as to maximize some utility function. In other words, actuation is a direct byproduct of a decision making process, whereby inferred states are mapped to selected actions, thereby impacting the environment in some desirable way. This high-level view is depicted in Figure 6.1.

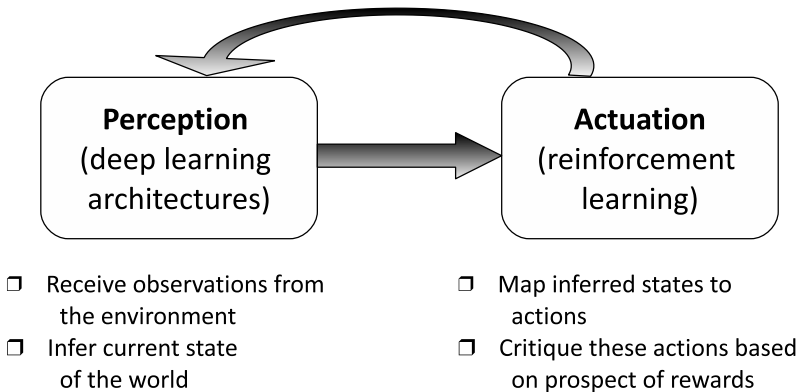


Fig. 6.1 Bipartite AGI architecture comprising of a perception and control/actuation subsystem. The role of the perception subsystem is viewed as state inference while the control subsystem maps inferred states to desired actions. Actuation impacts subsequent perceptive stimuli, as denoted by the feedback signal.

Two relatively new thrusts within machine learning contribute, respectively, to the core AGI components mentioned above. Deep machine learning (DML) is a niche that focuses on scalable information representation architectures that loosely mimic the manner by which the mammalian cortex interprets and represents observations. As a direct consequence, deep learning architectures [5] can serve as scalable state inference engines, driving perception subsystems.

Complementing DML is Reinforcement learning (RL) [11] – a fairly mature field of study, concerning algorithms that attempt to approximately solve an optimal control problem, whereby action selection is guided by the desire to maximize an agent’s expected reward prospect. RL is inspired by many studies of recent years, supporting the notion that much of the learning that goes on in mammalian brain is driven by rewards and their expectations, both positive and negative.

This chapter hypothesizes that the merger of these two technologies, in the context of the bipartite system architecture outlined above, may pave the way for a breakthrough in

our ability to build systems that will eventually exhibit human-level intelligence. Such merger is coined deep reinforcement learning (DRL). Moreover, recent advances in VLSI technology, particularly neuromorphic circuitry, suggest that the means to fabricate large-scale DRL systems are within our reach.

The rest of the chapter is structured as follows. In Section 6.2 we review deep learning architectures and motivate their role in designing perception engines. Section 6.3 reviews reinforcement learning and outlines how it can be merged with deep learning architectures. Section 6.4 discusses the scalability implications of designing AGI systems using emerging neuromorphic technology, while in Section 6.5 conclusions are drawn and future outlook is discussed.

6.2 Deep Learning Architectures

6.2.1 *Overcoming the Curse of Dimensionality*

Mimicking the efficiency and robustness with which the human brain represents information has been a core challenge in artificial intelligence research for decades. Humans are exposed to myriad of sensory data received every second of the day and are somehow able to capture critical aspects of this data in a way that allows for its future recollection. Over 50 years ago, Richard Bellman, who introduced dynamic programming theory and pioneered the field of optimal control, asserted that high dimensionality of data is a fundamental hurdle in many science and engineering applications. The main difficulty that arises, particularly in the context of real-world observations such as large visual fields, is that the learning complexity grows exponentially with linear increase in the dimensionality of the data. He coined this phenomenon the *curse of dimensionality* [1].

The mainstream approach of overcoming the Curse of Dimensionality has been to pre-process the data in a manner that would reduce its dimensionality to that which can be effectively processed, for example by a classification engine. This dimensionality reduction scheme is often referred to as feature extraction. As a result, it can be argued that the intelligence behind many pattern recognition systems has shifted to the human-engineered feature extraction process, which at times can be challenging and highly application-dependent [2]. Moreover, if incomplete or erroneous features are extracted, the classification process is inherently limited in performance.

Recent neuroscience findings have provided insight into the principles governing information representation in the mammal brain, leading to new ideas for designing systems that

represent information. Some researchers claim that the neocortex, which is associated with many cognitive abilities, does not explicitly pre-process sensory signals, but rather allows them to propagate through a complex hierarchy [3] of modules that, over time, learn to represent observations based on the regularities they exhibit [4]. This discovery motivated the emergence of the subfield of deep machine learning [5, 6], which focuses on computational models for information representation that exhibit similar characteristics to that of the neocortex.

In addition to the spatial dimensionality of real-life data, the temporal component also plays a key role. A sequence of patterns that we observe often conveys a meaning to us, whereby independent fragments of this sequence would be hard to decipher in isolation. We often infer meaning from events or observations that are received close in time [7, 8]. To that end, modeling the temporal component of the observations plays a critical role in effective information representation. Capturing spatiotemporal dependencies, based on regularities in the observations, is therefore viewed as a fundamental goal for deep learning systems.

Recent literature treats pure multi-layer perceptron (MLP) neural networks with more than two hidden layers as deep learning architectures. Although one can argue that technically that is a correct assertion, the mere fact that a learning system hosts multiple layers is insufficient to be considered as a deep learning architecture. The latter should also encompass the idea of a hierarchy of abstraction, whereby as one ascends the hierarchy more abstract notions are formed. This is not directly attainable in a simple MLP consisting of a large number of layers.

6.2.2 Spatiotemporal State Inference

A particular family of DML systems is *compositional* deep learning architectures. The latter are characterized by hosting multiple instantiations of a basic cortical circuit (or *node*) which populate all layers of the architecture. Each node is tasked with learning to represent the sequences of patterns that are presented to it by nodes in the layer that precede it. At the very lowest layer of the hierarchy nodes receive as input raw data (e.g. pixels of the image) and continuously construct a *belief state* that attempts to compactly characterize the sequences of patterns observed. The second layer, and all those above it, receive as input the belief states of nodes at their corresponding lower layers, and attempt to construct their own belief states that capture regularities in their inputs. Figure 6.2 illustrates a compositional deep learning architecture.

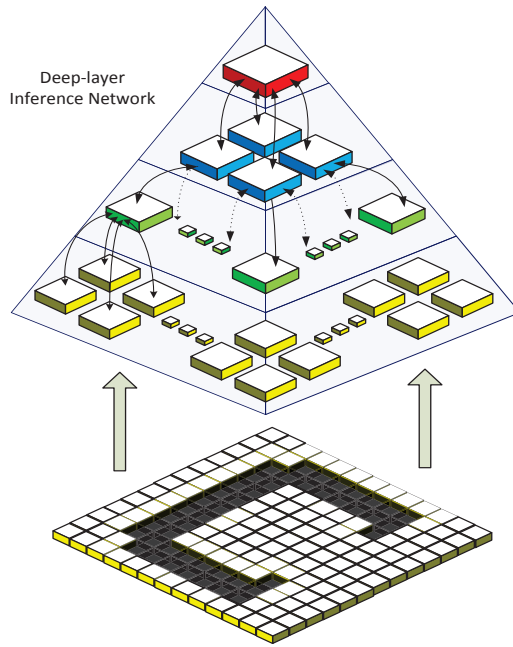


Fig. 6.2 Compositional deep machine learning architecture, comprising of multiple instantiations of a common cortical circuit, illustrated in the context of visual information processing.

Information flows both bottom up and top down. Bottom up processing essentially constitutes a feature extraction process, in which each layer aggregates data from the layer below it. Top down signaling helps lower layer nodes improve their representation accuracy by assisting in correctly disambiguating distorted observations.

An AGI system should be able to adequately cope in a world where partial observability is assumed. Partial observability means that any given observation (regardless of the modalities from which it originates) does not provide full information needed to accurately infer the true state of the world. As such, an AGI system should map sequences of observations to an internal state construct that is consistent for regular causes. This implies that a dynamic (i.e. memory-based) learning process should be exercised by each cortical circuit.

For example, if a person looks at a car in a parking lot he/she would recognize it as such since there is consistent signaling being invoked in their brain whenever car patterns are observed. In fact, it is sufficient to hear a car (without viewing it) to invoke similar signaling in the brain. While every person may have different signaling for common causes in the world, such signaling remains consistent for each person. This consistency property allows

a complementing control subsystem to map the (inferred) states to actions that impact the environment in some desirable way.

If a deep learning architecture is to form an accurate state representation, it should include both spatial and temporal information. As a result, each belief state should capture spatiotemporal regularities in the observations, rather than just spatial saliences.

The learning process at each node is unsupervised, guided by exposure to a large set of observations and allowing the salient attributes of these observations to be captured across the layers. In the context of an AGI system, signals originating from upper-layer nodes can be extracted to serve as inferred state representations. This extracted information should exhibit invariance to common distortions and variations in the observations, leading to representational robustness. In the context of visual data, robustness refers to the ability to exhibit invariance to a diverse range of transformations, including mild rotation, scale, different lighting conditions and noise.

It should be noted that although deep architectures may appear to completely solve or overcome the curse of dimensionality, in reality they do so by hiding the key assumption of locality. The latter means that the dependencies that may exist between two signals (e.g. pixels) that are spatially close are captured with relative detail, where as relationships between signals that are distant (e.g. pixels on opposite sides of a visual field) are represented with very little detail. This is a direct result of the nature of the architecture depicted in Figure 6.2, in which fusion of information from inputs that are distant to the hierarchy occurs at the higher layers.

It is also important to emphasize that deep learning architectures are not limited by any means to visual data. In fact, these architectures are modality agnostic, and attempt to discover underlying structure in data of any form. Moreover, fusion of information originating from different modalities is natural in deep learning and a pivotal requirement of AGI. If one imagines the architecture shown in Figure 6.1 to receive input at its lowest layer from multiple modalities, as one ascends the hierarchy, fusion of such information takes place by capturing regularities across the modalities.

6.3 Scaling Decision Making under Uncertainty

6.3.1 *Deep Reinforcement Learning*

While the role of the perception subsystem may be viewed as that of complex state inference, an AGI system must be able to take actions that impact its environment. In

other words, an AGI system must involve a controller that attempts to optimize some cost function. This controller is charged with mapping the inferred states to an action. In real-world scenarios, there is always some uncertainty. However, state signaling should exhibit the Markov property in the sense that it compactly represents the history that has led to the current state-of-affairs. This is a colossal assumption, and one that is unlikely to accurately hold. However, it is argued that while the Markov property does practically not hold, assuming that it does paves the way for obtaining “good enough”, albeit not optimal, AGI systems [21].

Reinforcement learning (RL) corresponds to a broad class of machine learning techniques that allow a system to learn how to behave in an environment that provides reward signals. A key related concept is that the agent learns by itself, based on acquired experience, rather than by being externally instructed or supervised. RL inherently facilitates autonomous learning as well as addresses many of the essential goals of AGI: it emphasizes the close interaction of an agent with the environment, it focuses on perception-to-action cycles and complete behaviors rather than separate functions and function modules, it relies on bottom-up intelligent learning paradigms, and it is not based on symbolic representations.

It should be emphasized that it is inferred states, rather than pure perceptive stimuli, that is mapped to actions by the system. On that note, although it is argued that RL-based AGI is not achieved via explicit symbolic representations, the latter do form implicitly in the architecture, primarily in the form of attractors in both the perception and control subsystems. As an example, regularities in the observations that offer semantic value to the agent, as reflected by sequences of rewards, will receive an internal representation, relative to other notions acquired, thereby resembling classical symbolic knowledge maps.

The ability to generalize is acknowledged as an inherent attribute of intelligent systems. Consequently, it may be claimed that no system can learn without employing some degree of approximation. The latter is particularly true when we consider large-scale, complex real-world scenarios, such as those implied by true AGI. Deep learning architectures can serve this exact purpose: they can provide a scalable state inference engine that a reinforcement learning based controller can map to actions.

A recent and very influential development in RL is the actor-critic approach to model-free learning, which is based on the notion that two distinct core functions accomplish learning: the first (the “actor”) produces actions derived from an internal model and the second (the “critic”) refines the action selection policy based on prediction of long-term

reward signals. As the actor gains proficiency, it is required to learn an effective mapping from inferred states of the environment to actions. In parallel to the growing support of RL theories in modern cognitive science, recent work in neurophysiology provides some evidence arguing that the actor-critic RL theme is widely exploited in the human brain.

The dominant mathematical methods supporting learning approximately solve the Hamilton-Jacobi-Bellman (HJB) equation of dynamic programming (DP) to iteratively adjust the parameters and structure of an agent as a means of encouraging desired behaviors [5]. A discounted future reward is typically used; however, researchers are aware of the importance of multiple time scales and the likelihood that training efficiency will depend upon explicit consideration of multiple time horizons. Consequently, the trade-offs between short and long term memory should be considered. Cognitive science research supports these observations, finding similar structures and mechanisms in mammalian brains [9].

The HJB equation of DP requires estimation of expected future rewards, and a suitable dynamic model of the environment that maps the current observations and actions (along with inferred state information) to future observations. Such model-free reinforcement learning assumes no initial knowledge of the environment, and instead postulates a generic structure, such as a deep learning architecture, that can be trained to model environmental responses to actions and exogenous sensory inputs.

Contrary to existing function approximation technologies, such as standard multi-layer perceptron networks, current neurophysiology research reveals that the structure of the human brain is dynamic, with explosive growth of neurons and neural connections during fetal development followed by pruning. Spatial placement also plays critical roles, and it is probable that the spatial distribution of chemical reward signals selectively influences neural adaptation to enhance learning [10]. It is suspected that the combination of multi-time horizon learning and memory processes with the dynamic topology of a spatially embedded deep architecture will dramatically enhance adaptability and effectiveness of artificial cognitive agents. This is expected to yield novel AGI frameworks that can overcome the limitations of existing AI systems.

RL can thus be viewed as a biologically-inspired decision making under uncertainty framework that is centered on the notion of learning from experience, through interaction with an environment, rather than by being explicitly guided by a teacher. What sets RL apart from other machine learning methods is that it aims to solve the *credit assignment problem*, in which an agent is charged with evaluating the long-term impact of actions it

takes. In doing so, the agent attempts to choose actions that maximize its estimated *value function*, based only on state information and nonspecific reward signals.

In a real-world setting, the agent constructs an estimated value function that expresses the prospect of rewards the agent expects to experience by taking a specific action at a given state. Temporal difference (TD) learning [11] is a central idea in reinforcement learning, and is primarily applied to model-free learning problems. The TD paradigm draws from both dynamic programming [1] and Monte Carlo methods [11]. Similar to dynamic programming, TD learning bootstraps in that it updates value estimates based on other value estimates, as such not having to complete an episode before updating its value function representation. Like Monte Carlo methods, TD is heuristic in that it uses experience, obtained by following a given policy (i.e. mapping of states to actions), to predict subsequent value estimates. TD updates are performed as a single step look-ahead that typically takes the form of

$$V(t+1) = V(t) + \alpha * (target - V(t)). \quad (6.1)$$

where *target* is derived from the Bellman equation [11] and depends on how rewards are evaluated over time, V_t denotes the value estimate of a given state at time t , and α is a small positive constant.

In real-world settings, particularly those relevant to AGI, only partial information regarding the true state of the world is available to the agent. The agent is thus required to form a belief state from observations it receives of the environment. Assuming the Markov property holds, but state information is inaccurate or incomplete, we say that the problem is partially observable. Deep learning architectures help overcome partial observability by utilizing their internal state constructs (across the different layers) to capture temporal dependencies. The latter disambiguate observations that are partial, noisy or otherwise insufficient to infer the state of the environment.

6.3.2 Actor-Critic Reinforcement Learning Themes in Cognitive Science

The fundamental notion of learning on the basis of rewards is shared among several influential branches of psychology, including behaviorism and cognitive psychology. The actor-critic architecture reflects recent trends in cognitive neuroscience and cognitive psychology that highlight task decomposition and modular organization. For example, visual information-processing is served by two parallel pathways, one specialized to object location in space and the other to object identification or recognition over space and time [12, 13]. This approach exploits a divide-and-conquer processing strategy in which particular

components of a complex task are computed in different cortical regions, and typically integrated, combined, or supervised by the prefrontal cortex.

Computational models of this dual-route architecture suggest that it has numerous benefits over conventional homogenous networks, including both learning speed and accuracy. More generally, the prefrontal cortex is implicated in a wide range of cognitive functions, including maintaining information in short-term or working memory, action planning or sequencing, behavioral inhibition, and anticipation of future states. These functions highlight the role of the prefrontal cortex as a key location that monitors information from various sources and provides top-down feedback and control to relevant motor areas (e.g., premotor cortex, frontal eye fields, etc.). In addition to recent work in cognitive neuroscience, theoretical models of working memory in cognitive psychology also focus on the role of a central executive that actively stores and manipulates information that is relevant for solving ongoing tasks.

A unique feature of the proposed AGI approach is a general-purpose cognitive structure for investigating both external and internal reward systems. Cognitive psychologists conceptualize these two forms of reward as extrinsic and intrinsic motivation [14]. Extrinsic motivation corresponds to changes in behavior as a function of external contingencies (e.g., rewards and punishments), and is a central element of Skinner's theory of learning. Meanwhile, intrinsic motivation corresponds to changes in behavior that are mediated by internal states, drives, and experiences, and is manifested in a variety of forms including curiosity, surprise, and novelty. The concept of intrinsic motivation is ubiquitous in theories of learning and development, including the notions of (1) mastery motivation (i.e., a drive for proficiency [15], (2) functional assimilation (i.e., the tendency to practice a new skill, e.g., Piaget, 1952), and (3) violation-of-expectation (i.e., the tendency to increase attention to unexpected or surprising events, e.g., [16]).

It is interesting to note that while external rewards play a central role in RL, the use of intrinsic motivation has only recently begun to receive attention from the machine-learning community. This is an important trend, for a number of reasons. First, intrinsic motivation changes dynamically in humans, not only as a function of task context but also general experience. Implementing a similar approach in autonomous-agent design will enable the agent to flexibly adapt or modify its objectives over time, deploying attention and computational resources to relevant goals and sub-goals as knowledge, skill, and task demands change. Second, the integration of a dual-reward system that includes both external and

intrinsic motivation is not only biologically plausible, but also more accurately reflects the continuum of influences in both human and non-human learning systems.

In parallel to the growing support of model-free Actor-Critic models in modern psychology, recent work in neurophysiology provides evidence suggesting that the Actor-Critic paradigm is widely exploited in the brain. In particular, it has been recently shown that the basal ganglia [17] can be coarsely modeled by an Actor-Critic version of temporal difference (TD) learning. The frontal dopaminergic input arises in a part of the basal ganglia called ventral tegmental area (VTA) and the substantia nigra (SN). The signal generated by dopaminergic (DA) neurons resembles the effective reinforcement signal of temporal difference (TD) learning algorithms.

Another important part of the basal ganglia is the striatum. This structure is comprised of two parts, the matriosome and the striosome. Both receive input from the cortex (mostly frontal) and from the DA neurons, but the striosome projects principally to DA neurons in VTA and SN. The striosome is hypothesized to act as a reward predictor, allowing the DA signal to compute the difference between the expected and received reward. The matriosome projects back to the frontal lobe (for example, to the motor cortex). Its hypothesized role is therefore in action selection.

6.4 Neuromorphic Devices Scaling AGI

The computational complexity and storage requirements from deep reinforcement learning systems limit the scale at which they may be implemented using standard digital computers. An alternative would be to consider custom analog circuitry as means of overcoming the limitations of digital VLSI technology. In order to achieve the largest possible learning system within any given constraints of cost or physical size, it is critical that the basic building blocks of the learning system be as dense as possible. Many operations can be realized in analog circuitry with a space saving of one to two orders of magnitude compared to a digital realization. Analog computation also frequently comes with a significant reduction in power consumption, which will become critical as powerful learning systems are migrated to battery-operated platforms.

This massive improvement in density is achieved by utilizing the natural physics of device operation to carry out computation. The benefits in density and power come with certain disadvantages, such as offsets and inferior linearity compared to digital implementations. However, the weaknesses of analog circuits are not major limitations since the

feedback inherent in the learning algorithms naturally compensates for errors/inaccuracies introduced by the analog circuits. The argument made here is that the brain is far from being 64-bit accurate, so relaxing accuracy requirements of computational elements, for the purpose of aggressively optimized for area, is a valid tradeoff.

The basic requirements of almost any machine learning algorithm include multiplication, addition, squashing functions (e.g. sigmoid), and distance/similarity calculation, all of which can be realized in a compact and power-efficient manner using analog circuitry. Summation is trivial in the current domain as it is accomplished by joining the wires with the currents to be summed. In the voltage domain, a feedback amplifier with $N + 1$ resistors in the feedback path can compute the sum of N inputs. A Gilbert cell [18] provides four-quadrant multiplication while using only seven transistors.

Table 6.1 contrasts the component count of digital and analog computational blocks. As discussed above, the learning algorithms will be designed to be robust to analog circuit imperfections, allowing the use of very small transistors. Digital designs vary widely in transistor count, as area can frequently be traded for speed. For the comparison, we used designs that are optimized for area and appropriate for the application. For example, an N -bit “shift-and-add” multiplier uses a single adder by performing the multiplication over N clock cycles. A fast multiplier might require as many as N times more adders. Digital registers were chosen over SRAMs, despite their larger size because SRAMs require significant peripheral circuitry (e.g. address decoders, sense amplifiers) making them poorly suited to a system requiring many small memories. For the analog elements, we also counted any necessary resistors or capacitors.

Table 6.1 Transistor Count Comparison of Analog and Digital Elements.

Operation	Analog	Digital (N bits)	Notes
Summation	12	$24N$	Feedback voltage adder; Ripple-carry adder
Multiplication	7	$48N$	Shift & add multiplier
Storage	15	$12N$	Feedback floating-gate cell; Digital register

A particularly suitable analog circuit, which is often used for computational purposes, is the floating gate transistor [19] (shown in Figure 6.3). Floating gates have proved themselves to be useful in many different applications; they make good programmable switches, allow for threshold matching in transistor circuits, and have been successfully used in the design of various adaptive systems.

Floating gates lack a DC path to ground, so any charge stored on the gate will stay there. Through the use of Fowler-Nordheim tunneling and hot electron injection, this trapped charge can be modified. Floating-gate memories can provide a finely tuned voltage to match thresholds between multiple transistors, for example, to yield a circuit which has nearly perfect matching, improving accuracy relative to conventional techniques. In learning systems, a floating gate can be used to store a weight or learned parameter.

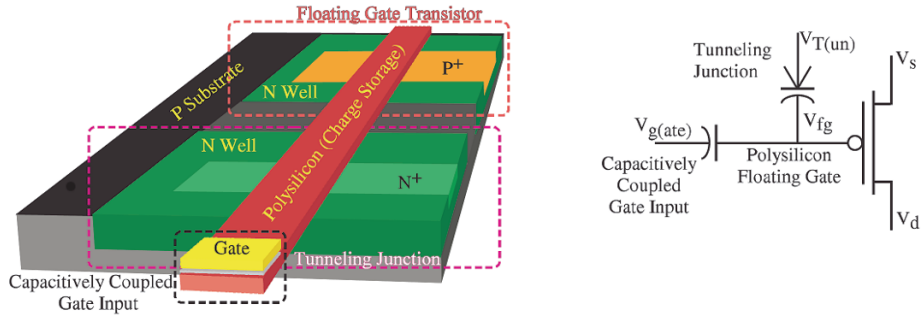


Fig. 6.3 (A) Cutout showing parts of a typical floating gate transistor. (B) A floating gate in schematic.

6.5 Conclusions and Outlook

This chapter has outlined a deep reinforcement learning based approach for achieving AGI. The merger between deep architectures as scalable state inference engines and reinforcement learning as a powerful control framework offers the potential for an AGI breakthrough. It was further argued that large-scale intelligence systems can be built using existing neuromorphic technology.

Many issues remain in enhancing the introduced approach to address the various critical attributes of true AGI systems. The capacity and role of intrinsic rewards, generated as a product on internal cognitive processes, needs to be better understood. The manner by which virtual goals are created in the brain merits further studying and modeling. The impact of traumatic experiences, for example, plays a key role in the human psyche, serving a critical purpose of imprinting vital information and/or experiences for long haul recollection and inference. While many such cognitive phenomena are poorly understood and are likely to pose modeling challenges, the paradigm proposed in this chapter is inherently generic and serves as solid basis for AGI research in the years to come. Pragmatically

speaking, the technology needed to experiment with deep reinforcement learning based AGI exists today. Moreover, such technology is within reach for many research institutions, rendering AGI breakthrough a possible reality in the near future.

Bibliography

- [1] R. Bellman, *Dynamic Programming*, Princeton University Press, (1957).
- [2] R. Duda, P. Hart, and D. Stork, *Pattern Recognition*, Wiley-Interscience, 2nd Edition, (2000).
- [3] T. Lee and D. Mumford, "Hierarchical Bayesian Inference in the Visual Cortex", *Journal of the Optical Society of America*, vol. 20, part 7, pp. 1434–1448, (2003).
- [4] T. Lee, D. Mumford, R. Romero, and V. Lamme, "The role of the primary visual cortex in higher level vision," *Vision Research*, vol. 38, pp. 2429–2454, (1998).
- [5] I. Arel, D. Rose, and T. Karnowski, "Deep Machine Learning - A New Frontier in Artificial Intelligence Research," *IEEE Computational Intelligence Magazine*, Vol. 14, pp. 12–18, Nov., (2010).
- [6] Y. Bengio, "Learning Deep Architectures for AI", *Foundations and Trends in Machine Learning*, (2009).
- [7] G. Wallis and H. Bülthoff, "Learning to recognize objects", *Trends in Cognitive Sciences*, vol. 3, issue 1, pp. 23–31, (1999).
- [8] G. Wallis and E. Rolls, "Invariant Face and Object Recognition in the Visual System," *Progress in Neurobiology*, vol. 51, pp. 167–194, (1997).
- [9] R.E. Suri and W. Schultz, "A neural network model with dopamine-like reinforcement signal that learns a spatial delayed response task," *Neuroscience*, vol. 91, no. 3, pp. 871–890, 1999.
- [10] W. Schultz, "Predictive reward signal of dopamine neurons," *The Journal of Neurophysiology*, vol. 80, no. 1, pp. 1–27, 1998.
- [11] R.S. Sutton and A.G. Barto, *Reinforcement Learning: An Introduction*, Cambridge MA, MIT Press, 1998.
- [12] A.D. Millner and M.A. Goodale, *The Visual Brain in Action*, Oxford University Press, 1996.
- [13] M. Mishkin, L. G. Ungerkeuder, and K. A. Macko, "Object vision and spatial vision: two cortical pathways," *Trends in Neuroscience*, vol. 6, pp. 414–417, 1983.
- [14] G.S. Ginsburg and P. Bronstein, "Family factors related to children's intrinsic/extrinsic motivational orientation and academic performance," *Child Development*, vol. 64, pp. 1461–1474, 1993.
- [15] S.A. Kelley, C.A. Brownell, and S.B. Campbell, "Mastery motivation and self-evaluative affect in toddlers: longitudinal relations with maternal behavior," *Child Development*, vol. 71, pp. 1061–71, 2000.
- [16] R. Baillargeon, "Physical reasoning in young infants: Seeking explanations for impossible events," *British Journal of Developmental Psychology*, vol. 12, pp. 9–33, 1994.
- [17] D. Joel, Y. Niv, and E. Ruppín, "Actor-critic models of the basal ganglia: New anatomical and computational perspectives," *Neural Networks*, vol. 15, pp. 535–547, 2002.
- [18] P. Gray, P. Hurst, S. Lewis, and R. Meyer, *Analysis and design of analog integrated circuits*, John Wiley and Sons, 2001.
- [19] P. Hasler and J. Dugger, "An analog floating-gate node for supervised learning," *IEEE Transactions on Circuits and Systems I*, vol. 52, no. 5, pp. 834–845, May 2005.
- [20] L.P. Kaelbling, M.L. Littman, and A.R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence Journal*, 101: 99–134, 1998.

Chapter 7

The LIDA Model as a Foundational Architecture for AGI

Usef Faghghi and Stan Franklin

*Computer Science Department & Institute for Intelligent Systems
The University of Memphis, Memphis, TN, 38152 USA*

E-mail: {ufaghghi, franklin}@memphis.edu

Artificial intelligence (AI) initially aimed at creating “thinking machines,” that is, computer systems having human level general intelligence. However, AI research has until recently focused on creating intelligent, but highly domain-specific, systems. Currently, researchers are again undertaking the original challenge of creating AI systems (agents) capable of human-level intelligence, or “artificial general intelligence” (AGI). In this chapter, we will argue that Learning Intelligent Distribution Agent (LIDA), which implements Baars’ Global Workspace Theory (GWT), may be suitable as an underlying cognitive architecture on which others might build an AGI. Our arguments rely mostly on an analysis of how LIDA satisfies Sun’s “desiderata for cognitive architectures” as well as Newell’s “test for a theory of cognition.” Finally, we measure LIDA against the architectural features listed in the BICA Table of Implemented Cognitive Architectures, as well as to the anticipated needs of AGI developers.

7.1 Introduction

The field of artificial intelligence (AI) initially aimed at creating “thinking machines,” that is, creating computer systems having human level general intelligence. However, AI research has until recently mostly focused on creating intelligent, but highly domain-specific, computer systems. At present however, researchers are again undertaking the original challenge of creating AI systems (agents) capable of human-level intelligence, or “artificial general intelligence” (AGI).

To do so it may well help to be guided by following question, *how do minds work?* Among different theories of cognition, we choose to work from the Global Workspace

Theory (GWT) of Baars [1,2] the most widely accepted psychological and neurobiological theory of the role of consciousness in cognition [3–6].

GWT is a neuropsychological theory of the role of consciousness in cognition. It views the nervous system as a distributed parallel system incorporating many different specialized processes. Various coalitions of these specialized processes facilitate making sense of the sensory data currently coming in from the environment. Other coalitions sort through the results of this initial processing and pick out items requiring further attention. In the competition for attention a winner emerges, and occupies what Baars calls the global workspace, the winning contents of which are presumed to be at least functionally conscious [7]. The presence of a predator, enemy, or imminent danger should be expected, for example, to win the competition for attention. However, an unexpected loud noise might well usurp consciousness momentarily even in one of these situations. The contents of the global workspace are broadcast to processes throughout the nervous system in order to recruit an action or response to this salient aspect of the current situation. The contents of this global broadcast enable each of several modes of learning. We will argue that Learning Intelligent Distribution Agent (LIDA) [8], which implements Baars' GWT, may be suitable as an underlying cognitive architecture on which to build an AGI.

The LIDA architecture, a work in progress, is based on the earlier IDA, an intelligent, autonomous, “conscious” software agent that does personnel work for the US Navy [9]. IDA uses locally developed artificial intelligence technology designed to model human cognition. IDA's task is to find jobs for sailors whose current assignments are about to end. She selects jobs to offer a sailor, taking into account the Navy's policies, the job's needs, the sailor's preferences, and her own deliberation about feasible dates. Then she negotiates with the sailor, in English via a succession of emails, about job selection. We use the word “conscious” in the functional consciousness sense of Baars' Global Workspace Theory [1, 2], upon which our architecture is based (see also [7]) .

7.2 Why the LIDA model may be suitable for AGI

The LIDA model of cognition is a fully integrated artificial cognitive system capable of reaching across a broad spectrum of cognition, from low-level perception/action to high-level reasoning. The LIDA model has two faces, its science side and its engineering side.

LIDA's science side fleshes out a number of psychological and neuropsychological theories of human cognition including GWT [2], situated cognition [10], perceptual sym-

bol systems [11], working memory [12], memory by affordances [13], long-term working memory [14], and the H-CogAff architecture [15].

The LIDA architecture engineering side explores architectural designs for software agents that promise more flexible, more human-like intelligence within their domains. It employs several modules that are designed using computational mechanisms drawn from the “new AI.” These include variants of the Copycat Architecture [16, 17], Sparse Distributed Memory [18, 19], the Schema Mechanism [20, 21], the Behavior Net [22], and the Subsumption Architecture [23]. However, an AGI developer using LIDA need make no commitment to any of these mechanisms. The computational framework of the LIDA architecture [24] allows free substitution of such mechanisms (see below). The required commitment is relatively modest, consisting primarily of a weak adherence to the LIDA cognitive cycle (see below). In addition, the LIDA architecture can accommodate the myriad features¹ that will undoubtedly be required of an AGI (see below). Thus the LIDA architecture, empirically based on psychological and biological principles, offers the flexibility to relatively easily experiment with different paths to an AGI. This makes us think of LIDA as eminently suitable for an underlying foundational architecture for AGI.

7.3 LIDA architecture

Any AGI will have to deal with tremendous amounts of sensory inputs. It will therefore need attention to filter the incoming sensory data to recruit resources in order to respond, and to learn. Note that this greatly resembles the Global Workspace Theory broadcast. By definition, every AGI must be able to operate in a wide variety of domains. It must therefore be capable of very flexible decision making. Flexible motivation, resulting from and modulated by feelings and emotions are in turn crucial to this end. The LIDA framework is setup accordingly. LIDA can be thought of as a proof of concept model for GWT. Many of the tasks in this model are accomplished by codelets [16] implementing the processors in GWT. Codelets are small pieces of code, each running independently. A class of codelets called attention codelets serves, with the global workspace, to implement attention. An attention codelet attempts to bring the contents of its coalition to the ‘consciousness’ spotlight. A broadcast then occurs, directed to all the processors in the system, to recruit resources with which to handle the current situation, and to learn.

¹Here we distinguish between features of an architecture such as one of its main components (e.g., Sensory Processing) and features of, say, an object such as its colors.

The LIDA architecture is partly symbolic and partly connectionist with all perceptual symbols being grounded in the physical world in the sense of Brooks [25]. Thus the LIDA architecture is embodied. (For further information on situated or embodied cognition, please see [26–29]. LIDA performs through its cognitive cycles (Figure 7.1), which occur five to ten times a second [30, 31], and depend upon saliency determination by the agent. A cognitive cycle starts with a sensation and usually ends with an action. The cognitive cycle is conceived of as an active process that allows interactions between the different components of the architecture. Thus, cognitive cycles are always on-going.

We now describe LIDA's primary mechanisms.

1) *Perceptual Associative Memory (PAM):* This corresponds neurally to the parts of different sensory cortices in humans (visual, auditory and somatosensory), plus parts of other areas (e.g. entorhinal cortex). PAM allows the agent to distinguish, classify and identify external and internal information. PAM is implemented in the LIDA architecture with a version of the slipnet [16]. There are connections between slipnet nodes. Segments of the slipnet are copied into the agent's Workspace as parts of the percept [32]. In LIDA, perceptual learning is learning to recognize new objects, new categorizations, and new relationships. With the conscious broadcast (Figure 7.1), new objects, categories, and the relationships among them and between them and other elements of the agent's ontology are learned by adding nodes (objects and categories) and links (relationships) to PAM. Existing nodes and links can have their base-level activations reinforced. The conscious broadcast begins and updates the process of learning to recognize and to categorize, both employing perceptual memory [8].

2) *Workspace:* This roughly corresponds to the human preconscious buffers of working memory [33]. This is the "place" that holds perceptual structures, which come from perception. It also includes previous percepts not yet decayed away, and local associations from episodic memories. These local associations are combined with the percepts to generate a Current Situational Model, the agent's understanding of what is going on right now. Information written in the workspace may reappear in different cognitive cycles until it decays away.

3) *Episodic memories:* These are memories for events (what, where and when). When the consciousness mechanism broadcasts information, it is saved into transient episodic memory (TEM) and is later consolidated into LIDA's declarative memory (DM) [34]. In LIDA, episodic learning refers to the memory of events – the what, the where and the

when [12,35]. In the LIDA model such learned events are stored in transient episodic memory [34,36] and in the longer-term declarative memory [34]. Both are implemented using sparse distributed memory [18], which is both associative and content addressable, and has other characteristics that correspond to psychological properties of memory. In particular it knows when it doesn't know, and exhibits the tip of the tongue phenomenon. Episodic learning in the LIDA model is also a matter of generate and test, with such learning occurring at the conscious broadcast of each cognitive cycle. Episodic learning is initially directed only to transient episodic memory. At a later time and offline, the undecayed contents of transient episodic memory are consolidated [37] into declarative memory, where they still may decay away or may last a lifetime.

4) Attentional Memory (ATM): ATM is implemented as a collection of a particular kind of codelet called an attention codelet. All attention codelets are tasked with finding their own specific content in the Current Situational Model (CSM) of the Workspace. For example, one codelet may look for a node representing fear. When an attention codelet finds its content it creates a coalition containing this content and related content. The coalition is added to the Global Workspace to compete for consciousness. Each attention codelet has the following attributes: 1) *concern*: that content, whose presence in the CSM, can trigger the codelet to act; 2) a base-level activation, a measure of the codelet's usefulness in bringing information to consciousness, as well as its general importance; and 3) a current activation which measures the degree of intersection between its concern and the content of the current situational model. The total activation measures the current saliency of its concern. We use a sigmoid function to both reinforce and decay the base-level and the current activations. The ATM includes several kinds of attention codelets. The *default* attention codelet reacts to whatever content it finds in the Current Situational Model in the Workspace, trying to bring its most energetic content to the Global Workspace. *Specific attention* codelets are codelets that may have been learned. They bring particular Workspace content to the Global Workspace. *Expectation codelets*, created during action selection, attempt to bring the result (or non-result) of a recently-executed action to consciousness. *Intention codelets* are attention codelets that bring any content that can help the agent reach a current goal to consciousness. That is, when the agent makes a volitional decision, an intention codelet is generated. There are attention codelets that react to the various dimensions of saliency, including novelty informativeness, importance, insistence, urgency and unexpectedness. Attentional learning is the learning of what to attend to [38,39]. In the LIDA model attentional learning involves attention codelets, small processes whose job

it is to focus the agent's attention on some particular portion of its internal model of the current situation. Again, learning occurs with the conscious broadcast with new attention codelets being created and existing attention codelets being reinforced.

5) Procedural Memory, Action Selection and Sensory-motor Memory: LIDA's procedural memory deals with deciding what to do next. It is similar to Drescher's schema mechanism but with fewer parameters [20,40]. The scheme net is a directed graph in which each of the nodes has a context, an action, and results. As a result of the conscious broadcast, schemes from Procedural Memory are instantiated and put into the Action Selection mechanism. The Action Selection mechanism then chooses an action and Sensory-Motor Memory executes the action (Figure 7.1). LIDA uses Maes' Behavior Network with some modifications [41] as its Action Selection mechanism [22]. Thus, in LIDA's architecture, while Procedural Memory and the Action Selection mechanism are responsible for deciding what will be done next, Sensory-Motor memory is responsible for deciding how tasks will be performed. Thus, each of these memory systems requires distinct mechanisms. In LIDA, procedural learning encodes procedures for possibly relevant behaviors into Procedural Memory (Figure 7.1). It is the learning of new actions and action sequences with which to accomplish new tasks. It is the learning of under what circumstances to perform new behaviors, or to improve knowledge of when to use existing behaviors. These procedural skills are shaped by reinforcement learning, operating by way of conscious processes over more than one cognitive cycle [8].

It must be noted that the LIDA model for the four aforementioned modes of learning, supports the instructional learning of new memory entities as well as the selectionist reinforcement of existing entities.

7.4 Cognitive architectures, features and the LIDA model

An AGI has to be built on some cognitive architecture, and by its nature, should share many features with other cognitive architectures. The most widely known cognitive architectures include Newell's Soar architecture [42–44], Anderson's ACT-R architecture [45–47], Sun's CLARION architecture [48], and Franklin's LIDA architecture [8]. The aforementioned cognitive architectures each have their strengths and weaknesses when it comes to defining a theory of mind [49, 50]. Some researchers have also tried to identify the most important features needed to construct biologically-inspired cognitive architectures (BICA). In particular, an AGI may well need the features listed by Sun (2004), and

by Newell [51], as well as features from the BICA table [52]. The BICA table is not shown in this paper because its size is beyond the size of this document (readers can refer to the online version for full details). The LIDA model seems to have room for all these features. In the next sections, we describe Sun’s desiderata and Newell’s functional criteria for cognitive architectures and, in italics, the LIDA model’s features that correspond to each of these cognitive architecture criteria. An assessment of LIDA against the features from the BICA table follows.

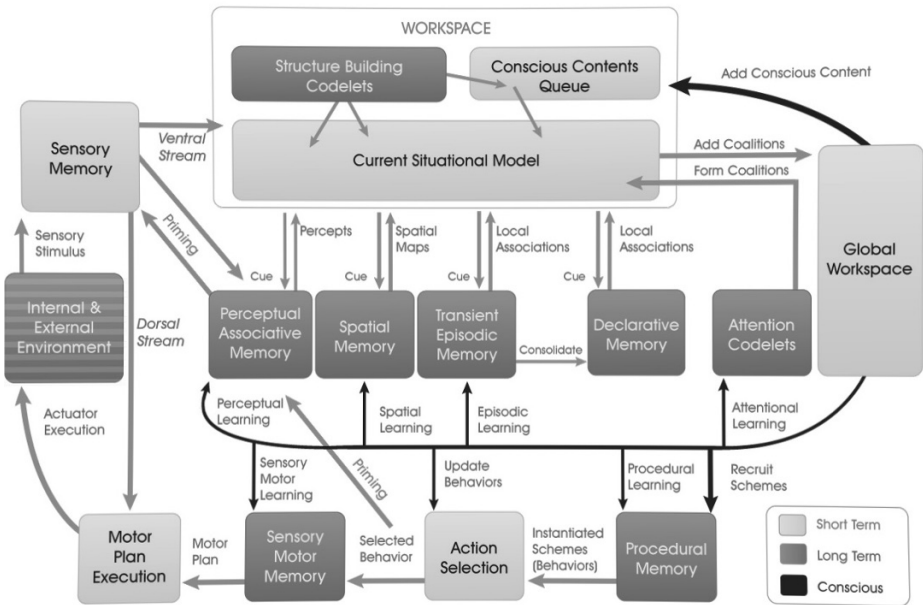


Fig. 7.1 LIDA’s Architecture

7.4.1 Ron Sun’s Desiderata [53]

In his article, Sun “proposes a set of essential desiderata for developing cognitive architectures,” and argues “for the importance of taking into full consideration these desiderata in developing future architectures that are more cognitively and ecologically realistic.” Though, AGI is not explicitly mentioned – the article predates the use of the term “AGI” – one could infer that Sun would consider them desiderata for an AGI as well. Here are some

of his desiderata interspersed with our assessment of how well the LIDA model does, or does not, achieve them.

Ecological realism: “Taking into account everyday activities of cognitive agents in their natural ecological environments.”

LIDA does allow ecological realism. It was designed for the everyday life of animals and/or artificial agents. Animals typically respond to incoming stimuli, often in search of food, mates, safety from predators, etc. Doing so requires frequent sampling of the environment, interpreting stimuli, attending to the most salient, and responding appropriately. The cascading LIDA cognitive cycles provides exactly this processing.

Bio-evolutionary realism supplements ecological realism. This feature refers to the idea that intelligences in species are on a continuum, and that cognitive models of human intelligence should not be limited to strictly human cognitive processes, but rather should be reducible to models of animal intelligence as well.

The LIDA model integrates diverse research on animal cognition. It has been used to provide both an ontology of concepts and their relations, and a working model of an animal’s cognitive processes [54, 55]. In addition to helping to account for a broad range of cognitive processes, the LIDA model can help to comparatively assess the cognitive capabilities of different animal species.

Cognitive realism. Cognition is highly variable within species. Cognitive realism refers to the idea that cognitive architectures should seek to replicate only essential characteristics of human cognition.

The reader is referred to the “Why the LIDA model may be suitable for AGI” section above to see that this condition is satisfied.

Eclecticism of methodologies and techniques. Adhering too closely to any methodology or paradigm will only prevent the creation of new or improved cognitive architectures. It is best to take a more broad-based approach.

Again the reader is referred to the “Why the LIDA model may be suitable for AGI” section.

Reactivity. Reactivity refers to fixed responses to given stimuli (as opposed to full-fledged analyses of stimuli to select a response) that characterize many human behaviors [15].

The LIDA model's reactive part is setup as a relatively direct connection between incoming sensory data and the outgoing actions of effectors. At the end of every cognitive cycle LIDA selects an appropriate behavior in response to the current situation. This consciously mediated, but unconscious, selection is always reactive.

Sequentiality. Sequentiality refers to the chronological nature of human everyday activities.

In LIDA, cognitive cycles allow the agent to perform its activities in a sequential manner. The cycles can cascade. But, these cascading cognitive cycles must preserve the sequentiality of the LIDA agent's stream of functional consciousness, as well as its selection of an action in each cycle [56].

Routineness. Routineness refers to the fact that humans' every day behaviors are made of routines, which are constantly and smoothly adapting to the changing environment.

The LIDA model is equipped with both reactive and deliberative high-level cognitive processes. Such processes become routine when they are incorporated (learned) into LIDA's procedural memory as schemes representing behavior streams.

Trial-and-error adaptation. Trial-and-error adaptation refers to the trial-and-error process through which humans learn and develop reactive routines.

LIDA learns from experience, which may yield several lessons over several cognitive cycles. Such lessons include newly perceived objects and their relationship to already known objects and categories, relationships among objects and between objects and actions, effects of actions on sensation, and improved perception of sensory data. All of LIDA's learning be it, perceptual, episodic, or procedural, is very much trial and error (generate and test as AI researchers would say). LIDA is profligate in its learning, with new entities and reinforcement of existing entities learned with every broadcast. Those that are sufficiently reinforced (tested) remain. The others decay away as "errors."

7.4.2 Newell's functional criteria (adapted from Lebiere and Anderson 2003)

Newell proposed multiple criteria that a human cognitive architecture should satisfy in order to be functional [57, 58]. Lebiere and Anderson [51] combined his two overlapping lists into the twelve criteria, phrased as questions, listed below. Each criterion described will be followed by an analysis of how LIDA does, or does not, satisfy it.

Flexible behavior: Does the architecture behave as an (almost) arbitrary function of the environment? Is the architecture computationally universal with failure?

This criterion demands flexibility of action selection. In LIDA, motivation for actions, learning and perceiving, come from feelings and emotions. These provide a much more flexible kind of motivation for action selection than do drives, causations or rules, producing more flexible action selection. In LIDA, various types of learning, including learning to recognize or perform procedures, also contribute to flexible behavior. LIDA's sophisticated action selection itself allows such flexibility as switching back and forth between various tasks. LIDA is flexible in what to attend to, at any given time, increasing the flexibility of action selection as well. We suspect that LIDA cannot do everything that can be done with a Turing machine; it is not computationally universal. We also suspect that this is not necessary for AGI.

Real-time operation: Does the architecture operate in real time? Given its timing assumptions, can it respond as fast as humans?

LIDA's cognitive cycles individually take approximately 300 ms, and they sample the environment cascading at roughly five to ten times per-second [59]. There is considerable empirical evidence from neuroscience suggestive of, and consistent with, such cognitive cycling in humans [60–66]. An earlier software agent, IDA (see above), based on the LIDA architecture, found new billets for sailors in about the same time as it took a human “detailer” [59].

Rationality: Does the architecture exhibit rational, i.e., effective adaptive behavior? Does the system yield functional behavior in the real world?

In the LIDA model, feelings and emotions play important role in decision making. The LIDA model can feature both the affective and rational human-inspired models of decision making [67]. LIDA's predecessor IDA, controlled by much the same architecture, was quite functional [68], promising the same for LIDA controlled agents.

Knowledgeable in terms of size: Can it use vast amounts of knowledge about the environment? How does the size of the knowledge base affect performance?

In the LIDA model, selective attention filters potentially large amounts of incoming sensory data. Selective attention also provides access to appropriate internal resources that allow the agent to select appropriate actions and to learn from vast amounts of data produced during interactions in a complex environment. The model has perceptual, episodic,

attentional and procedural memory for the long term storage of various kinds of information.

Knowledgeable in terms of variety: Does the agent integrate diverse knowledge? Is it capable of common examples of intellectual combination?

A major function of LIDA's preconscious Workspace is precisely to integrate diverse knowledge in the process of updating the Current Situational Model from which the contents of consciousness is selected. Long-term sources of this knowledge include Perceptual Associative Memory and Declarative Memory. There are several sources of more immediate knowledge that come into play. LIDA is, in principle, "capable of common examples of intellectual combination," though work has only begun on the first implemented LIDA based agent promising such capability.

Behaviorally robust: Does the agent behave robustly in the face of error, the unexpected, and the unknown? Can it produce cognitive agents that successfully inhabit dynamic environments?

LIDA's predecessor, IDA, was developed for the US Navy to fulfill tasks performed by human resource personnel called detailers. At the end of each sailor's tour of duty, he or she is assigned to a new billet. This assignment process is called distribution. The Navy employs almost 300 full time detailers to effect these new assignments. IDA's task is to facilitate this process, by automating the role of detailer. IDA was tested by former detailers and accepted by the Navy [68].

Linguistic: Does the agent use (natural) language? Is it ready to take a test of language proficiency?

IDA communicates with sailors by email in unstructured English. However, we think of this capability as pseudo-natural-language, since it is accomplished only due to the relatively narrow domain of discourse. Language comprehension and language production are high-level cognitive processes in humans. In the LIDA model, such higher-level processes are distinguished by requiring multiple cognitive cycles for their accomplishment. In LIDA, higher-level cognitive processes can be implemented by one or more behavior streams; that is, streams of instantiated schemes and links from procedural memory. Thus LIDA should, in principle, be capable of natural language understanding and production. In practice, work on natural language has just begun.

Self-awareness: Does the agent exhibit self-awareness and a sense of self? Can it produce functional accounts of phenomena that reflect consciousness?

Researchers in, philosophy, neuroscience and psychology postulate various forms of a “self” in humans and animals. All of these selves seem to have a basis in some form of consciousness. GWT suggests that a self-system can be thought of “... as the dominant context of experience and action.” [2, Chapter 9]. Following others (see below) the various selves in an autonomous agent may be categorized into three major components, namely: 1) the Proto-Self; 2) the Minimal (Core) Self; and 3) the Extended Self. The LIDA model provides for the basic building blocks from which to implement the various parts of a multi-layered self-system as hypothesized by philosophers, psychologists and neuroscientists [69]. In the following, we discuss each component, and their sub-selves, very briefly (for more detail see [69]).

1) The Proto-Self: Antonio Damasio conceived the Proto-self as a short-term collection of neural patterns of activity representing the current state of the organism [70]. In LIDA, the Proto-self is implemented as the set of global and relevant parameters in the various modules including the Action Selection and the memory systems, and the underlying computer system’s memory and operating system; **2) the Minimal (Core) Self:** The minimal or core self [71] is continually regenerated in a series of pulses, which blend together to give rise to a continuous stream of consciousness. The Minimal Self can be implemented as sets of entities in the LIDA ontology, that is, as computational collections of nodes in the slipnet of LIDA’s perceptual associative memory; and **3) the Extended Self:** The extended self consists of (a) the autobiographical self, (b) the self-concept, (c) the volitional or executive self, and (d) the narrative self. In human beings, the autobiographical self develops directly from episodic memory. In the LIDA model, the autobiographical self can be described as the local associations from transient episodic memory and declarative memory which come to the workspace in every cognitive cycle. The self-concept consists of enduring self-beliefs and intentions, particularly those relating to personal identity and properties. In the LIDA model, the agent’s beliefs are in the semantic memory and each volitional goal has an intention codelet. The volitional self provides executive function. In the LIDA model, deliberate actions are implemented by behavior streams. Thus, LIDA has a volitional self. The narrative self is able to report actions, intentions, etc., sometimes equivocally, contradictorily or self-deceptively. In the LIDA model, feeling, motivation, and attention nodes play a very important role in the Narrative Self. That is, after understanding a self-report request, the LIDA model could, in principle, generate a report based on its understanding of such a request.

Adaptive through learning: Does the agent learn from its environment? Can it produce the variety of human learning?

LIDA is equipped with perceptual, episodic, procedural, attentional learning, all modulated by feelings and emotions. As humans do, LIDA learns continually and implicitly with each conscious broadcast in each cognitive cycle.

Developmental: Does the agent acquire capabilities through development? Can it account for developmental phenomena?

Since LIDA learns as humans do, we would expect a LIDA controlled agent to go through a developmental period of rapid learning as would a child. The work on replicating data from developmental experiments has just begun.

Evolvable: Can the agent arise through evolution? Does the theory relate to evolutionary and comparative considerations?

Since LIDA is attempted to model humans and other animals, presumably the model should be evolvable and comparative, at least in principle.

Be realizable within the brain: Do the components of the theory exhaustively map onto brain processes?

Shanahan [6] devotes two chapters to a compelling argument that the brain is organized so as to support a conscious broadcast. LIDA is beginning to build on this insight, using the work of Freeman and colleagues [72], to create a non-linear dynamical systems bridge between LIDA and the underlying brain. Whether this bridge will lead to an exhaustive mapping is not at all clear as yet.

7.4.3 BICA table

Any AGI is likely to be produced by a very diverse collection of cognitive modules and their processes. There is a computational framework for LIDA [24] that requires only a modest commitment to the underlying assumptions of the LIDA architecture. One can introduce into LIDA's framework a large variety of differently implemented modules and processes so that, many possible AGI architectures could be implemented from the LIDA framework. One advantage of doing it in this way is that all of these AGI architectures implemented on the top of the LIDA's framework, would use a common ontology based on the LIDA model as presented to AGI 2011 [24].

In the following, we will give an assessment of the LIDA model against the features of the BICA Table of Implemented Cognitive Architectures [52]. Column 1 of the BICA Table contains a list of features proposed by developers of cognitive architectures to be at least potentially useful, if not essential, for the support of an AGI. Subsequent columns are devoted to individual cognitive architectures with a cell describing how its column architecture addresses its row feature. The rest of this section is an expansion of the column devoted to LIDA in the BICA table.

Note that all the **Basic overview** features listed in the BICA's first column are detailed earlier in this chapter. We will discuss the rest of the features in the following:

Support for Common Components: *The LIDA model supports all features mentioned in this part such as episodic and semantic memories. However, the auditory mechanism is not implemented in a LIDA-based agent as yet.*

Support for Common Learning Algorithms: *The LIDA model supports different types of learning such as episodic, perceptual, procedural, and attentional learning. However, the Bayesian Update and Gradient Descent Methods (e.g., Backpropagation) are not implemented in a LIDA-based agent.*

Common General Paradigms Modeled: *The LIDA model supports features listed in this part such as decision making and problem solving. However, perceptual illusions, meta-cognitive tasks, social psychology tasks, personality psychology tasks, motivational dynamics are not implemented in a LIDA-based agent.*

Common Specific Paradigms Modeled columns: 1) Stroop; 2) Task Switching; 3) Tower of Hanoi/London; 4) Dual Task; 5) *N*-Back; 6) Visual perception with comprehension; 7) Spatial exploration; 8) Learning and navigation; 9) Object/feature search in an environment; 10) Learning from instructions; 11) Pretend-play.

Although the Common Specific Paradigms Modeled features listed above are not implemented in LIDA, in principle LIDA is capable of implementing each of them. For instance, a LIDA-based agent is replicating some attentional tasks à la Van Bockstaele's and his colleagues [73].

Meta-Theoretical Questions:

- 1) Uses only local computations? *Yes, throughout the architecture with the one exception of the conscious broadcast which is global;*
- 2) Unsupervised learning? *Yes. The LIDA model supports four different modes of learning, perceptual, episodic, attentional and procedural;*

- 3) Supervised learning? *While in principle possible for a LIDA agent, supervised learning per se is not part of the architecture;*
- 4) Can it learn in real time? *Yes (see above);*
- 5) Can it do fast stable learning; i.e., adaptive weights converge on each trial without forcing catastrophic forgetting? *Yes. One shot learning in several modes occurs with the conscious broadcast during each cognitive cycle. With sufficient affective support and/or sufficient repeated attention, such learning can be quite stable;*
- 6) Can it function autonomously? *Yes. A LIDA-based agent can, in principle, operate machines and drive vehicles autonomously;*
- 7) Is it general-purpose in its modality; i.e., is it brittle? *A LIDA-based agent can, in principle, be developed to be general purpose and robust in real world environments;*
- 8) Can it learn from arbitrarily large databases; i.e., not toy problems? *Yes, this question is already answered in the previous sections;*
- 9) Can it learn about non-stationary databases; i.e., environmental rules change unpredictably? *Yes, a LIDA-based agent is, in principle, capable of working properly in an unpredictable environment;*
- 10) Can it pay attention to valued goals? *Yes, already explained earlier in this chapter;*
- 11) Can it flexibly switch attention between unexpected challenges and valued goals? *Yes. A LIDA-based agent attends to what is most salient based on its situational awareness;*
- 12) Can reinforcement learning and motivation modulate perceptual and cognitive decision-making? *Yes;*
- 13) Can it adaptively fuse information from multiple types of sensors and modalities? *In principle, yes, but it has yet to be implemented in particular domains with multiple senses.*

7.5 Discussion, Conclusions

In this chapter, we argue that LIDA may be suitable as an underlying cognitive architecture on which others might build an AGI. Our arguments rely mostly on an analysis of how LIDA satisfies Sun's "desiderata for cognitive architectures" as well as Newell's "test for a theory of cognition." We also measured LIDA against the architectural features listed in the BICA Table of Implemented Cognitive Architectures, as well as to the anticipated needs of AGI developers.

As can be seen in Section 7.4 above, the LIDA model seems to meet all of Sun's "... essential desiderata for developing cognitive architectures," and Newell's criteria that a human cognitive architecture should satisfy in order to be functional. In addition, the LIDA architecture seems to be able, at least in principle, of incorporating each of the features listed in the BICA Table of Implemented Cognitive Architectures. Thus the LIDA architecture would seem to offer the requisite breadth of features.

The LIDA computational framework offers software support for the development of LIDA based software agents, as well as LIDA based control systems for autonomous robots [24]. As described in Section 7.2 above, developing an AGI based loosely on the LIDA architecture requires only a modest commitment. Higher-level cognitive processes such as reasoning, planning, deliberation, etc., must be implemented by behavior streams, that is, using cognitive cycles. But this is not a strong commitment, since, using the LIDA computational framework, any individual module in LIDA's cognitive cycle can be modified at will, or even replaced by another designed by the AGI developer. Thus we are left with the contention that various AGI systems can effectively be developed based loosely on the LIDA architecture and its computational framework. Such systems would lend themselves to relatively easy incremental improvements by groups of developers and, due to their common foundation and ontology, would also allow relatively straightforward testing and comparison. Thus the LIDA architecture would seem to be an ideal starting point for the development of AGI systems.

Bibliography

- [1] B.J. Baars. *In the Theater of Consciousness: The Workspace of the Mind*. Oxford: Oxford University Press (1997).
- [2] B.J. Baars. *A cognitive theory of consciousness*. Cambridge: Cambridge University Press (1988).
- [3] B.J. Baars. The conscious access hypothesis: origins and recent evidence. *Trends in Cognitive Science* 47–52 (2002).
- [4] S. Dehaene & L. Naccache. Towards a cognitive neuroscience of consciousness: basic evidence and a workspace framework. *Cognition* 79, 1–37 (2001).
- [5] N. Kanwisher. Neural events and perceptual awareness. *Cognition* 79:89, 89–113 (2001).
- [6] M. Shanahan. *Embodiment and the Inner Life*. Oxford: Oxford University Press (2010).
- [7] S. Franklin. IDA: A Conscious Artifact? *Journal of Consciousness Studies* 10, 47–66 (2003).
- [8] S. Franklin & F.G.J. Patterson. The LIDA Architecture: Adding New Modes of Learning to an Intelligent, Autonomous, Software Agent. *Integrated Design and Process Technology, IDPT-2006, San Diego, CA, Society for Design and Process Science* (2006).
- [9] S. Franklin, A. Kelemen & L. McCauley. IDA: A Cognitive Agent Architecture. *IEEE Conf. on Systems, Man and Cybernetics*, 2646–2651 (1998).

- [10] F.J. Varela, E. Thompson & E. Rosch. *The embodied mind: Cognitive Science and Human Experience*. MIT Press, Cambridge, MA, USA (1991).
- [11] L.W. Barsalou. *Perceptual Symbol Systems*. Vol. 22 (MA: The MIT Press, 1999).
- [12] A.D. Baddeley. The episodic buffer: a new component of working memory? *Trends in Cognitive Science* 4, 417–423 (2000).
- [13] A.M. Glenberg. What memory is for. *Behavioral and Brain Sciences*, 1–19 (1997).
- [14] K.A. Ericsson & W. Kintsch. Long-term working memory. *Psychological Review* 102, 21–245 (1995).
- [15] A. Sloman. What Sort of Architecture is Required for a Human-like Agent? In *Foundations of Rational Agency*, ed. M. Wooldridge, and A. Rao. Dordrecht, Netherlands: Kluwer Academic Publishers (1999).
- [16] D. Hofstadter, R & M. Mitchell. The Copycat Project: A model of mental fluidity and analogy-making In *Advances in Connectionist and Neural Computation theory, Vol. 2: Logical Connections*, ed. K.J. Holyoak, and J.A. Barnden, N.J. Norwood: Ablex. (1994).
- [17] J. Marshall. Metacat: A self-watching cognitive architecture for analogy-making. *Proceedings of the 24th Annual Conference of the Cognitive Science Society* 631–636 (2002).
- [18] P. Kanerva. *Sparse Distributed Memory*. Cambridge MA: The MIT Press (1988).
- [19] R.P.N. Rao & O. Fuentes. Hierarchical Learning of Navigational Behaviors in an Autonomous Robot using a Predictive Sparse Distributed Memory. *Machine Learning* 31, 87–113 (1998).
- [20] G.L. Drescher. *Made-Up Minds: A Constructivist Approach to Artificial Intelligence*. Cambridge, MA: MIT Press (1991).
- [21] H.H. Chaput, B. Kuipers & R. Miiikkulainen. Constructivist Learning: A Neural Implementation of the Schema Mechanism. *Workshop for Self-Organizing Maps, Kitakyushu, Japan* (2003).
- [22] P. Maes. How to do the right thing. *Connection Science* 1, 291–323 (1989).
- [23] R.A. Brooks. Intelligence without Representation. *Artificial intelligence*. Elsevier (1991).
- [24] J. Snaider, R. McCall & S. Franklin. The LIDA Framework as a General Tool for AGI. *Paper presented at the The Fourth Conference on Artificial General Intelligence, Mountain View, California, USA* (2011).
- [25] R.A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation* 2, pp. 14–23 (1986).
- [26] M.L. Anderson. Embodied Cognition: A Field Guide. *Artificial Intelligence* 149, 91–130 (2003).
- [27] T. Ziemke, J. Zlatev & R. M. Frank. *Body, Language and Mind: Volume 1: Embodiment*. (Mouton de Gruyter, 2007).
- [28] S. Harnad. The Symbol Grounding Problem. *Physica D* 42, 335–346 (1990).
- [29] M. de Vega, A. Glenberg & A. Graesser. *Symbols and Embodiment: Debates on meaning and cognition*. Oxford: Oxford University Press (2008).
- [30] B.J. Baars & S. Franklin. How conscious experience and working memory interact. *Trends in Cognitive Sciences* 7 (2003).
- [31] S. Franklin, B.J. Baars, U. Ramamurthy & M. Ventura. The Role of Consciousness in Memory. *Brains, Minds and Media* 1, *bmm150* (urn:nbn:de:0009-3-1505) (2005).
- [32] R. McCall, S. Franklin & D. Friedlander. Grounded Event-Based and Modal Representations for Objects, Relations, Beliefs, Etc. *Paper presented at the FLAIRS-23, Daytona Beach, FL* (2010).
- [33] A.D. Baddeley. Working memory and conscious awareness. In *Theories of memory*, (eds. Alan Collins, S. Gathercole, M. A Conway, & P. Morris) 11–28 (Erlbaum, 1993).
- [34] S. Franklin. Cognitive Robots: Perceptual associative memory and learning. In *Proceedings of the 14th Annual International Workshop on Robot and Human Interactive Communication* (2005).

- [35] E. Tulving. *Elements of Episodic Memory*. New York: Oxford University Press (1983).
- [36] M.A. Conway. Sensory-perceptual episodic memory and its context: autobiographical memory. In *Episodic Memory*, ed. A. Baddeley, M. Conway, and J. Aggleton. Oxford: Oxford University Press (2002).
- [37] R. Stickgold & M.P. Walker. Memory consolidation and reconsolidation: what is the role of sleep? *Trends Neurosci* **28**, 408–415 (2005).
- [38] W.K. Estes. *Classification and Cognition*. Oxford: Oxford University Press (1993).
- [39] Z. Vidnyánszky & W. Sohn. Attentional learning: learning to bias sensory competition. *Journal of Vision* **3** (2003).
- [40] G.L. Drescher. Learning from Experience Without Prior Knowledge in a Complicated World. *Proceedings of the AAAI Symposium on Parallel Models*. AAAI Press (1988).
- [41] A. Negatu & S. Franklin. An action selection mechanism for ‘conscious’ software agents. *Cognitive Science Quarterly* **2**, 363–386 (2002).
- [42] P. Rosenbloom, J. Laird & A. Newell. *The Soar Papers: Research on Integrated Intelligence*. Cambridge, Massachusetts: MIT Press (1993).
- [43] J.E. Laird, A. Newell & P.S. Rosenbloom. Soar: an architecture for general intelligence. *Artificial Intelligence* **33**, 1–64 (1987).
- [44] J.F. Lehman, J.E. Laird & P.S. Rosenbloom. A gentle introduction to Soar, an architecture for human cognition. In *Invitation to Cognitive Science Methods, Models, and Conceptual Issues*, Vol. **4** (eds. S. Sternberg & D. Scarborough) (MA: MIT Press, 1998).
- [45] J.R. Anderson. *Rules of the mind*. (Mahwah, NJ: Lawrence Erlbaum Associates, 1993).
- [46] J.R. Anderson. *The Architecture of Cognition*. Cambridge, MA: Harvard University Press (1983).
- [47] J.R. Anderson, D. Bothell, M.D. Byrne, S. Douglass, C. Lebiere & Y. Qin. An integrated theory of the mind. *Psychological Review* **111**, 1036–1060 (2004).
- [48] R. Sun. *The CLARION cognitive architecture: Extending cognitive modeling to social simulation Cognition and Multi-Agent interaction*. Cambridge University Press, New York (2006).
- [49] U. Faghihi. *The use of emotions in the implementation of various types of learning in a cognitive agent*. Ph.D thesis, University of Quebec at Montreal (UQAM), (2011).
- [50] D. Vernon, G. Metta & G. Sandini. A Survey of Artificial Cognitive Systems: Implications for the Autonomous Development of Mental Capabilities in Computational Agents. *IEEE Transactions on Evolutionary Computation, Special Issue on Autonomous Mental Development* **11**, 151–180 (2007).
- [51] J.R. Anderson & C. Lebiere. The Newell Test for a theory of cognition. *Behavioral And Brain Sciences* **26** (2003).
- [52] A.V. Samsonovich. Toward a Unified Catalog of Implemented Cognitive Architectures. *Proceeding of the 2010 Conference on Biologically Inspired Cognitive Architectures*, 195–244 (2010).
- [53] R. Sun. Desiderata for cognitive architectures. *Philosophical Psychology* **17**, 341–373 (2004).
- [54] S. Franklin & M.H. Ferkin. An Ontology for Comparative Cognition: a Functional Approach. *Comparative Cognition & Behavior Reviews* **1**, 36–52 (2006).
- [55] S. D’Mello & S. Franklin. A cognitive model’s view of animal cognition. *Cognitive models and animal cognition. Current Zoology*. (in press) (2011).
- [56] J. Snaider, R. McCall & S. Franklin. Time production and representation in a conceptual and computational cognitive model. *Cognitive Systems Research*. (in press).
- [57] A. Newell. *Unified Theory of Cognition*. Cambridge, MA: Harvard University Press (1990).
- [58] A. Newell. Precis of Unified theories of cognition. *Behavioral and Brain Sciences* (1992).
- [59] T. Madl, B.J. Baars & S. Franklin. The Timing of the Cognitive Cycle. *PLoS ONE* (2011).
- [60] S. Doesburg, J. Green, J. McDonald & L. Ward. Rhythms of consciousness: binocular rivalry reveals large-scale oscillatory network dynamics mediating visual perception. *PLoS One*. **4**:

- c6142 (2009).
- [61] W. Freeman. The limbic action-perception cycle controlling goal-directed animal behavior. *Neural Networks* **3**, 2249–2254 (2002).
- [62] J. Fuster. Upper processing stages of the perception-action cycle. *Trends in Cognitive Sciences* **8**, 143–145 (2004).
- [63] M. Massimini, F. Ferrarelli, R. Huber, S.K. Esser & H. Singh. Breakdown of Cortical Effective Connectivity During Sleep. *Science* **309** (2005).
- [64] M. Sigman & S. Dehaene. Dynamics of the Central Bottleneck: Dual-Task and Task Uncertainty. *PLoS Biol.* **4** (2006).
- [65] N. Uchida, A. Kepecs & Z. F. Mainen. Seeing at a glance, smelling in a whiff: rapid forms of perceptual decision making. *Nature Reviews Neuroscience* **7**, 485–491 (2006).
- [66] J. Willis & A. Todorov. First Impressions: Making Up Your Mind After a 100-Ms Exposure to a Face. *Psychological Science* **17**, 592–599 (2006).
- [67] W. Wallach, S. Franklin & C. Allen. In *Topics in Cognitive Science, special issue on Cognitive Based Theories of Moral Decision Making* (eds. W. Wallach & S. Franklin) 454–485 (Cognitive Science Society, 2010).
- [68] L. McCauley & S. Franklin. A Large-Scale Multi-Agent System for Navy Personnel Distribution. *Connection Science* **14**, 371–385 Comments: special issue on agent autonomy and groups. (2002).
- [69] U. Ramamurthy & S. Franklin. Self System in a model of Cognition. *Proceedings of Machine Consciousness Symposium at the Artificial Intelligence and Simulation of Behavior Convention (AISB'11)*, University of York, UK, 51–54 (2011).
- [70] A.R. Damasio. *The Feeling of What Happens: Body and Emotion in the Making of Consciousness*. New York: Harcourt Inc (1999).
- [71] S. Gallagher. Philosophical conceptions of the self: implications for cognitive science. *Trends in Cognitive Science* **4**, 14–21 (2000).
- [72] C. Skarda & W.J. Freeman. How Brains Make Chaos in Order to Make Sense of the World. *Behavioral and Brain Sciences* **10**, 161–195 (1987).
- [73] B. Van Bockstaele, B. Verschuere, J.D. Houwer & G. Crombez. On the costs and benefits of directing attention towards or away from threat-related stimuli: A classical conditioning experiment. *Behaviour Research and Therapy* **48**, 692–697 (2010).

Chapter 8

The Architecture of Human-Like General Intelligence

Ben Goertzel¹, Matt Iklé² and Jared Wigmore³

¹ *Novamente LLC, 1405 Bernerd Place, Rockville MD*

² *Dept. of Mathematics and Computing, Adams State College, Alamosa CO*

³ *School of Design, Hong Kong Polytechnic University*

ben@goertzel.org

By exploring the relationships between different AGI architectures, one can work toward a holistic cognitive model of human-level intelligence. In this vein, here an integrative architecture diagram for human-like general intelligence is proposed, via merging of lightly modified version of prior diagrams including Aaron Sloman’s high-level cognitive model, Stan Franklin and the LIDA group’s model of working memory and the cognitive cycle, Joscha Bach and Dietrich Dörner’s Psi model of motivated action and cognition, James Albus’s three-hierarchy intelligent robotics model, and the author’s prior work on cognitive synergy in deliberative thought and metacognition, along with ideas from deep learning and computational linguistics. The purpose is not to propose an actual merger of the various AGI systems considered, but rather to highlight the points of compatibility between the different approaches, as well as the differences of both focus and substance. The result is perhaps the most comprehensive architecture diagram of human-cognition yet produced, tying together all key aspects of human intelligence in a coherent way that is not tightly bound to any particular cognitive or AGI theory. Finally, the question of the dynamics associated with the architecture is considered, including the potential that human-level intelligence requires cognitive synergy between these various components is considered; and the possibility of a “trickiness” property causing the intelligence of the overall system to be badly suboptimal if any of the components are missing or insufficiently cooperative. One idea emerging from these dynamic consideration is that implementing the *whole* integrative architecture diagram may be necessary for achieving anywhere near human-level, human-like general intelligence.

8.1 Introduction

Cognitive science appears to have a problem with integrative understanding. Over the last few decades, cognitive science has discovered a great deal about the structures and

dynamics underlying the human mind. However, as in many other branches of science, there has been more focus on detailed analysis of individual aspects, than on unified holistic understanding. As a result of this tendency, there are not many compelling examples of holistic “cognitive architecture diagrams” for human intelligence – diagrams systematically laying out all the pieces needed to generate human intelligence and how they interact with each other.

Part of the reason why global human cognitive architecture diagrams are not so common is, of course, a lack of agreement in the field regarding all the relevant issues. Since there are multiple opinions regarding nearly every aspect of human intelligence, it would be difficult to get two cognitive scientists to fully agree on every aspect of an overall human cognitive architecture diagram. Prior attempts to outline detailed mind architectures have tended to follow highly specific theories of intelligence, and hence have attracted only moderate interest from researchers not adhering to those theories. An example is Minsky’s work presented in *The Emotion Machine* [13], which arguably does constitute an architecture diagram for the human mind, but which is only loosely grounded in current empirical knowledge and stands more as a representation of Minsky’s own intuitive understanding.

On the other hand, AGI appears to have a problem with the mutual comprehensibility and comparability of different research approaches. The AGI field has in recent years seen a proliferation of cognitive architectures, each purporting to cover all aspects needed for the creation of human-level general intelligence. However, the differences of language and focus among the various approaches has often made it difficult for researchers to fully understand each others’ work, let alone collaborate effectively.

This chapter describes a conceptual experiment aimed at addressing both of the above problems together. We aim to present a coherent, overall architecture diagram for human, and human-like, general intelligence, via combining the architecture diagrams associated with a number of contemporary AGI architectures. While the exercise is phrased in terms of diagrams, of course the underlying impetus is conceptual integration; and our hope is that the exercise described here will serve as a starting point for ongoing exploration of the relationships between multiple AGI architectures and cognitive theories.

The architecture diagram we give here does not reflect our own idiosyncratic understanding of human intelligence, as much as a combination of understandings previously presented by multiple researchers (including ourselves), arranged according to our own taste in a manner we find conceptually coherent. With this in mind, we call it “the integrative diagram” (a longer, grander and more explanatory name would be “The First Integra-

tive Human-Like Cognitive Architecture Diagram”). We have made an effort to ensure that as many pieces of the integrative diagram as possible are well grounded in psychological and even neuroscientific data, rather than mainly embodying speculative notions; however, given the current state of knowledge, this could not be done to a complete extent, and there is still some speculation involved here and there.

While based largely on understandings of human intelligence, the integrative diagram is intended to serve as an architectural outline for human-like general intelligence more broadly. For example, the OpenCog AGI architecture which we have co-created is explicitly not intended as a precise emulation of human intelligence, and does many things quite differently than the human mind, yet can still fairly straightforwardly be mapped into the integrative diagram.

Finally, having presented the integrative diagram which focuses on *structure*, we present some comments on the dynamics corresponding to that structure, focusing on the notion of *cognitive synergy*: the hypothesis that multiple subsystems of a generally intelligent system, focused on learning regarding different sorts of information, must interact in such a way as to actively aid each other in overcoming combinatorial explosions. This represents a fairly strong hypothesis regarding how the different components in the integrative diagram interact with each other. Further, it gives a way of addressing one of the more vexing issues in the AGI field: the difficulty of measuring partial progress toward human-level AGI. We will conjecture that this difficulty is largely attributable to a “trickiness” property of cognitive synergy in the integrative diagram. One of the upshots of our discussion of dynamics is: We consider it likely that to achieve anything remotely like human-like general intelligence, it will be necessary to implement basically all the components in the integrative diagram, in a thorough and richly interconnected way. Implementing half the boxes in the diagram is not likely to get us to a system with half the general intelligence of a human. The architecture of human-like cognition is a richly interconnected whole.

8.2 Key Ingredients of the Integrative Human-Like Cognitive Architecture Diagram

In assembling the integrative diagram, we have drawn on the work of researchers at the intersection of AGI and cognitive science – that is, researchers largely motivated by human cognitive science and neuroscience, but with aims of producing comprehensive architectures for human-like AGI. The main ingredients used in assembling the diagram are:

- Aaron Sloman’s high-level architecture diagram of human intelligence [14], drawn from his CogAff architecture, which it strikes us as a particularly clear embodiment of “modern common sense” regarding the overall architecture of the human mind. We have added only a couple items to Sloman’s high-level diagram, which we felt deserved an explicit high-level role that he did not give them: emotion, language and reinforcement.
- The LIDA architecture diagram presented by Stan Franklin and Bernard Baars [3]. We think LIDA is an excellent model of working memory and what Sloman calls “reactive processes”, with well-researched grounding in the psychology and neuroscience literature. We have adapted the LIDA diagram only very slightly for use here, changing some of the terminology on the arrows, and indicating where parts of the LIDA diagram indicate processes elaborated in more detail elsewhere in the integrative diagram.
- The architecture diagram of the Psi model of motivated cognition, presented by Joscha Bach in [4] based on prior work by Dietrich Dörner [5]. This diagram is presented without significant modification; however it should be noted that Bach and Dörner present this diagram in the context of larger and richer cognitive models, the other aspects of which are not all incorporated in the integrative diagram.
- James Albus’s three-hierarchy model of intelligence [1], involving coupled perception, action and reinforcement hierarchies. Albus’s model, utilized in the creation of intelligent unmanned automated vehicles, is a crisp embodiment of many ideas emergent from the field of intelligent control systems.
- Deep learning networks as a model of perception (and action and reinforcement learning), as embodied for example in the work of Itamar Arel [2] and Jeff Hawkins [10]. The integrative diagram adopts this as the basic model of the perception and action subsystems of human intelligence. Language understanding and generation are also modeled according to this paradigm.
- The OpenCog [7] integrative AGI architecture (in which we have played a key role), which places greatest focus on various types of long-term memory and their interrelationship, and is used mainly to guide the integrative architecture’s treatment of these matters.

Most of these ingredients could be interpreted as holistic explanations of human-like intelligence on their own. However, each of them places a focus in a different place, more elaborated in some regards than others. So it is interesting to collage the architecture diagrams from the different approaches together, and see what results. The product of this

exercise does not accord precisely with any of the component AGI architectures, and is not proposed as an architecture diagram for an AGI. However, we believe it has value as an exercise in integrative cognitive science. It is a mind-architecture diagram, drawing preferentially on different cognitive-science-inspired AGI approaches in those aspects where they have been most thoroughly refined.

One possible negative reaction to the integrative diagram might be to say that it's a kind of Frankenstein monster, piecing together aspects of different theories in a way that violates the theoretical notions underlying all of them! For example, the integrative diagram takes LIDA as a model of working memory and reactive processing, but from the papers on LIDA it's unclear whether the creators of LIDA construe it more broadly than that. The deep learning community tends to believe that the architecture of current deep learning networks, in itself, is close to sufficient for human-level general intelligence – whereas the integrative diagram appropriates the ideas from this community mainly for handling perception, action and language. Etc.

On the other hand, in a more positive perspective, one could view the integrative diagram as consistent with LIDA, but merely providing much more detail on some of the boxes in the LIDA diagram (e.g. dealing with perception and long-term memory). And one could view the integrative diagram as consistent with the deep learning paradigm – via viewing it, not as a description of components to be explicitly implemented in an AGI system, but rather as a description of the key structures and processes that must emerge in deep learning network, based on its engagement with the world, in order for it to achieve human-like general intelligence.

It seems to us that different communities of cognitive science and AGI researchers have focused on different aspects of intelligence, and have thus each created models that are more fully fleshed out in some aspects than others. But these various models all link together fairly cleanly, which is not surprising as they are all grounded in the same data regarding human intelligence. Many judgment calls must be made in fusing multiple models in the way that the integrative diagram does, but we feel these can be made without violating the spirit of the component models. In assembling the integrative diagram, we have made these judgment calls as best we can, but we're well aware that different judgments would also be feasible and defensible. Revisions are likely as time goes on, not only due to new data about human intelligence but also to evolution of understanding regarding the best approach to model integration.

Another possible argument against the ideas presented here is that there's nothing new – all the ingredients presented have been given before elsewhere. To this our retort is to quote Pascal: “Let no one say that I have said nothing new ... the arrangement of the subject is new.” The various architecture diagrams incorporated into the integrative diagram are either extremely high level (Sloman's diagram) or focus primarily on one aspect of intelligence, treating the others very concisely by summarizing large networks of distinction structures and processes in small boxes. The integrative diagram seeks to cover all aspects of human-like intelligence at a roughly equal granularity – a different arrangement.

8.3 An Architecture Diagram for Human-Like General Intelligence

The integrative diagram is presented here in a series of seven figures.

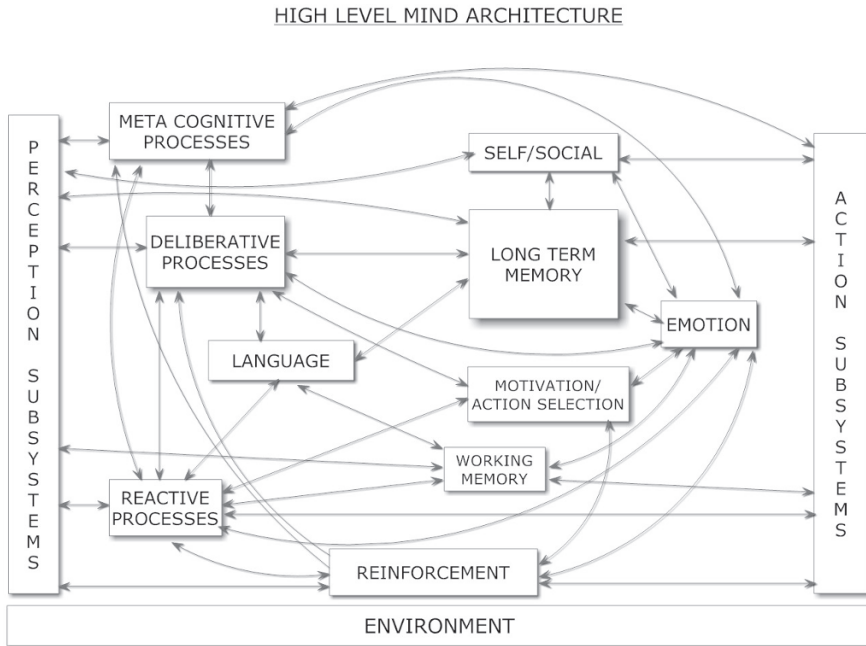


Fig. 8.1 High-Level Architecture of a Human-Like Mind

Figure 8.1 gives a high-level breakdown into components, based on Sloman's high-level cognitive-architectural sketch [14]. This diagram represents, roughly speaking, “modern common sense” about how a human-like mind is architected. The separation between

structures and processes, embodied in having separate boxes for Working Memory vs. Reactive Processes, and for Long Term Memory vs. Deliberative Processes, could be viewed as somewhat artificial, since in the human brain and most AGI architectures, memory and processing are closely integrated. However, the tradition in cognitive psychology is to separate out Working Memory and Long Term Memory from the cognitive processes acting thereupon, so we have adhered to that convention. The other changes from Sloman’s diagram are the explicit inclusion of language, representing the hypothesis that language processing is handled in a somewhat special way in the human brain; and the inclusion of a reinforcement component parallel to the perception and action hierarchies, as inspired by intelligent control systems theory (e.g. Albus as mentioned above) and deep learning theory. Of course Sloman’s high level diagram in its original form is intended as inclusive of language and reinforcement, but we felt it made sense to give them more emphasis.

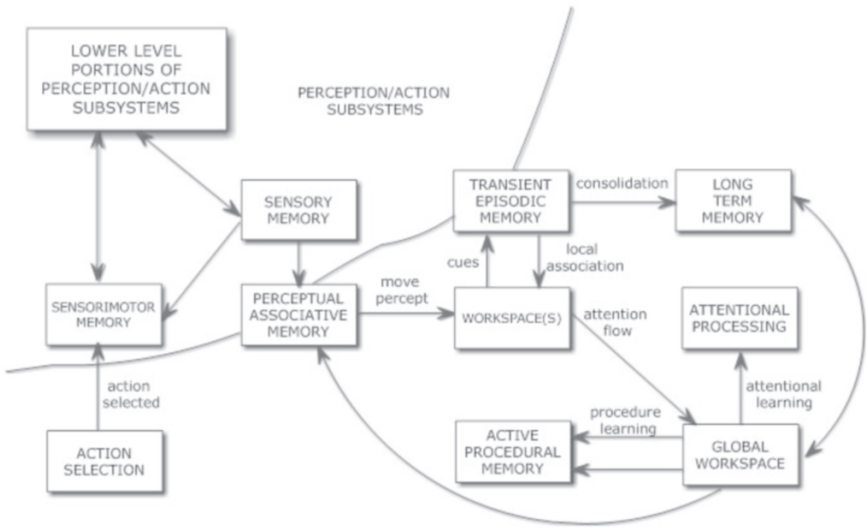


Fig. 8.2 Architecture of Working Memory and Reactive Processing, closely modeled on the LIDA architecture

Figure 8.2, modeling working memory and reactive processing, is essentially the LIDA diagram as given in prior papers by Stan Franklin, Bernard Baars and colleagues [3]. The boxes in the upper left corner of the LIDA diagram pertain to sensory and motor processing, which LIDA does not handle in detail, and which are modeled more carefully by deep learning theory. The bottom left corner box refers to action selection, which in the integrative diagram is modeled in more detail by Psi. The top right corner box refers to

Long-Term Memory, which the integrative diagram models in more detail as a synergetic multi-memory system (Figure 8.4).

The original LIDA diagram refers to various “codelets”, a key concept in LIDA theory. We have replaced “attention codelets” here with “attention flow”, a more generic term. We suggest one can think of an attention codelet as a piece of information that it’s currently pertinent to pay attention to a certain collection of items together.

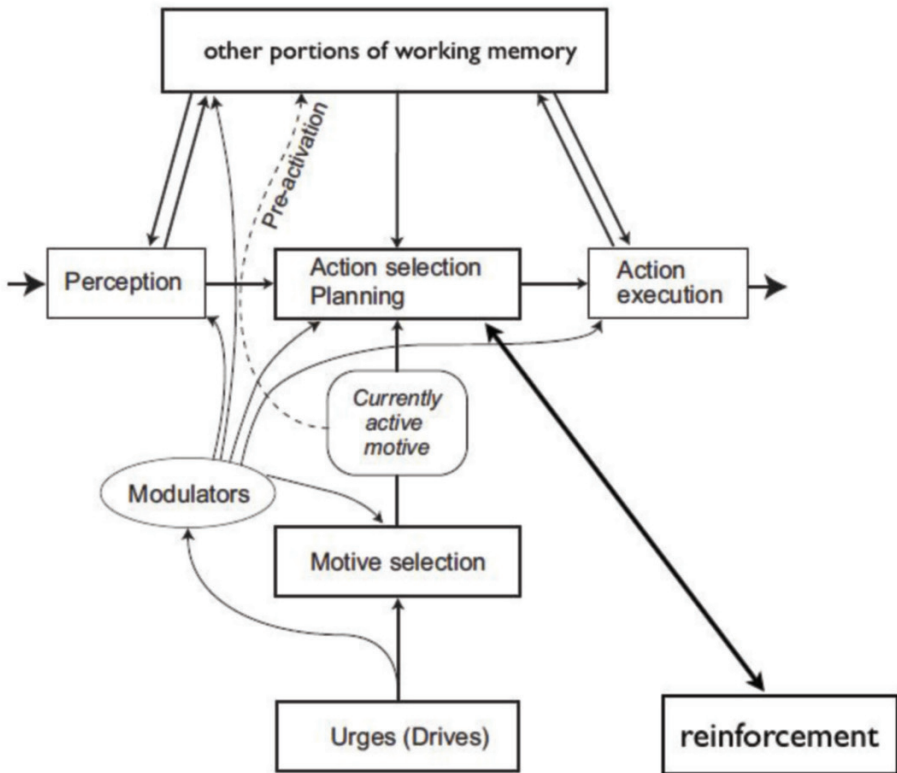


Fig. 8.3 Architecture of Motivated Action

Figure 8.3, modeling motivation and action selection, is a lightly modified version of the Psi diagram from Joscha Bach’s book *Principles of Synthetic Intelligence* [4]. The main difference from Psi is that in the integrative diagram the Psi motivated action framework is embedded in a larger, more complex cognitive model. Psi comes with its own theory of working and long-term memory, which is related to but different from the one given in the

integrative diagram – it views the multiple memory types distinguished in the integrative diagram as emergent from a common memory substrate. Psi comes with its own theory of perception and action, which seems broadly consistent with the deep learning approach incorporated in the integrative diagram. Psi’s handling of working memory lacks the detailed, explicit workflow of LIDA, though it seems broadly conceptually consistent with LIDA.

In Figure 8.3, the box labeled “Other parts of working memory” is labeled “Protocol and situation memory” in the original diagram. The Perception, Action Execution and Action Selection boxes have fairly similar semantics to the similarly labeled boxes in the LIDA-like Figure 8.2, so that these diagrams may be viewed as overlapping. The LIDA model doesn’t explain action selection and planning in as much detail as Psi, so the Psi-like Figure 8.3 could be viewed as an elaboration of the action-selection portion of the LIDA-like Figure 8.2. In Psi, reinforcement is considered as part of the learning process involved in action selection and planning; in Figure 8.3 an explicit “reinforcement box” has been added to the original Psi diagram, to emphasize this.

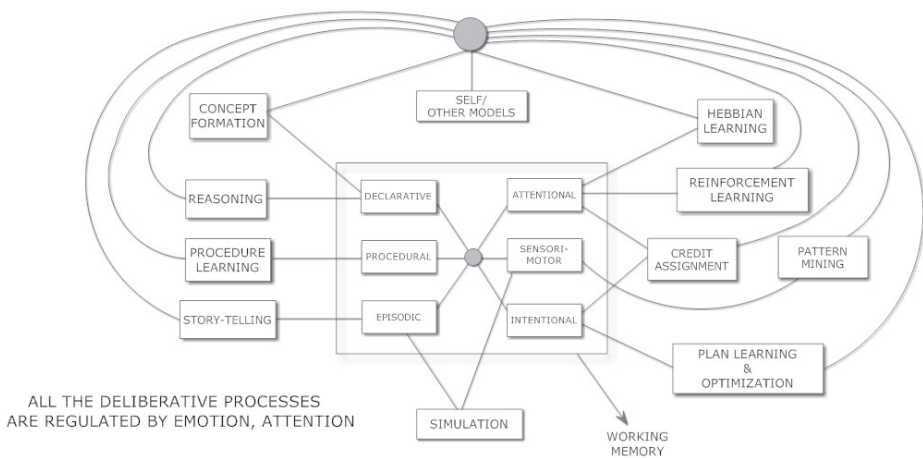


Fig. 8.4 Architecture of Long-Term Memory and Deliberative and Metacognitive Thinking

Figure 8.4, modeling long-term memory and deliberative processing, is derived from our own prior work studying the “cognitive synergy” between different cognitive processes associated with different types of memory. The division into types of memory is fairly standard. Declarative, procedural, episodic and sensorimotor memory are routinely distinguished; we like to distinguish attentional memory and intentional (goal) memory as well,

and view these as the interface between long-term memory and the mind's global control systems. One focus of our AGI design work has been on designing learning algorithms, corresponding to these various types of memory, that interact with each other in a synergistic way [7], helping each other to overcome their intrinsic combinatorial explosions. There is significant evidence that these various types of long-term memory are differently implemented in the brain, but the degree of structure and dynamical commonality underlying these different implementations remains unclear.

Each of these long-term memory types has its analogue in working memory as well. In some cognitive models, the working memory and long-term memory versions of a memory type and corresponding cognitive processes, are basically the same thing. OpenCog is mostly like this – it implements working memory as a subset of long-term memory consisting of items with particularly high importance values. The distinctive nature of working memory is enforced via using slightly different dynamical equations to update the importance values of items with importance above a certain threshold. On the other hand, many cognitive models treat working and long term memory as more distinct than this, and there is evidence for significant functional and anatomical distinctness in the brain in some cases. So for the purpose of the integrative diagram, it seemed best to leave working and long-term memory subcomponents as parallel but distinguished.

Figure 8.4 also encompasses metacognition, under the hypothesis that in human beings and human-like minds, metacognitive thinking is carried out using basically the same processes as plain ordinary deliberative thinking, perhaps with various tweaks optimizing them for thinking about thinking. If it turns out that humans have, say, a special kind of reasoning faculty exclusively for metacognition, then the diagram would need to be modified. Modeling of self and others is understood to occur via a combination of metacognition and deliberative thinking, as well as via implicit adaptation based on reactive processing.

Figure 8.5 models perception, according to the basic ideas of deep learning theory. Vision and audition are modeled as deep learning hierarchies, with bottom-up and top-down dynamics. The lower layers in each hierarchy refer to more localized patterns recognized in, and abstracted from, sensory data. Output from these hierarchies to the rest of the mind is not just through the top layers, but via some sort of sampling from various layers, with a bias toward the top layers. The different hierarchies cross-connect, and are hence to an extent dynamically coupled together. It is also recognized that there are some sensory modalities that aren't strongly hierarchical, e.g. touch and smell (the latter being better modeled as something like an asymmetric Hopfield net, prone to frequent chaotic dynam-

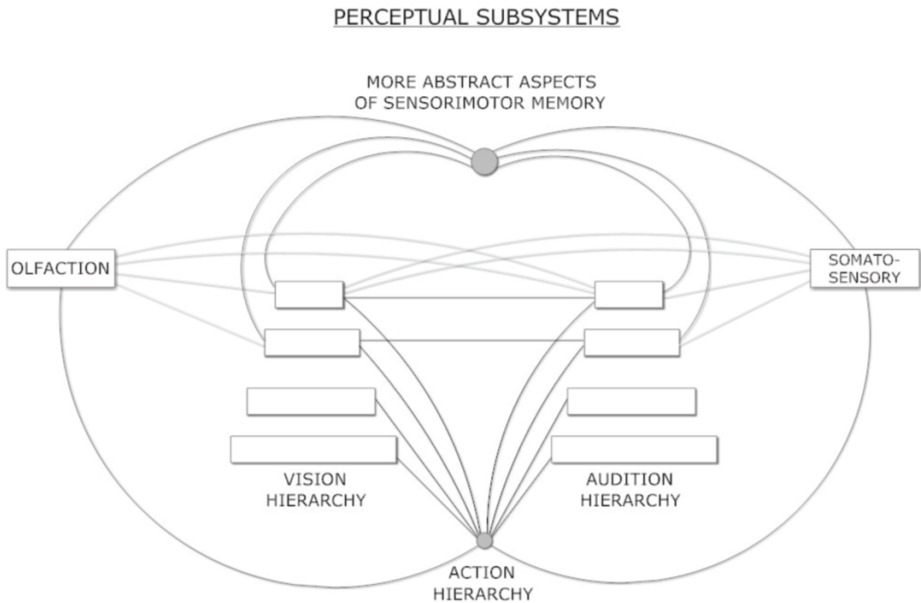


Fig. 8.5 Architecture for Multimodal Perception

ics [12]) – these may also cross-connect with each other and with the more hierarchical perceptual subnetworks. Of course the suggested architecture could include any number of sensory modalities; the diagram is restricted to four just for simplicity.

The self-organized patterns in the upper layers of perceptual hierarchies may become quite complex and may develop advanced cognitive capabilities like episodic memory, reasoning, language learning, etc. A pure deep learning approach to intelligence argues that all the aspects of intelligence emerge from this kind of dynamics (among perceptual, action and reinforcement hierarchies). Our own view is that the heterogeneity of human brain architecture argues against this perspective, and that deep learning systems are probably better as models of perception and action than of general cognition. However, the integrative diagram is not committed to our perspective on this – a deep-learning theorist could accept the integrative diagram, but argue that all the other portions besides the perceptual, action and reinforcement hierarchies should be viewed as descriptions of phenomena that emerge in these hierarchies due to their interaction.

Figure 8.6 shows an action subsystem and a reinforcement subsystem, parallel to the perception subsystem. Two action hierarchies, one for an arm and one for a leg, are shown for concreteness, but of course the architecture is intended to be extended more broadly. In

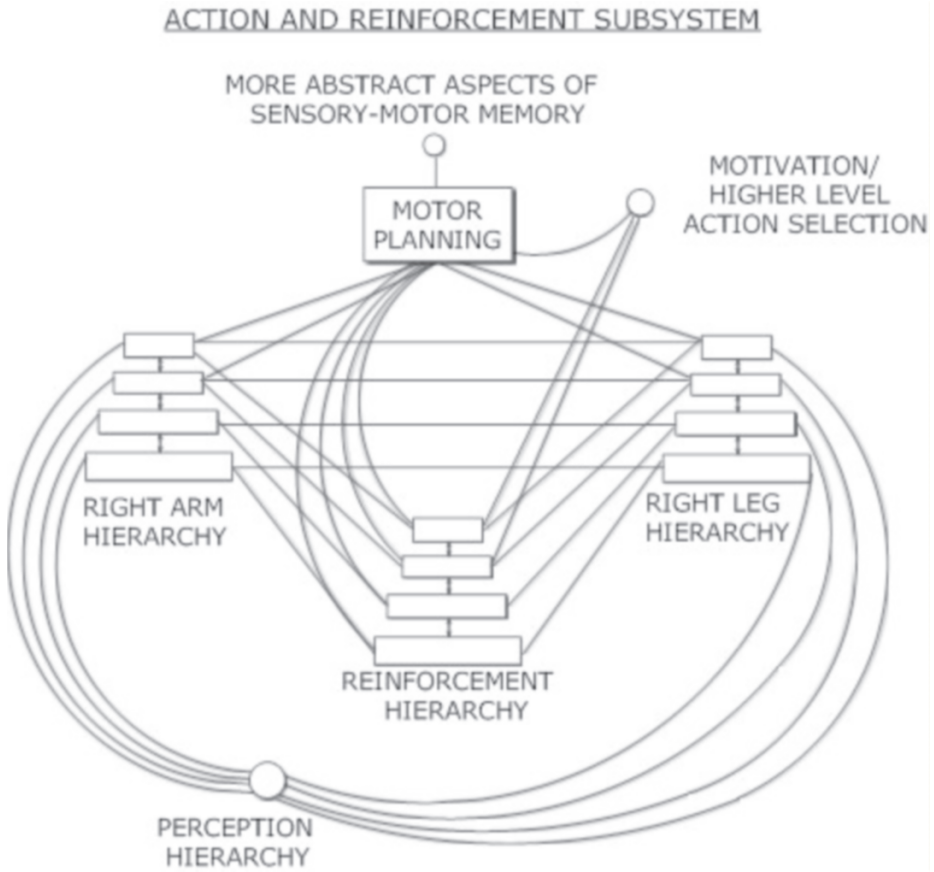


Fig. 8.6 Architecture for Action and Reinforcement

the hierarchy corresponding to an arm, for example, the lowest level would contain control patterns corresponding to individual joints, the next level up to groupings of joints (like fingers), the next level up to larger parts of the arm (hand, elbow). The different hierarchies corresponding to different body parts cross-link, enabling coordination among body parts; and they also connect at multiple levels to perception hierarchies, enabling sensorimotor coordination. Finally there is a module for motor planning, which links tightly with all the motor hierarchies, and also overlaps with the more cognitive, inferential planning activities of the mind, in a manner that is modeled different ways by different theorists. Albus [1] has elaborated this kind of hierarchy quite elaborately.

The reward hierarchy in Figure 8.6 provides reinforcement to actions at various levels on the hierarchy, and includes dynamics for propagating information about reinforcement up and down the hierarchy.

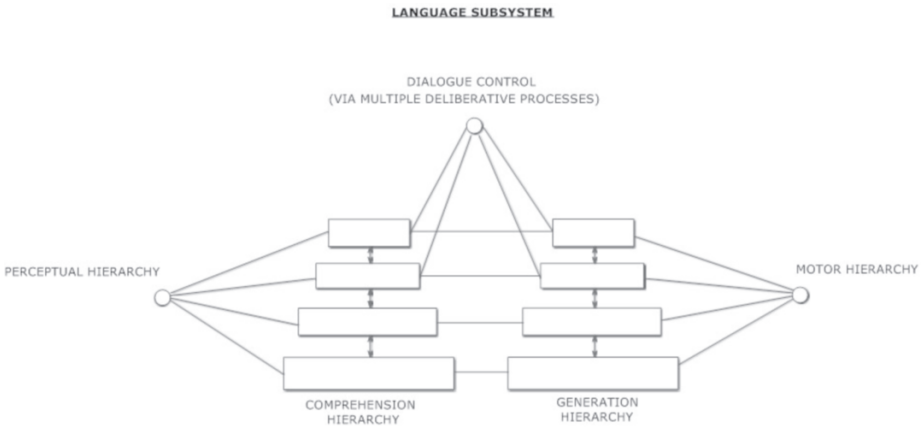


Fig. 8.7 Architecture for Language Processing

Figure 8.7 deals with language, treating it as a special case of coupled perception and action. The traditional architecture of a computational language comprehension system is a pipeline [9, 11], which is equivalent to a hierarchy with the lowest-level linguistic features (e.g. sounds, words) at the bottom, and the highest level features (semantic abstractions) at the top, and syntactic features in the middle. Feedback connections enable semantic and cognitive modulation of lower-level linguistic processing. Similarly, language generation is commonly modeled hierarchically, with the top levels being the ideas needing verbalization, and the bottom level corresponding to the actual sentence produced. In generation the primary flow is top-down, with bottom-up flow providing modulation of abstract concepts by linguistic surface forms.

So, that's it – an integrative architecture diagram for human-like general intelligence, split among 7 different pictures, formed by judiciously merging together architecture diagrams produced via a number of cognitive theorists with different, overlapping foci and research paradigms.

Is anything critical left out of the diagram? A quick perusal of the table of contents of cognitive psychology textbooks suggests to me that if anything major is left out, it's also unknown to current cognitive psychology. However, one could certainly make an argument for explicit inclusion of certain other aspects of intelligence, that in the integrative

diagram are left as implicit emergent phenomena. For instance, creativity is obviously very important to intelligence, but, there is no “creativity” box in any of these diagrams – because in our view, and the view of the cognitive theorists whose work we’ve directly drawn on here, creativity is best viewed as a process emergent from other processes that are explicitly included in the diagrams.

8.4 Interpretation and Application of the Integrative Diagram

A tongue-partly-in-cheek definition of a biological pathway is “a subnetwork of a biological network, that fits on a single journal page.” Cognitive architecture diagrams have a similar property – they are crude abstractions of complex structures and dynamics, sculpted in accordance with the size of the printed page, and the tolerance of the human eye for absorbing diagrams, and the tolerance of the human author for making diagrams.

However, sometimes constraints – even arbitrary ones – are useful for guiding creative efforts, due to the fact that they force choices. Creating an architecture for human-like general intelligence that fits in a few (okay, 7) fairly compact diagrams, requires one to make many choices about what features and relationships are most essential. In constructing the integrative diagram, we have sought to make these choices, not purely according to our own tastes in cognitive theory or AGI system design, but according to a sort of blend of the taste and judgment of a number of scientists whose views we respect, and who seem to have fairly compatible, complementary perspectives.

What is the use of a cognitive architecture diagram like this? It can help to give newcomers to the field a basic idea about what is known and suspected about the nature of human-like general intelligence. Also, it could potentially be used as a tool for cross-correlating different AGI architectures. If everyone who authored an AGI architecture would explain how their architecture accounts for each of the structures and processes identified in the integrative diagram, this would give a means of relating the various AGI designs to each other.

The integrative diagram could also be used to help connect AGI and cognitive psychology to neuroscience in a more systematic way. In the case of LIDA, a fairly careful correspondence has been drawn up between the LIDA diagram nodes and links and various neural structures and processes [6]. Similar knowledge exists for the rest of the integrative diagram, though not organized in such a systematic fashion. A systematic curation of links between the nodes and links in the integrative diagram and current neuroscience knowl-

edge, would constitute an interesting first approximation of the holistic cognitive behavior of the human brain.

Finally (and harking forward to the next section), the big omission in the integrative diagram is *dynamics*. Structure alone will only get you so far, and you could build an AGI system with reasonable-looking things in each of the integrative diagram's boxes, interrelating according to the given arrows, and yet still fail to make a viable AGI system. Given the limitations the real world places on computing resources, it's not enough to have adequate representations and algorithms in all the boxes, communicating together properly and capable doing the right things given sufficient resources. Rather, one needs to have all the boxes filled in properly with structures and processes that, when they act together using feasible computing resources, will yield appropriately intelligent behaviors via their cooperative activity. And this has to do with the complex interactive dynamics of all the processes in all the different boxes – which is something the integrative diagram doesn't touch at all. This brings us again to the network of ideas we've discussed under the name of “cognitive synergy,” to be discussed more extensively below.

It might be possible to make something similar to the integrative diagram on the level of dynamics rather than structures, complementing the structural integrative diagram given here; but this would seem significantly more challenging, because we lack a standard set of tools for depicting system dynamics. Most cognitive theorists and AGI architects describe their structural ideas using boxes-and-lines diagrams of some sort, but there is no standard method for depicting complex system dynamics. So to make a dynamical analogue to the integrative diagram, via a similar integrative methodology, one would first need to create appropriate diagrammatic formalizations of the dynamics of the various cognitive theories being integrated – a fascinating but onerous task.

When we first set out to make an integrated cognitive architecture diagram, via combining the complementary insights of various cognitive science and AGI theorists, we weren't sure how well it would work. But now we feel the experiment was generally a success – the resultant integrated architecture seems sensible and coherent, and reasonably complete. It doesn't come close to telling you everything you need to know to understand or implement a human-like mind – but it tells you the various processes and structures you need to deal with, and which of their interrelations are most critical. And, perhaps just as importantly, it gives a concrete way of understanding the insights of a specific but fairly diverse set of cognitive science and AGI theorists as complementary rather than contradictory.

8.5 Cognitive Synergy

The architecture of the mind, ultimately, has no meaning without an associated *dynamics*. Architecture emerges from dynamics, and channels dynamics. The cognitive dynamics of AGI systems is a large topic which we won't attempt to thoroughly pursue here, but we will mention one dynamical principle that we feel is essential for properly interpreting the integrative diagram: cognitive synergy.

Cognitive synergy has been proposed as a “general principle of feasible general intelligence” [8]. It is both a conceptual hypothesis about the structure of generally intelligent systems in certain classes of environments, and a design principle that one may use to guide the architecting of AGI systems.

First we review how cognitive synergy has been previously developed in the context of “multi-memory systems” – i.e., in the context of the diagram given above for long-term memory and deliberative processing, Figure 8.4. In this context, the cognitive synergy hypothesis states that human-like, human-level intelligent systems possess a combination of environment, embodiment and motivational system that makes it important for them to possess memories that divide into partially but not wholly distinct components corresponding to the categories such as:

- Declarative memory
- Procedural memory (memory about how to do certain things)
- Sensory and episodic memory
- Attentional memory (knowledge about what to pay attention to in what contexts)
- Intentional memory (knowledge about the system's own goals and subgoals)

The essential idea of cognitive synergy, in the context of multi-memory systems possessing the above memory types, may be expressed in terms of the following points:

- (1) Intelligence, relative to a certain set of environments, may be understood as the capability to achieve complex goals in these environments.
- (2) With respect to certain classes of goals and environments, an intelligent system requires a “multi-memory” architecture, meaning the possession of a number of specialized yet interconnected knowledge types, including: declarative, procedural, attentional, sensory, episodic and intentional (goal-related). These knowledge types may be viewed as different sorts of pattern that a system recognizes in itself and its environment.

- (3) Such a system must possess knowledge creation (i.e. pattern recognition / formation) mechanisms corresponding to each of these memory types. These mechanisms are also called “cognitive processes.”
- (4) Each of these cognitive processes, to be effective, must have the capability to recognize when it lacks the information to perform effectively on its own; and in this case, to dynamically and interactively draw information from knowledge creation mechanisms dealing with other types of knowledge
- (5) This cross-mechanism interaction must have the result of enabling the knowledge creation mechanisms to perform much more effectively in combination than they would if operated non-interactively. This is “cognitive synergy.”

Interactions as mentioned in Points 4 and 5 in the above list are the real conceptual meat of the cognitive synergy idea. One way to express the key idea here is that most AI algorithms suffer from combinatorial explosions: the number of possible elements to be combined in a synthesis or analysis is just too great, and the algorithms are unable to filter through all the possibilities, given the lack of intrinsic constraint that comes along with a “general intelligence” context (as opposed to a narrow-AI problem like chess-playing, where the context is constrained and hence restricts the scope of possible combinations that needs to be considered). In an AGI architecture based on cognitive synergy, the different learning mechanisms must be designed specifically to interact in such a way as to palliate each others’ combinatorial explosions – so that, for instance, each learning mechanism dealing with a certain sort of knowledge, must synergize with learning mechanisms dealing with the other sorts of knowledge, in a way that decreases the severity of combinatorial explosion.

One prerequisite for cognitive synergy to work is that each learning mechanism must recognize when it is “stuck,” meaning it’s in a situation where it has inadequate information to make a confident judgment about what steps to take next. Then, when it does recognize that it’s stuck, it may request help from other, complementary cognitive mechanisms.

The key point we wish to make here regarding cognitive synergy is that this same principle, previously articulated mainly in the context of deliberative processes acting on long-term memory, seems intuitively to hold on the level of the integrative diagram. Most likely, cognitive synergy holds not only between the learning algorithms associated with different memory systems, but also between the dynamical processes associated with different large-scale components, such as are depicted in the different sub-diagrams of the integrative diagram depicted above. If this is so, then all the subdiagrams depend inti-

mately on each other in a dynamic sense, meaning that the processes within each of them must be attuned to the processes within each of the others, in order for the whole system to operate effectively. We do not have a proof of this hypothesis at present, so we present it as our intuitive judgment based on informal integration of a wide variety of evidence from cognitive science, neuroscience and artificial intelligence.

Of course, some pieces of the integrative diagram are bound to be more critical than others. Removing the language box might result in an AGI system with the level of intelligence of a great ape rather than a human, whereas removing significant portion of the perception box might have direr consequences. Removing episodic memory might yield behavior similar to certain humans with brain lesions, whereas removing procedural memory would more likely yield an agent with a basic inability to act in the world. But the point of cognitive synergy is not just that all the boxes in the integrative diagram are needed for human-level intelligence, but rather that the dynamics inside all the boxes need to interact closely in order to achieve human-level intelligence. For instance, it's not just removing the perception box that would harm the system's intelligence – forcing the perception box to operate dynamically in isolation from the processes inside the other boxes would have a similar effect.

8.6 Why Is It So Hard to Measure Partial Progress Toward Human-Level AGI?

Why it is so hard to measure partial progress toward human-level AGI? The reason is not so hard to understand, if one thinking in terms of cognitive synergy and system integration.

Supposing the integrative diagram is accurate – then why can't we get, say, 75% of the way to human level intelligence by implementing 75% of the boxes in the integrative diagram? The reason this doesn't work, we suggest, is that cognitive synergy possesses a frustrating but important property called “trickiness.”

Trickiness has implications specifically for the evaluation of *partial* progress toward human-level AGI. It's not entirely straightforward to create tests to measure the *final achievement* of human-level AGI, but there are some fairly obvious candidates for evaluation methods. There's the Turing Test (fooling judges into believing you're human, in a text chat) the video Turing Test, the Robot College Student test (passing university, via being judged exactly the same way a human student would), etc. There's certainly no agree-

ment on which is the most meaningful such goal to strive for, but there's broad agreement that a number of goals of this nature basically make sense.

On the other hand, it's much less clear how one should measure whether one is, say, 50 percent of the way to human-level AGI? Or, say, 75 or 25 percent?

It's possible to pose many "practical tests" of incremental progress toward human-level AGI, with the property that IF a proto-AGI system passes the test using a certain sort of architecture and/or dynamics, then this implies a certain amount of progress toward human-level AGI *based on particular theoretical assumptions about AGI*. However, in each case of such a practical test, it seems intuitively likely *to a significant percentage of AGI researchers* that there is some way to "game" the test via designing a system specifically oriented toward passing that test, and which doesn't constitute dramatic progress toward AGI.

Some examples of practical tests of this nature would be

- The Wozniak "coffee test": go into an average American house and figure out how to make coffee, including identifying the coffee machine, figuring out what the buttons do, finding the coffee in the cabinet, etc.
- Story understanding – reading a story, or watching it on video, and then answering questions about what happened (including questions at various levels of abstraction)
- Graduating (virtual-world or robotic) preschool
- Passing the elementary school reading curriculum (which involves reading and answering questions about some picture books as well as purely textual ones)
- Learning to play an arbitrary video game based on experience only, or based on experience plus reading instructions

One interesting point about tests like this is that each of them seems to *some* AGI researchers to encapsulate the crux of the AGI problem, and be unsolvable by any system not far along the path to human-level AGI – yet seems to other AGI researchers, with different conceptual perspectives, to be something probably game-able by narrow-AI methods. And of course, given the current state of science, there's no way to tell which of these practical tests really can be solved via a narrow-AI approach, except by having a lot of people try really hard over a long period of time.

A question raised by these observations is whether there is some *fundamental reason* why it's hard to make an objective, theory-independent measure of intermediate progress toward advanced AGI. Is it just that we haven't been smart enough to figure out the right test – or is there some conceptual reason why the very notion of such a test is problematic?

We suggest that a partial answer is provided by the “trickiness” of cognitive synergy. Recall that, in its simplest form, the cognitive synergy hypothesis states that human-level AGI intrinsically depends on the synergetic interaction of multiple components. In this hypothesis, for instance, it might be that there are 10 critical components required for a human-level AGI system. Having all 10 of them in place results in human-level AGI, but having only 8 of them in place results in having a dramatically impaired system – and maybe having only 6 or 7 of them in place results in a system that can hardly do anything at all.

Of course, the reality is almost surely not as strict as the simplified example in the above paragraph suggests. No AGI theorist has really posited a list of 10 crisply-defined subsystems and claimed them necessary and sufficient for AGI. We suspect there are many different routes to AGI, involving integration of different sorts of subsystems. However, if the cognitive synergy hypothesis is correct, then human-level AGI behaves *roughly* like the simplistic example in the prior paragraph suggests. Perhaps instead of using the 10 components, you could achieve human-level AGI with 7 components, but having only 5 of these 7 would yield drastically impaired functionality – etc. Or the point could be made without any decomposition into a finite set of components, using continuous probability distributions. To mathematically formalize the cognitive synergy hypothesis in its fully generality would become quite complex, but here we’re only aiming for a qualitative argument. So for illustrative purposes, we’ll stick with the “10 components” example, just for communicative simplicity.

Next, let’s suppose that for any given task, there are ways to achieve this task using a system that is much simpler than any subset of size 6 drawn from the set of 10 components needed for human-level AGI, but works much better for the task than this subset of 6 components (assuming the latter are used as a set of only 6 components, without the other 4 components).

Note that this supposition is a good bit stronger than mere cognitive synergy. For lack of a better name, we’ll call it *tricky cognitive synergy*. The tricky cognitive synergy hypothesis would be true if, for example, the following possibilities were true:

- creating components to serve as parts of a synergetic AGI is *harder* than creating components intended to serve as parts of simpler AI systems without synergetic dynamics
- components capable of serving as parts of a synergetic AGI are necessarily *more complicated* than components intended to serve as parts of simpler AGI systems.

These certainly seem reasonable possibilities, since to serve as a component of a synergetic AGI system, a component must have the internal flexibility to usefully handle interactions with a lot of other components as well as to solve the problems that come its way. In a CogPrime context, these possibilities ring true, in the sense that tailoring an AI process for tight integration with other AI processes within CogPrime, tends to require more work than preparing a conceptually similar AI process for use on its own or in a more task-specific narrow AI system.

It seems fairly obvious that, if tricky cognitive synergy really holds up as a property of human-level general intelligence, the difficulty of formulating tests for intermediate progress toward human-level AGI follows as a consequence. Because, according to the tricky cognitive synergy hypothesis, any test is going to be more easily solved by some simpler narrow AI process than by a *partially complete* human-level AGI system.

8.7 Conclusion

We have presented an integrative diagram summarizing and merging multiple researchers' views regarding the architecture of human-level general intelligence. We believe the results of our work demonstrate a strong degree of overlap and synergy between different contemporary perspectives on AGI, and illustrate that a substantial plurality of the AGI field is moving toward consensus on the basic architecture of human-like general intelligence. Also, we suggest the integrative diagram may be useful from a purely cognitive science view, as a coherent high-level picture of all the parts of the human mind and how they work together.

We have also presented an argument that, to achieve anything remotely similar to human-level general intelligence, it will be necessary to implement *all* of the integrative diagram, not just isolated bits and pieces. We believe the arguments given here regarding trickiness provide a plausible explanation for the empirical observation that positing tests for intermediate progress toward human-level AGI is a very difficult prospect. If the theoretical notions sketched here are correct, then this difficulty is not due to incompetence or lack of imagination on the part of the AGI community, nor due to the primitive state of the AGI field, but is rather intrinsic to the subject matter. And in that case, the practical implication for AGI development is, very simply, that one shouldn't worry a lot about producing intermediary results that are compelling to skeptical observers. Just at 2/3 of a human brain may not be much use, similarly, 2/3 of an AGI system may not be much use. Lack of

impressive intermediary results may not imply one is on a wrong development path; and comparison with narrow AI systems on specific tasks may be badly misleading as a gauge of incremental progress toward human-level AGI.

Thus our overall conclusion is both optimistic and pessimistic. If one implements a system instantiating the integrative diagram, and fills in each box with processes that cooperate synergetically with the processes in the other boxes to minimize combinatorial explosion – then one will get a human-level general intelligence. On the other hand, due to the trickiness of cognitive synergy, such a system may not display dramatic general intelligence until it is just about finished!

Bibliography

- [1] J. S. Albus and A. M. Meystel. *Engineering of Mind: An Introduction to the Science of Intelligent Systems*. Wiley and Sons, 2001.
- [2] I. Arel, D. Rose, and R. Coop. Destin: A scalable deep learning architecture with application to high-dimensional robust pattern recognition. *Proc. AAAI Workshop on Biologically Inspired Cognitive Architectures*, 2009.
- [3] Bernard Baars and Stan Franklin. Consciousness is computational: The lida model of global workspace theory. *International Journal of Machine Consciousness*, 2009.
- [4] Joscha Bach. *Principles of Synthetic Intelligence*. Oxford University Press, 2009.
- [5] Dietrich Dörner. *Die Mechanik des Seelenwagens. Eine neuronale Theorie der Handlungsregulation*. Verlag Hans Huber, 2002. ISBN 345683814X.
- [6] Stan Franklin and Bernard Baars. Possible neural correlates of cognitive processes and modules from the lida model of cognition. *Cognitive Computing Research Group, University of Memphis*, 2008. <http://ccrg.cs.memphis.edu/tutorial/correlates.html>.
- [7] Ben Goertzel. Opencog prime: A cognitive synergy based architecture for embodied artificial general intelligence. In *ICCI 2009, Hong Kong*, 2009a.
- [8] Ben Goertzel. Cognitive synergy: A universal principle of feasible general intelligence? 2009b.
- [9] Ben Goertzel *et al.* A general intelligence oriented architecture for embodied natural language processing. In *Proc. of the Third Conf. on Artificial General Intelligence (AGI-10)*. Atlantis Press, 2010.
- [10] Jeff Hawkins and Sandra Blakeslee. *On Intelligence*. Brown Walker, 2006.
- [11] Daniel Jurafsky and James Martin. *Speech and Language Processing*. Pearson Prentice Hall, 2009.
- [12] Guang Li, Zhengguo Lou, Le Wang, Xu Li, and Walter J. Freeman. Application of chaotic neural model based on olfactory system on pattern recognition. *ICNC*, 1:378–381, 2005.
- [13] Marvin Minsky. *The Emotion Machine*. 2007.
- [14] Aaron Sloman. Varieties of affect and the cogaff architecture schema. In *Proceedings of the Symposium on Emotion, Cognition, and Affective Computing*, AISB-01, 2001.

Chapter 9

A New Constructivist AI: From Manual Methods to Self-Constructive Systems

Kristinn R. Thórisson

*Center for Analysis & Design of Intelligent Agents, Reykjavik University
and*

*Icelandic Institute for Intelligent Machines
Menntavegur 1, IS-101 Reykjavik, Iceland*

thorisson@{ru.is, iim.is}

The development of artificial intelligence (AI) systems has to date been largely one of manual labor. This constructionist approach to AI has resulted in systems with limited-domain application and severe performance brittleness. No AI architecture to date incorporates, in a single system, the many features that make natural intelligence general-purpose, including system-wide attention, analogy-making, system-wide learning, and various other complex transversal functions. Going beyond current AI systems will require significantly more complex system architecture than attempted to date. The heavy reliance on direct human specification and intervention in constructionist AI brings severe theoretical and practical limitations to any system built that way.

One way to address the challenge of artificial general intelligence (AGI) is replacing a top-down architectural design approach with methods that allow the system to manage its own growth. This calls for a fundamental shift from hand-crafting to self-organizing architectures and self-generated code – what we call a *constructivist AI* approach, in reference to the self-constructive principles on which it must be based. Methodologies employed for constructivist AI will be very different from today’s software development methods; instead of relying on direct design of mental functions and their implementation in a cognitive architecture, they must address the *principles* – the “seeds” – from which a cognitive architecture can automatically grow. In this paper I describe the argument in detail and examine some of the implications of this impending paradigm shift.

9.1 Introduction

Artificial intelligence researchers have traditionally been rather optimistic about the rate of progress in the field. The origin of this optimism dates back numerous decades, possibly

as far back as our understanding of the electrical properties of neurons and their role in controlling animal behavior, but at the very least to the invention of using electricity for automatic calculation and related tasks – tasks that only humans used to be capable of. These realizations seemed so laden with potential that even the most ardent skeptics couldn't help imagining near-future scenarios where electrical machines would be doing all sorts of tasks requiring intelligence, helping out in every area of human endeavor.

In spite of measurable progress since the early discoveries, one big question still remains unanswered: How does the human and animal mind work? Bits and pieces of answers have been popping out of neuroscience, cognitive science, psychology, and AI research, but a holistic picture seems as far in the future as ever. What we would like to see – and this is a vision shared by many of those who started the field of AI over 50 years ago – is an artificial system that can learn numerous disparate tasks and facts, reliably perform them in a variety of circumstances and environments of real-world complexity; a system that can apply itself to learning a wide range of skills in a wide range of domains. Such a system would be considered by most as “generally intelligent”. A trivial example is an AI that can learn, over a period of say 3 months, to cook dinner, do the dishes, and fix automobiles. An AI that can acquire the skills to invent new things, solve global warming, and negotiate peace treaties would be yet another step forward, but for all we know, this might not be too far-fetched if we can achieve the former.

A functioning brain is a reasonably good place to start studying how thought works – after all natural intelligence is what gave us the idea to build artificial minds in the first place. Consider functions of natural minds such as global attention with introspective capabilities, the ability to discover, understand and abstract facts and causal chains, to make analogies and inferences, and to learn a large amount of vastly different skills, facts and tasks, including the control of one's own thoughts. These are features that seem simply too critical to leave out when attempting to build an intelligent system. It is in part the historical neglect of such key features – but especially *their integration* – that motivates the present discussion.

The vast collection of atoms and molecules found in everything we see in this universe tells us little about familiar phenomena such as oceans, rainforests and Picasso paintings. In the same way, brain neurons are but one of the many building blocks behind the complex phenomenon we normally recognize as intelligence. Just like the atoms in the trunk of a tree or water molecules on the surface of the Earth, neurons participate in patterns of interaction that form complex structures of lower granularity, all of which are more complex than any

single neuron alone. Without the right superstructures, brain neurons are nothing more than fancy amoebas, and just as far from what we recognize as high-level intelligence.

Studying only neurons for finding out how the mind works is akin to restricting oneself to atoms when studying weather patterns. Similarly, languages used today for programming computers are too restrictive – instead of helping us build complex animated systems they focus our attention on grains of sand, which are then used to implement bricks – software modules – which, no surprise, turn out to be great for building brick houses, but are too inflexible for creating the mobile autonomous robot we were hoping for. In short, modern software techniques are too inflexible for helping us realize the kinds of complex dynamic systems necessary to support general intelligence.

What is called for are new methodologies that can bring more power to AI development teams, enabling them to study cognition in a much more holistic way than possible today, and focus on *architecture* – the operation of the system as a whole. To see why this is so we need to look more closely at what it is that makes natural intelligence special.

9.2 The Nature of (General) Intelligence

Whether or not natural intelligence is at the center of our quest for thinking machines, it certainly gives us a benchmark; even simple animals are capable of an array of skills way beyond the most advanced man-made machines. Just to take an example, a fully grown human brain must contain millions of task-specific abilities. This fact in itself is impressive, but pales in comparison to the fact that these skills have been acquired largely autonomously by the system itself, through self-directed growth, development, and training, and are managed, selected, improved, updated, and replaced dynamically, while the system is in use. An integrated system with these capabilities is able to apply acquired skills in realtime in varied circumstances; it can determine – on the fly – when and how to combine them to achieve its goals. And in spite of already having a vast amount of acquired skills, it must retain the ability to acquire new ones, some of which may have very little overlap with any prior existing knowledge.

The field of artificial intelligence has so far produced numerous partial solutions to what we normally call intelligent behavior. These address isolated sub-topics such as playing board games, using data from a camera to perform a small set of predetermined operations, transcribing spoken words into written ones, and learning a limited set of actions from experience. In systems that learn, the learning targets (goals) are hand-picked by the

programmer; in systems that can to some extent “see” their environment, the variations in operating contexts and lighting must be highly restricted; in systems that can play board games, the types of games are limited in numerous ways – in short, there are significant limitations to the systems in each of these areas. No system has yet been built that can learn two or more of these domains by itself, given only the top-level goal of simply learning “any task that comes your way”. These systems are nonetheless labeled as “artificially intelligent” by a majority of the research community.

In contrast, what we mean by “general” intelligence is the ability of a system to learn many skills (e.g. games, reading, singing, playing racketball, building houses, etc.) and to learn to perform these in many different circumstances and environments. To some extent one could say that the ability to “learn to learn” may be an important characteristic of such a system. Wang (2004) puts this in the following way:

“If the existing domain-specific AI techniques are seen as *tools*, each of which is designed to solve a special problem, then to get a general-purpose intelligent system, it is not enough to put these tools into a toolbox. What we need here is a *hand*. To build an integrated system that is self-consistent, it is crucial to build the system around a general and flexible *core*, as the hand that uses the tools [assuming] different forms and shapes.”

Many abilities of an average human mind, that are still largely missing from present AI systems, may be critical for higher-level intelligence, including: The ability to learn through experience via examples or through instruction; to separate important things from unimportant ones in light of the system’s goals and generalize this to a diverse set of situations, goals and tasks; steering attention to important issues, objects and events; the ability to quickly judge how much time to spend on the various subtasks of a task in a complex sequence of actions; the ability to continuously improve attention steering; to make analogies between anything and everything; the ability to learn a wide range of related or unrelated skills without damaging what was learned before; and the ability to use introspection (thinking about one’s own thinking) as a way to understand and improve one’s own behavior and mental processes, and ultimately one’s performance and existence in the world.

These are but a few examples of many *pan-architectural* abilities that involve large parts of the entire system, including the process of their development, training, and situated application. All of these abilities are desirable for a generally intelligent artificial mind,

and many of them may be necessary. For example, a system working on the docks and helping to tie boats to the pier must be able to ignore rain and glaring sun, the shouts of others doing their own tasks, detect the edge of the pier, catch a glimpse of the rope in the air, prepare its manipulators to catch it, catch it, and tie it to the pier – all within the span of a few seconds. In a real-world scenario this task has so many parameters that the system's realtime attentional capabilities must be quite powerful. It is hard to see how a general-purpose intelligence can be implemented without an attention mechanism that can – in realtime – learn to shift focus of attention effectively from internal events (e.g. remembering the name of a colleague) to external events (e.g. apologizing to those present for not remembering her name), and at runtime – while operating – improve this skill based on the goals presented by social norms. Natural intelligences probably include many such attention mechanisms, at different levels of detail. Attention is just one of the seemingly complex transversal skills – pan-architectural skills that represent in some way fundamental operational characteristics of the system – without which it seems rather unlikely that we will ever see artificial general intelligence, whether in the lab or on the street.

The enormous gap in size and complexity between a single neuron and a functioning animal brain harbors a host of challenging questions; the size and nature of this challenge is completely unclear. What is clear, however, is that a fully formed brain is made up of a network of neurons forming substructure upon substructure of causal information connections, which in turn form architectures within architectures within architectures, nested at numerous levels of granularity, each having a complex relationship with the others both within and between layers of granularity [10, 34]. This relationship, in the form of data connections, determines how the mind deals with information captured by sensory organs, how it is manipulated, responded to, and learned from over time. The architectures implement highly coordinated storage systems, comparison mechanisms, reasoning systems, and control systems. For example, our sense of balance informs us how to apply forces to our arm when we reach for a glass of water while standing on a rocking boat. Past experience in similar situations tells us to move slowly. Our sense of where the glass is positioned in space informs our movement; the information from our eyes and inner ear combines to form a plan, and produce control signals for the muscles, instructing them how to get the hand moving in the direction of the glass without us falling down or tipping the glass over. A system that can do this in real-time is impressive; a system that can *learn* to do this, for a large set of variations thereof, along with a host of other tasks, must also have a highly flexible architecture.

Elsewhere I have discussed how the computational architecture, software implementation, and the cognitive architecture that it implements, need not be isomorphic [35]. Thus, the arguments made here do not rest on, and do not *need* to rest on, assumptions about the particular *kind of architecture* in the human and animal mind; in the drive for a new methodology we are primarily concerned with the operational characteristics that the system we aim to build – the architecture – must have. Past discussions about cognitive architecture, whether the mind is primarily “symbol-based” (cf. [20]), “dynamical” (cf. [9, 41]), “massively modular” (cf. [2]), or something else entirely, can thus be put aside. Instead we focus on the sheer *size* of the system and the extreme *architectural plasticity*, while exhibiting a tight integration calling for a *high degree of interdependencies between an enormous set of functions*. Just as an ecosystem cannot be understood by studying one lake and three inhabiting animal species, intelligence cannot be realized in machines by modeling only a few of its necessary features; by neglecting critical interdependencies of its relatively large number of heterogeneous mechanisms most dissections are prevented from providing more than a small fragment of the big picture. General intelligence is thus a system that implements numerous complex functions organized at multiple levels of organization. This is the kind of system we want to build, and it cannot be achieved with the present methodologies, as will become evident in the next section.

To summarize, work towards artificial *general* intelligence (AGI) cannot ignore necessary features of such systems, including:

- *Tight integration*: A general-purpose system must tightly and finely coordinate a host of skills, including their acquisition, transitions between skills at runtime, how to combine two or more skills, and transfer of learning between them over time at many levels of temporal and topical detail.
- *Transversal functions*: Related to the last point, but a separate issue; lies at the heart of system flexibility and generality: The system must have pan-architectural characteristics that enable it to operate consistently as a whole, to be highly adaptive (yet robust) in its own operation across the board, including meta-cognitive abilities. Some such functions have been listed already, namely attention, learning, analogy-making capabilities, and self-inspection, to name some.
- *Time*: Ignoring (general) temporal constraints is not an option if we want AGI. Time is a semantic property, and the system must be able to understand – and be able to *learn* to understand – time as a real-world phenomenon in relation to its own skills and architectural operation.

- *Large architecture*: An architecture that is considerably larger and more complex than systems being built in AI labs today is likely unavoidable, unless we are targeting toy systems. In a complex architecture the issue of concurrency of processes must be addressed, a problem that has not yet been sufficiently resolved in present software and hardware. This scaling problem cannot be addressed by the usual “we’ll wait for Moore’s law to catch up” [19] because the issue does not primarily revolve around speed of execution, but around the nature of the architectural principles of the system and their runtime operation.

This list could be extended even further with various other desirable features found in natural intelligences such as predictable robustness and graceful degradation; the important point to understand here is that if some (or all) of these features *must* exist in the same system, then it is highly unlikely that we can create a system that addresses them unless we address them *at the same time*, for (at least) one obvious reason: For any partially operating complex architecture, retrofitting one or more of these into it would be a practical – and possibly a theoretical – impossibility (cf. [11]). As I myself and others have argued before [43], the conclusion can only be that *the fundamental principles of an AGI must be addressed holistically*. This is clearly a reason to think long and hard about our methodological approach to the work at hand.

9.3 Constructionist AI: A Critical Look

In computer science, *architecture* refers to the layout of a large software system made up of many interacting parts, often organized as structures within structures, and an operation where the “whole is greater than the sum of the parts”. Although not perfect, the metaphorical reference to physical structures and urban layout is not too far off; in both cases system designs are the result of compromises that the designers had to make based on a large and often conflicting set of constraints, which they resolved through their own insight and ingenuity. In both cases *traffic* and *sequence of events* is steered by *how things are connected*; in the case of software architecture, it is the traffic of *information* – who does what when and who sends and receives what information when.¹

There is another parallel between urban planning and software development that is even more important; it has to do with the tools and methodologies used to design and imple-

¹I use the term “software architecture” here somewhat more inclusively than the metaphorical sense might imply, to cover both the parts responsible for system operation, processing, and data manipulation, as well as the data items and data structures that they operate on.

ment the target subject. When using computer simulations as an integral part of a research methodology, as is done increasingly in many fields including astrophysics, cognitive science, and biology, an important determinant of the speed of progress is how such software models are developed and implemented; the methodology used to write the software and the surrounding software framework in which they are developed and run, is in fact *a key determinant of progress*. Could present methodologies in computer science be used to address the challenges that artificial *general* intelligence (AGI) presents?

Since software systems are developed by human coders, the programming languages now used have been designed for human use and readability, and it is no surprise that they reflect prototypical ways in which humans understand the world. For instance, the most popular method for organizing programming languages in recent years is based on “object orientation,” in which data structures and processes are organized into groups intended to give the human designer a better overview of the system being developed. Dependencies between the operation and information flow in such groups, that is, what particulars of what groups of processes are allowed to receive, process, and output the various types of data, is decided at design time. This also holds for the groups’ operational semantics, where each group is essentially a special-purpose processor that has special-purpose behavior, with a pre-defined role in the system as a whole, defined by the software coder at design time. In this constructionist approach variables, commands, and data, play the role of bricks; the software developer takes the role of the “construction worker.”

The field of artificial intelligence too relies on a constructionist approach: Systems are built by hand, and both the gross and fine levels of architectures are laid down “brick by brick.” This is where we find the most important similarity between architecture and software development, and the one with which we are primarily concerned here; in both cases *everything* – from the fine points to the overall layout of the final large-scale processing structures – is *defined, decided* and *placed in the system* by *human designers*.

As history unequivocally shows, all AI systems developed to date with constructionist methodologies have had the following drawbacks, when compared to natural intelligence:

- They are extremely *limited* in what they can do, and certainly don’t come anywhere *close* to addressing the issues discussed in the prior section, especially transversal functions such as global attention mechanisms and system-wide learning.
- They are *brittle*, stripped of the ability to operate outside of the limited scope targeted by their designer. More often than not they also tend to be brittle when operating *within* their target domain, especially when operating for prolonged periods, which

reveals their sensitivity to small errors in their implementation and lack of graceful degradation in respect to partial failure of some components.

To be sure, AI researchers often extend and augment typical software development methodologies in various ways, going beyond what can be done with “standard” approaches. The question then becomes, how far could constructionist methodology be taken?

Over the last few decades only a handful of methodologies have been proposed for building large, integrated AI systems – Behavior-Oriented Design (BOD; [5]), the Subsumption Architecture [4], Belief, Desires, Intentions (BDI; cf. [28]), and the Constructionist Design Methodology (CDM) [37] are among them. CDM rests on solid present-day principles of software engineering, but is specifically designed to help developers manage system expansion; its principles allow continuous architecture-preserving expansion of complex systems involving large numbers of executable modules, computers, developers, and research teams.²

The Cognitive Map architecture [22], implemented on the Honda ASIMO robot, enables it to play card games with children using reciprocal speech and gesture. Implementing and coordinating state of the art vision and speech recognition methods, novel spatio-temporal interpretation mechanisms and human-robot interaction, this system is fairly large, and it is among the relatively few that make it a specific goal to push the envelope on scale and breadth of integration.

The Cognitive Map architecture has benefited greatly from the application of the CDM to its construction, as system development has involved many developers over several years. The architecture is implemented as multiple semi-independent modules running on multiple CPUs interacting over a network. Each module is responsible for various parts of the robot’s operations, including numerous perceptors for detecting and indexing perceived phenomena and various types of deciders, spatial, and semantic memory systems, action controllers, etc. Each of the modules is a (hand-crafted) piece of software, counting anywhere from a few dozen lines of code to tens of thousands. As most of the modules are at the smaller end of this spectrum, interaction and network traffic during system runtime is substantial. The components in the Cognitive Map architecture are coordinated via an infrastructure called Psyclone AIOS, a middleware that targets large AI systems based on multiple dynamically interacting modules, with powerful blackboard-based data sharing

²Results of the application of CDM have been collected for several types of systems, in many contexts, at three different research institutions, CADIA [15, 29, 38], Honda Research Labs (HRI-US) [21, 22] and the Computer Graphics and User Interfaces Lab at Columbia University [37].

mechanisms [39]. Psyclone AIOS allows multiple programming languages to be used together, supports easy distribution of processes over a network, handles data streams that must be routed to various destinations at runtime (e.g. from video cameras), and offers various other features that help with AI architecture development. The system supports directly the principles of the CDM and is certainly among the most flexible such systems involving the creation and management of large architectures.³

In light of progress within the field of robotics and AI, the ASIMO Cognitive Map architecture is a strong contender: It integrates a large set of functionality in more complex ways than comparable systems did only a few years ago. It relies on a methodology specifically targeted to AI and robotic systems, where multiple diverse functions implemented via thousands of lines of code must be tightly integrated and coordinated in realtime. So how large is the advancement demonstrated by this system?

The resulting system has all of the same crippling flaws as the vast majority of such systems built before it: It is brittle, and complex to the degree that it is becoming exponentially more expensive to add features to it – we are already eyeing the limit. Worst of all, it has still not addressed – with the exception of a slight increase in breadth – a single one of the key features listed above as necessary for AGI systems. In spite of good intentions, neither the CDM – nor any of the other methodologies, for that matter – could help address the difficult questions of transversal functions and architectural construction which are orders of magnitude larger than attempted to date.

Some of the systems I have been involved with developing have contained an unusually large number of modules, with sizes varying from very small (a few dozen lines of code) to very large (tens of thousands). We refer to them as “granular” architectures (cf. [15]). In these systems the algorithms coordinating the modules (i.e. the gross architecture) dynamically control which components are active at what time, which ones receive input from where, etc. Some of the components can change dynamically, such as the Indexers/Deciders in the Cognitive Map [21], and learn, such as the module complex for learning turntaking in the artificial radio-show host [15]. Increased granularity certainly helps making the architecture more powerful. But yet again, the modules in these systems are typically black-box with prescribed dependencies, which precludes them from automatically changing their operation, expand their reach, or even modify their input and output profiles in any significant way, beyond what the coder could prescribe. Their inputs and outputs are directly dependent on the surrounding architecture, which is restricted by the

³Other solutions addressing similar needs include Elvin [33], the Open Agent Architecture [18] and NetP [13].

inability of components to change their operation. Many component technologies used in these architectures are built from different theoretical assumptions about their operating context, increasing this dependency problem.

On the practical side, many problems get worse as the architectures get bigger, e.g. for lack of fault tolerance (such as code-level bugs). Some new problems are introduced, especially architecture-level problems and those we call *interaction problems*: Loosely-coupled modules often have complex (and infrequent) patterns of interaction that are difficult to understand for the developers in the runtime system; interaction problems grow exponentially as the system gets bigger. A constructionist approach does not help us unlock tight interdependencies between components, or remove the need for humans to oversee and directly interact with the system at the code level.

Examples of other AI architectures with ambitions towards artificial general intelligence include LIDA [8], AKIRA [25], NARS [45], SOAR [16], CLARION [32], ACT-R [1], OSCAR [27], and Ikon Flux [23]. However, those that have been tested on non-trivial tasks, such as ACT-R, SOAR, and CLARION, are based on constructionist methodologies with clear limitations in scaling arising thereof; those that promise to go beyond current practices, e.g. Ikon Flux, NARS, LIDA, and OSCAR, suffer from having either been applied only to toy problems, or are so new that thorough evaluation is still pending.

No matter how dynamic and granular the components of an architecture are made, or which expanded version of a constructionist methodology is being applied, a heavy reliance on manual construction has the following effects:

- System components that are fairly static. Manual construction limits the complexity that can be built into each component.
- The sheer number of components that can form a single architecture is limited by what a designer or team can handle.
- The components and their interconnections in the architecture are managed by algorithms that are hand-crafted themselves, and thus also of limited flexibility.

Together these three problems remove hopes of autonomous architectural adaptation and system growth. Without system-wide adaptation, the systems cannot break free of targeted learning. Like most if not all other engineered systems of a comparable scale and level of integration, e.g. telephone networks, CPUs, and power grids, these systems are incapable of architecture-level evolution, precluding architecture-wide learning (what one might metaphorically think of as “cognitive growth”) and deep automatic adaptation, all of which precludes general-purpose systems capable of applying themselves autonomously to

arbitrary problems and environments. That is precisely the kind of flexibility we want a new methodology to enable us to imbue an AI architecture with.

The solution – and most fruitful way for AI in the coming decades – rests on the development of a new style of programming, with greater attention given to the architectural makeup, structure, and nature of large complex systems, bringing with it the element of automated systems management, resting on the principles of transparent operational semantics.

9.4 The Call for a New Methodology

Available evidence strongly indicates that *the power of general intelligence*, arising from a *high degree of architectural plasticity*, is of a *complexity well beyond the maximum reach of traditional software methodologies*. At least three shortcomings of constructionist AI need to be addressed in a new methodology: Scale, integration, and flexibility. These are fundamental shortcomings of *all software systems developed to date*, yet, as we have seen above, they *must* all be overcome *at the same time* in the same system, if we wish to achieve artificial general intelligence. Any approach that is successful in addressing these must therefore represent *a fundamentally new type of methodology*.

Scale matters in at least two ways. First, we have reason to believe that a fairly large set of cognitive functions is necessary for even modestly complex real-world environments. A small system, one that was of a size that could be implemented by a handful of engineers in half a decade, is not likely to support the reliable running of supernumerary functions. And historical evidence certainly does not help refute this claim: In spite of decades of dealing with the various problems of scaling beyond toy contexts (“context” interpreted in a rather general form, as in “the ocean” versus “the desert”; “indoors” versus “outdoors,” etc.), standard component-based software methodology has theoretical limitations in the size of systems that it can allow to be built; as these systems are programmed by humans, their size and complexity is in fact restricted by the industriousness of a dedicated team of researchers in the same way that building a house is. This is the reason why we still have not seen systems that scale easily. What is needed is the equivalent of a highly automated factory. Second, a small system is not very likely to lend sufficient support to the kind of functions that characterize higher-level intelligences, such as system-wide analogy-making, abstraction, cross-domain knowledge and knowledge transfer, dynamic

and learnable attention, all of which require transversal functionality of some sort to be of general use.⁴

The issue of integration ultimately revolves around software architecture. Most architectures built to date are coarse-grained, built of relatively large modules, because this is the kind of architecture that traditional methodologies most naturally support. The size of components in constructionist systems built to date varies from “a few” to “dozens”, depending on which system you look at. To take some concrete examples, in our own single-mind (as opposed to multi-agent) systems we have had close to 100 modules, most of which are only a few pages of C++ code each, but often include two or three significantly larger ones (thousands of lines or more) in the full system (see e.g. [38]). Each of these may have somewhere from 0 to, at most, 20 parameters that can be changed or tuned at runtime. Such a system simply does not permit the highly dynamic communication and behavior patterns required for these sophisticated functionalities. More importantly, the architecture is too inflexible for sub-functions to be shared between modules at the gross-architecture level. In such an arrangement tight integration is precluded: For AGIs we need a tighter, deeper integration of cognitive functions; we need a methodology that allows us to design architectures composed of tens of thousands of components with ease – where the smallest component is peewee-size (the size of a medium-size C++ operation [40]) and where the combination of these into larger programs, and their management, is largely automatic.

We are looking for more than a linear increase in the power of our systems to operate reliably, and in a variety of (unforeseen) circumstances; experience strongly suggests that a linear increase in present methods will not bring this about: Nothing in traditional methods shows even a hint of how more flexible systems could be built using that approach. One of the biggest challenges in AI today is to move away from brittle, limited-domain systems. Hand-crafted software systems tend to break very easily, for example when taking inputs outside the scope of those anticipated by their designers or because of unexpected interaction effects amongst the systems’ components. What seems clear is that a new methodology must inevitably revolve around what could be thought of as “meta”-programs; programs that guide the creation of new programs that guide the system’s interaction patterns with the world, and possibly test these in simulation mode beforehand, as a “mental exercise”, to predict how well they might work. The systems need to have a reliable way to judge whether prior knowledge exists to solve the problem, and whether the problem or situation

⁴Although system-wide learning and self-modification could be realized in isolation by a small system, these features are likely to be impossible to maintain in a large system when taking a constructionist approach, and we need the system to incorporate all of these features in a unified manner.

falls within what the system has already encountered or whether it finds itself in completely new circumstances. So, we need a methodology for designing a system whose output, cognitive work, is a set of programs that are more or less new compared to what existed in the system before; programs that can be given a chance to guide the system in its interactions with new operating contexts (domains), and ways to assess the results of such “hypothesis testing”: The programs must be compared and evaluated on the grounds of the results they achieve. It may even be necessary for the evaluation itself to be learnable, for, after all, the system should be as self-organizing as possible. For a large system, doing all of this with present reinforcement learning and genetic algorithm techniques is likely to be too limited, too slow, or, most probably, impossible; the system must therefore be endowed with analogy making, reasoning, and inference capabilities to support such skills.

If we can create systems that can adapt their operating characteristics from one context to another, and propose what would have to be fairly new techniques with a better chance of enabling the system’s operation in the new environment, then we have created a system that can *change its own architecture*. And that is exactly what I believe we need: For any real-world domain (e.g. an indoor office environment) that is sufficiently different from another real-world domain (e.g. a busy street), creating a system that can not only operate but *learn* to operate in both, as well as in new domains, must be a system that can change its own operation in fundamentally new ways – these systems would have self-organizing architectures that largely manage their own growth.

9.5 Towards a New Constructivist AI

A system that can learn to change its own architecture in sensible ways is a *constructivist AI* system, as it is to some significant extent *self-constructive* [36]. The name and inspiration comes partly from Piaget [26], who argued that during their youth humans develop cognitive faculties via a self-directed “constructive” process, emphasizing the *active* role that a learning mind itself has in any learning process. Piaget’s ideas later became the foundation for the work by Drescher [7], who brought this idea to artificial intelligence, arguing that AI should study the way minds *grow*. The present work shares Drescher’s aim for more powerful ways of learning, aligning with the general hypothesis behind his work that an (artificial) mind requires sophisticated abilities to build representations and grow its knowledge about the world based on its own direct experiences. But in the present work we take this idea a step further by arguing for a fundamental change in *methodolog-*

ical assumptions, emphasizing the need for new principles of automatic management of *whole AI architectures* – i.e. the mind itself: It is not only the system’s learning but the control structures themselves that must be part of such cognitive development and constant (auto-)construction. The methodologies employed for such AI development – constructivist AI – are likely to be *qualitatively different* from today’s software development methods.

Here a drive for constructivist AI – that is, a system that can modify its own internal structures to improve its own operational characteristics, based on experience – arises thus from two fundamental assumptions, namely that (a) constructionist methodologies (i.e. by and large all traditional software development techniques) are not sufficient – both in principle and in practice – to realize systems with artificial general intelligence; (b) the hypothesis that automation of not just parameter tuning or control of (coarse-grained) module operation but of *architectural construction* (as understood in computer science) and management is what is needed to address this shortcoming. This is in line with the ideas presented by the proponents of second-order cybernetics and cybernetic epistemology [42] [12], which studies the nature of self-regulation.

In our approach we are by and large ruling out *all* methodologies that require some form of hand-coding of domain-level operational functionality (what Wang metaphorically referred to as “tools” [43]), as well as any and all approaches that require extensive hand-coding of the final static architecture for an artificial general intelligence (metaphorically referred to as “hand” by [43], limiting initial (manual) construction to a form of “seed” – a kind of meta-program – which automates the management of all levels below. This is what we call constructivist AI. Pure constructivist systems do not exist yet in practice, but some theoretical work already shows promise in this direction.

The following are topics that I consider likely to play a critical role in the impending paradigm shift towards constructivist AI. The topics are: *Temporal grounding*, *feedback loops*, *pan-architectural pattern matching*, *small white-box components*, and *architecture meta-programming and integration*. Other key factors probably exist that are not included here, so this list should be considered a necessary but insufficient set of topics to focus on in the coming decades as we turn our sights to building larger, more self-organizing systems.

9.5.1 *Temporal Grounding*

As now seems widely accepted in the AI community, it is fairly useless to talk of an entity being “intelligent” without referencing the context in which the entity operates. In-

telligence must be judged by its behavioral effects on the world in particular circumstances which are not part of the entity's operation: We cannot rightfully show an entity to be smart unless we consider both what the entity does and what kinds of challenges the environment presents. This means that intelligent behavior requires *grounding* – a meaningful “hook-up” between an intelligent system's thoughts and the world in which it operates. This grounding must include a connection to both space and time: Ignoring either would cripple the entity's possibility of acting intelligently in its world, whether real or virtual. So, for one, the entity must have a means to be *situated* in this world – it must have a body, a (limited) collection of sensors to accept raw data through and some (limited) set of actuators – to affect its surroundings. By the same token, its body must be connected to the entity's internal thought/computational processes to transfer the results of its thinking to the body.⁵

The issue goes beyond situatedness, which is a necessary but not sufficient condition for grounding: via situated perception and action, a feedback loop is created allowing the system to adapt to its environment and to produce models of the world that enable it to plan in that world, using predicted results of sequences of actions (plans). Results that do not match predictions become grounds for revision of its models of the world, and thus enable it to learn to exist in the given environment (cf. [44]). Therefore, to be grounded, an intelligent entity must be able to compute using processes that have a causal, predictable relationship with the external reality.

This leads us to a discussion of *time*. As any student of computer science knows, computation can be discussed, scrutinized and reasoned about without regard for how long it actually takes in a particular implemented system. It may be argued that this simplification has to some extent helped advance the fields of computer science and engineering. However, lack of a stronger foundation for the semantics of the actual, realtime execution of computational operations has hampered progress in fields dealing with highly time-critical topics, such as embedded systems, user interfaces, distributed networks, and artificial intelligence systems. As others have pointed out, timeliness is a semantic property [17] and must be treated as such. To be grounded, the computational operations of an intelligent entity must have a causal, *temporally contextualized and predictable* – and thus temporally meaningful – relationship with the external reality.

To make an intelligent machine that does not understand time is a strange undertaking. No example of natural intelligence exists where time isn't integral in its operation:

⁵Froese (2007) gives a good overview of past research on these topics.

When it comes to doing intelligent things in the world, time is of the essence. Indeed, in a world without the arrow of time there would be little need for the kind of intelligence we see in nature. The lack of a strong connection between computational operations and the temporal dimension is preventing a necessary theoretical and practical understanding of the construction of large architectural solutions that operate predictably under external time constraints. We need to find ways to build an operational knowledge of external time into AI architectures.

To use its own mental powers wisely an intelligent machine must not only be able to understand the march of the real-world clock itself, it should preferably understand its own capabilities and limitations with regards to time, lest it cannot properly make plans to guide its own learning or evolution. One method is to link the execution of the software tightly with the CPU's operation and inputs from the system's operating environment to create a clear semantic relationship between logical operations and the passing of realtime (running of the CPU), in a way that allows the system to do the inferencing and modeling of this relation itself and use this in its own operation throughout the abstraction layers in the entire architecture, producing a temporally grounded system. This is not mere conjecture; in the Ikon Flux system [23] lambda terms were used to implement this idea in a system containing hundreds of thousands of such terms, showing that this is indeed possible on large architectural scales, even on present hardware. And as mentioned above, the full perception-action loop needs to be included in these operational semantics.

There are thus at least three key aspects of temporal representation. The first is the perception of *external time*. The system exists in some world; this world has a clock; any system that cannot reasonably and accurately sense time at a resolution relevant to its operation will not be able to take actions with regards to events that march along to this clock, and thus – by definition – is not intelligent. Second, an intelligent system must have a representation of *mental time* and be able to estimate how long its own mental operations take. Third, an AI architecture must understand how the first two aspects relate, so that mental actions can be planned for based on *externally or internally-imposed timelines and deadlines*. The challenge is how to implement this in distributed, fine-grained architectures with parallel execution of subcomponents.

9.5.2 Feedback Loops

Any generally intelligent system operating in the real world, or a world of equivalent complexity, will have vastly greater information available than the mind of such a system

will have time to process. Thus, only a tiny fraction of the available data in the world is being processed by the system at any point in time. The actual data selected to be thought about must be selected based on the system's goals – of which there will be many, for any system of reasonable complexity. To be able to respond to unexpected events the system must further divide its processing capability between thinking about long-term things (those that have not happened yet and are either wanted, or to be avoided), and those that require immediate processing. Any mechanism that manages how the system chooses this division is generally called *attention*, or considered a part of an attentional mechanism. Attention is intricately linked with the perception-action loop, which deals with how the system monitors for changes in the world, and how quickly it is able to respond to these. For any intelligent system in a reasonably complex environment the nature and operation of these mechanisms are of utmost importance, as they are fundamental to the cognitive makeup of the system and put limits on all its other cognitive abilities.

Focus on the perception-action loop in current AI curricula is minimal. A quick look at some of the more popular textbooks on the subject reveals hardly any mention of the subject. Given that this most important loop in intelligent systems is largely ignored in the mainstream AI literature, it is no surprise that the little discussion there is dwells on their trivial aspects. Yet the only means for intelligent systems to achieve stability far from (thermodynamic) equilibrium is through feedback loops. The growth of a system, and its adaptation to genuinely new contexts, must rely on feedback loops to stabilise the system and protect it from collapsing. Furthermore, any expansion or modification of existing skills or capabilities, whether it is to support more complex inferencing, making skills more general-purpose or improving the performance on a particular task, requires an evaluation feedback loop. For general-purpose intelligence such loops need to permeate the intelligence architecture.

The way any entity achieves grounding is through feedback loops: repeated interactions which serve as experiments on the context in which the entity finds itself, and the abstractions which it has built of that context, of its own actions, and of its tasks. Feedback loops related to the complexities of tasks are key to a system's ability to learn particular tasks, and about the world. But the process must involve not only experience (feedback) of its actions on the context *outside* itself, it must also involve the context of its *internal processes*. So self-modeling is a necessary part of any intelligent being. The feedback loop between the system's thinking and its evaluation of its own cognitive actions is therefore just as important as that to the external world, because this determines the system's

ability to *learn to learn*. This, it can be argued, is a key aspect of general intelligence. Together these extremely important feedback loops provide a foundation for any increases in a system's intelligence.

Self-organization requires feedback loops, and constructionist AI methodologies make no contributions in this respect. The science of self-organization is a young discipline that has made relatively slow progress (cf. [30, 46]). As a result, concrete results are hard to come by (cf. [14]). Perhaps one of the important contributions that this field has to offer at present is to show how the principles behind self-organization call for a way of thinking that is very different from that employed in traditional software development methodologies.

9.5.3 *Pan-Architectural Pattern Matching*

Complex, tightly-integrated intelligence architectures will not work without large-scale pattern matching, that is, pattern matching that involves large portions of the system itself, both in knowledge of domain (e.g. tasks, contexts, objects, etc.) and of architecture (e.g. perception and action structures, hypothesis generation methods, etc.). Such pattern-matching functionality plays many roles; I will mention a few.

Any creature living in a complex world must be able to classify and remember the salient features of a large number of contexts.⁶ Without knowing *which* features to remember, as is bound to happen regularly as various new contexts are encountered, it must store *potential* features – a much larger set than the (ultimately) relevant features – and subsequently hone these as it experiences an increasingly larger numbers of contexts over time. In a complex environment like the real-world, the number of potential states or *task-relevant contexts* a being may find itself in is virtually infinite. Yet the being's processing power, unlike the number of contexts it may find itself in, is finite. So, as already mentioned, it must have some sort of attention.⁷ At any point in time the attentional mechanism selects memories, mental processes, memories of having applied/used a mental process for a particular purpose, or all of the above, to determine which mental process to apply in the present, identify potential for improvement, or simply for the purpose of reminiscing about the past. The pan-architectural nature of such mechanisms crystallizes in the rather large amounts of recall required for prior patterns involving not only features of objects to

⁶One way to think of contextual change, and hence the differentiators between one context and another, is as the smallest amount of change in a particular environmental configuration that renders a priorly successful behavior for achieving a particular goal in that configuration unable to achieve that goal after the change.

⁷Here "attention" refers to a broader set of actions than our typical introspective notion of attention, including the global control of parts of the mind that are active at any point in time, as well as what each one is doing.

be recognized, or the contexts in which these objects (including the creature itself) may be at any point, but also involving the way in which the being controls its attention in these contexts with regards to its task (something which it must also be able to learn), the various analogies it has made for the purpose of choosing a course of action, and the mechanisms that made these analogies possible. To do all this in realtime in one and the same system is obviously a challenge. This will, however, be impossible without the ability to compare key portions of the system's knowledge and control structures through large-scale pattern matching.

Yet another example of a process for which such transversal pattern matching is important is the growth of the system as it gets smarter with experience. To grow in a particular way, according to some specification,⁸ the architecture must have built-in ways to compare its own status between days, months, and years, and verify that this growth is according to the specification. This might involve pattern matching of large parts of the realtime mind, that is, the part of the mind that controls the creature from moment to moment at different points in time. For a large, heterogeneous architecture such architecture-scale pattern matching can get quite complicated. But it is unlikely that we will ever build highly intelligent artificial systems without it.

9.5.4 *Transparent Operational Semantics*

As already discussed, most integration in AI systems has involved relatively small numbers of the functions found in natural minds. A close look at these components – whether they are computer vision, speech recognition, navigation capabilities, planning, or other such specialized mechanisms – reveals internals with an intricate structure based on programming languages with syntax targeted for human programmers, involving mixtures of commercial and home-brewed algorithms. The syntax and semantics of “black-box” internals is difficult or impossible to discover from the outside, by observing only their inputs, outputs, and behaviors. This is a critical issue in self-organizing systems: The more opaque complex mechanisms encompassed by any single component in an architecture are, the harder it is to understand its operational semantics. In other words, the greater the complexity of atomic components, the greater the intelligence required to understand them. For this reason, to make architectures that construct themselves, we must move away from large black-box components.

⁸Such a specification could be small or medium-sized, compared to the resulting system, and it could be evolved, as our DNA has been, or provided via new meta-programming methods.

But the grand goal of self-construction cuts even deeper than dictating the size of our building blocks: It calls for them to have a somewhat different nature. Without exception, present programming languages used in AI are designed for human interpretation. By requiring human-level intelligence to be understood, these programming languages have little chance of being interpreted by other software programs *automatically*; their operational semantics are well above a semantic threshold of complexity that can be understood by automatic methods presently available. Creating systems that can inspect their own code, for the purpose of improving their own operation, thus requires that we first solve the problem we started out to solve, namely, the creation of an artificial general intelligence. We need to move away from programming languages with complex syntax and semantics (*all* programming languages intended for humans), towards transparent programming languages with simple syntax and simple operational semantics. Such programming languages must have fewer basic atomic operations, and their combinatorics would be based on simpler principles than current programming languages. A foundational mechanism of such a programming language is likely to be small and large-scale pattern matching: Systems built with it would likely be fairly uniform in their semantics, from the small scale to the gross architecture level, as then the same small number of pattern matching operations could be used throughout the system – regardless of the level of granularity – to detect, compare, add, delete, and improve any function implemented at any level of detail, from code snippets to large architectural constructs.

Small “white-box” (transparent) components, executed asynchronously, where each component implements one of only a few primitive functions, could help streamline the assembly of architectural components. As long as each component is based on a few fundamental principles, it can easily be inspected; assemblies of these will thus also be easily inspectable, and in turn enable the detection (identification, analysis, and modification) of functional patterns realized by even large parts of an architecture. This is a prerequisite for implementing automatic evaluation and learning operational semantics, which lies at the heart of constructivist AI. Incidentally, this is also what is called for to enable transversal functions implementing the kinds of introspection, system-wide learning and dynamic attention which I have already argued as being necessary for artificial general intelligence.

How small need the components be? Elsewhere we have pointed out the need to move towards what we call “peewee-size” granularity [40] – systems composed of hundreds of

thousands of small modules, possibly millions.⁹ To see why the size of the smallest modifiable entities must be small, we need only look at a hypothetical cognitive operation involving hierarchical summation of excitation signals at several levels of detail: at the lowest as well as the highest levels the system – e.g. for improving its operation – may need to change addition to subtraction in particular places. In this case the operation being modified is addition. In a large system we are likely to find a vast number of such cases where, during its growth, a system needs to modify its operation at such low levels of functional detail. If each peewee-size module is no larger than a lambda term or small function written in e.g. C++ we have reached the “code level,” as normally meant by that term – this should thus be of a sufficiently low-level of granularity for building highly flexible systems: self-constructive on temporal and complexity scales that we consider useful, yet running on hardware that is already available.

I am aware of only one architecture that has actually implemented such an approach, the *Loki* system, which was built using the Ikon Flux framework [23], a system based on Lambda terms. The system implemented a live virtual performer in the play *Roma Amor* which ran for a number of months at Cite des Sciences et de L’Industrie in Paris in 2005, proving beyond question that this approach is tractable. Whether the extremely small size of peewee granularity is *required* for self-construction or whether larger components can be used is an important question that we are unable to answer at the moment. But whatever their size, the components – architectural building blocks – must be expressible using simple syntax, as rich syntax begets rich semantics, and rich semantics call for smarter self-inspection mechanisms whose required smarts eventually rise above a threshold of complexity beyond which self-construction and self-organization can be bootstrapped, capsizing the whole attempt. The finer the granularity and the simpler the syntax the more likely it is to succeed in this regard; however, there may also be a lower bound, i.e. building blocks should not be “too simple”; if the operational semantics is kept at a level that is “small enough but not smaller” than what can support self-inspection, there is reason to believe a window opens up within which both practical and powerful components can be realized.

⁹These numbers can be thought of as a rough guide – the actual number for any such architecture will of course depend on a host of things that are hard to currently foresee, including processor speed, cost of memory transactions, architectural distributedness, and more.

9.5.5 *Integration and Architecture Metaconstruction*

Architectural meta-programming is needed to handle larger and more complex systems [3, 31], scaling up to systems with architectural designs that are more complex than even the most complex systems yet engineered, such as microprocessors, the Terrestrial telephone network, or the largest known natural neural networks [24]. A cognitive architecture supporting many of the features seen in natural intelligence will be highly coordinated and highly *integrated* – more so than probably any man-made dynamic system today. All of the issues already discussed in this section are relevant to achieving architectural meta-programming and integration: General principles for learning a variety of contexts can equally well be applied to the architecture itself, which then becomes yet another context.

Constructivist AI will certainly be easier if we find a “cognitive principle” as hypothesized by [6], where the same small set of basic principles can be used throughout to construct every function of a cognitive system. Perhaps fractal architectures – exhibiting self-similarity at multiple levels of granularity – based on simple operational semantics is just that principle. But it is fairly unlikely that a single principle alone will open up the doors to artificial general intelligence – I find it more likely to be based around something like the numerous electro-spatio-temporal principles, rooted in physics and chemistry, that make a car engine run than, say, the principle of flight. Either way, there is no getting around focusing on methods that deal more efficiently with large, distributed, semi-autonomously evolving architectures with heterogeneous functionality and a high degree of flexibility at coarse-grain levels. New meta-programming languages, constructivist design methodologies, and powerful visualization systems must be developed for significant progress to be made. Transversal functions, as described above, are what ultimately forces us to look at the whole architecture when thinking about our new methodology: Without taking the operation of the whole into account, pan-architectural features are precluded. The transition to architectures built via a constructivist methodology will be challenging.

9.6 Conclusions

The hope for generally intelligent machines has not disappeared, yet functions critical to generally intelligent systems continue to be largely ignored, examples being the ability to learn to operate in new environments, introspection, and pan-architectural attention.

Systems whose gross architecture is mostly designed from the top-down and programmed by hand, constructionist AI, has been the norm since the field’s inception, more

than 50 years ago. Few methodologies have been proposed specifically for AI and researchers have relied on standard software methodologies (with small additions and modifications), one of the more popular ones in recent times being object-oriented programming and component-based architectures. As far as large AI systems go these methodologies rely on fairly primitive tools for integration and have generally resulted in brittle systems with little or no adaptation ability and targeted domain application.

Based on the weaknesses inherent in current practices, the conclusion argued for here is that the limitations of present AI software systems cannot be addressed through incremental improvement of current practices, even assuming continued exponential growth of computing power, because the approach has fundamental theoretical and practical limitations: AI systems built to date show that while some integration is possible using current software development methods and extensions thereof (cf. [37]), the kind of deep integration needed for developing general artificial intelligence is unlikely to be attained this way. Standard software development methods do not scale; they assume that their results can be linearly combined, but this is unlikely to produce systemic features that seem key in general intelligence; too many difficult problems, including a freely roving attentional mechanism, equally capable of real-world inspection and introspection, system-wide learning, cognitive growth and improvement, to take some key examples, would be left by the wayside. To create generally intelligent systems we will need to build significantly larger and more complex systems than can be built with present methods.

To address the limitations of present methodologies a paradigm shift is needed – a shift towards *constructivist AI*, comprised of new methodologies that emphasize auto-generated code and self-organization. Constructivist AI calls for a very different approach than offered by traditional software methodologies, shifting the focus from manual “brick-laying” to creating the equivalent of self-constructing “factories”. Architectures would be formed through interactions between flexible autonomous auto-construction principles, where a complex environment and initial “seed” code would interact to automatically create the kinds of architectures needed for general-purpose intelligence.

In this paper I have outlined some of the key topics that need to be advanced in order for this paradigm shift to happen, including a stronger emphasis on feedback loops, temporal grounding, architecture metaprogramming and integration, pan-architectural pattern matching and transparent operational semantics with small white-box components. The list, while non-exhaustive, clearly illustrates the relatively large shift in focus that needs to happen, as most of these topics are not well understood today. To increase our

chances of progress towards artificial general intelligence, future work in AI should focus on constructivist-based tools including new development environments, programming languages, and architectural metaconstruction principles.

Acknowledgments

This is an updated version of my BICA 2009 keynote paper [36]. I would like to thank Eric Nivel for brilliant insights and numerous discussions on these topics, as well as excellent suggestions for improving the paper. Big thanks to Pei Wang, Hannes H. Vilhjalmsón, Deon Garrett, Kevin McGee, Hrafn Th. Thorisson and Gudny R. Jonsdottir for valuable comments and suggestions for improvement, and to the anonymous reviewers for helpful comments. Thanks to the HUMANOBS team for helping lay the groundwork for an exciting future. This work was supported in part by the EU-funded project HUMANOBS: Humanoids That Learn Socio-Communicative Skills Through Observation, contract no. FP7-STREP-231453 (www.humanobs.org), and by a Strategic Research Programme Centres of Excellence and Research Clusters 2009-2016 project grant (IIIM; www.iiim.is) awarded by the Science and Technology Policy Board of Iceland and managed by the Icelandic Center for Research (Rannís; www.rannis.is).

Bibliography

- [1] Anderson, J. R. (1996). Act: A simple theory of complex cognition, *American Psychologist* **51**, pp. 355–365.
- [2] Barrett, H. C. and Kurzban, R. (2006). Modularity in cognition: Framing the debate, *Psychological Review* **113**(3), pp. 628–647.
- [3] Baum, E. B. (2009). Project to build programs that understand, in *Proceedings of the Second Conference on Artificial General Intelligence*, pp. 1–6.
- [4] Brooks, R. A. (1986). Robust layered control system for a mobile robot, *IEEE Journal of Robotics and Automation* **2**(1), p. 14–23.
- [5] Bryson, J. (2003). The behavior-oriented design of modular agent intelligence, *Lecture notes in computer science* **2592**, pp. 61–76.
- [6] Cassimatis, N. (2006). A cognitive substrate for achieving human-level intelligence, *A.I. Magazine* **27**(2), pp. 45–56.
- [7] Drescher, G. L. (1991). *Made-up minds: a constructivist approach to artificial intelligence* (M.I.T. Press, Boston, Massachusetts).
- [8] Franklin, S. (2011). Global workspace theory, shanahan, and LIDA, *International Journal of Machine Consciousness* **3**(2).
- [9] Froese, T. (2007). On the role of AI in the ongoing paradigm shift within the cognitive sciences, *50 Years of Artificial Intelligence - Lecture Notes in Computer Science* **4850**, pp. 63–75.
- [10] Garel, S. and Rubenstein, J. L. R. (2004). Patterning of the cerebral cortex, in M. S. Gazzaniga (ed.), *The Cognitive Neurosciences III*, pp. 69–84.

- [11] Garlan, D., Allen, R. and Ockerbloom, J. (1995). Architectural mismatch or why it's hard to build systems out of existing parts, in *Proceedings of the Seventeenth International Conference on Software Engineering*, pp. 179–185.
- [12] Heylighen, F. and Joslyn, C. (2001). Cybernetics and second order cybernetics, in R. A. Mayers (ed.), *Encyclopedia of Physical Science and Technology*, Vol. 4 (Academic Press), pp. 155–170.
- [13] Hsiao, K., Gorniak, P. and Roy, D. (2005). Netp: A network API for building heterogeneous modular intelligent systems, in *Proceedings of AAAI 2005 Workshop on modular construction of human-like intelligence. AAAI Technical Report WS-05-08*, pp. 24–31.
- [14] Iizuka, H. and Paolo, E. A. D. (2007). Toward spinozist robotics: Exploring the minimal dynamics of behavioural preference, *Adaptive Behavior* **15(4)**, pp. 359–376.
- [15] Jonsdottir, G. R., Thórisson, K. R. and Eric, N. (2008). Learning smooth, human-like turntaking in realtime dialogue, in *IVA '08: Proceedings of the 8th international conference on Intelligent Virtual Agents* (Springer-Verlag, Berlin, Heidelberg), ISBN 978-3-540-85482-1, pp. 162–175, http://dx.doi.org/10.1007/978-3-540-85483-8_17.
- [16] Laird, J. (2008). Extending the soar cognitive architecture, in *Proceedings of the 2008 conference on Artificial General Intelligence*, pp. 224–235.
- [17] Lee, E. E. (2009). Computing needs time, *Communications of the ACM* **52(5)**, pp. 70–79.
- [18] Martin, D., Cheyer, A. and Moran, D. (1999). The open agent architecture: A framework for building distributed software systems, *Applied Artificial Intelligence* **13(1-2)**, pp. 91–128.
- [19] Moore, G. E. (1965). Cramming more components onto integrated circuits, *Electronics Review* **31(8)**.
- [20] Newell, A. and Simon, H. A. (1976). Computer science as empirical enquiry: Symbols and search, *Communications of the Association for Computing Machinery* **19(3)**, p. 113–126.
- [21] Ng-Thow-Hing, V., List, T., Thórisson, K. R., Lim, J. and Wormer, J. (2007). Design and evaluation of communication middleware in a distributed humanoid robot architecture, in *IROS '07 Workshop: Measures and Procedures for the Evaluation of Robot Architectures and Middleware*.
- [22] Ng-Thow-Hing, V., Thórisson, K. R., Sarvadevabhatla, R. K., Wormer, J. and List, T. (2009). Cognitive map architecture: Facilitation of human-robot interaction in humanoid robots, *IEEE Robotics & Automation* **16(1)**, pp. 55–66.
- [23] Nivel, E. (2007). Ikon flux 2.0, Tech. rep., Reykjavik University Department of Computer Science, technical Report RUTR-CS07006.
- [24] Oshio, K., Morita, S., Osana, Y. and Oka, K. (1998). C. elegans synaptic connectivity data, *Technical Report of CCEP, Keio Future, No. 1, Keio University*.
- [25] Pezzulo, G. and Calvi, G. (2007). Designing modular architectures in the framework akira, *Multiagent and Grid Systems*, pp. 65–86.
- [26] Piaget, J. (1950). *The Psychology of Intelligence* (Routledge and Kegan Paul, London, England).
- [27] Pollock, J. L. (2008). Oscar: An architecture for generally intelligent agents, *Frontiers in Artificial Intelligence and Applications* **171**, pp. 275–286.
- [28] Rao, A. S. and Georgeff, M. P. (1991). Modeling rational agents within a BDI-architecture, in J. Allen, R. Fikes and E. Sandewall (eds.), *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning* (Morgan Kaufmann publishers Inc.: San Mateo, CA, USA), pp. 473–484.
- [29] Saemundsson, R. J., Thórisson, K. R., Jonsdottir, G. R., Arinbjarnar, M., Finnsson, H., Gudnason, H., Hafsteinsson, V., Hannesson, G., Isleifsdottir, J., Jóhannsson, Á., Kristjánsson, G. and Sigmundarson, S. (2006). Modular simulation of knowledge development in industry: A multi-level framework, in *WEHIA - Proc. of the First Intl. Conf. on Economic Science with Heterogeneous Interacting Agents* (Bologna, Italy).

- [30] Salthe, S. N. and Matsuno, K. (1995). Self-organization in hierarchical systems, *Journal of Social and Evolutionary Systems* **18**(4), pp. 327–3.
- [31] Sanz, R., López, I., Rodríguez, M. and Hernández, C. (2007). Principles for consciousness in integrated cognitive control, in *Neural Networks*, Vol. 20, pp. 938–946.
- [32] Sun, R., Merrill, E. and Peterson, T. (2001). From implicit skills to explicit knowledge: A bottom-up model of skill learning, *Cognitive Science* **25**, pp. 203–244.
- [33] Sutton, P., Arkins, R. and Segall, B. (2001). Supporting disconnectedness - transparent information delivery for mobile and invisible computing, in *CCGrid 2001 IEEE International Symposium on Cluster Computing and the Grid*, pp. 277–285.
- [34] Swanson, L. W. (2001). Interactive brain maps and atlases, in M. A. Arbib and J. S. Grethe (eds.), *Computing the Brain* (Academic Press), pp. 167–177.
- [35] Thórisson, K. R. (2008). Modeling multimodal communication as a complex system. in I. Wachsmuth and G. Knoblich (eds.), *ZiF Workshop, Lecture Notes in Computer Science*, Vol. 4930 (Springer), ISBN 978-3-540-79036-5, pp. 143–168.
- [36] Thórisson, K. R. (2009). From constructionist to constructivist A.I. in A. Samsonovich (ed.), *Keynote, AAAI Fall Symposium Series: Biologically Inspired Cognitive Architectures; AAAI Tech Report FS-09-01* (AAAI press), pp. 175–183.
- [37] Thórisson, K. R., Benko, H., Arnold, A., Abramov, D., Maskey, S. and Vaseekaran, A. (2004). Constructionist design methodology for interactive intelligences, *A.I. Magazine* **25**(4), pp. 77–90.
- [38] Thórisson, K. R. and Jonsdottir, G. R. (2008). A granular architecture for dynamic realtime dialogue, in *Intelligent Virtual Agents, IVA08*, pp. 1–3.
- [39] Thórisson, K. R., List, T., Pennock, C. and DiPirro, J. (2005). Whiteboards: Scheduling blackboards for semantic routing of messages & streams. in *AAAI-05, AAAI Technical Report WS-05-08*, pp. 8–15.
- [40] Thórisson, K. R. and Nivel, E. (2009). Achieving artificial general intelligence through peewee granularity, in *Proceedings of the Second Conference on Artificial General Intelligence*, pp. 222–223.
- [41] van Gelder, T. J. (1995). What might cognition be, if not computation? *Journal of Philosophy* **91**, pp. 345–381.
- [42] von Foerster, H. (1995). *The Cybernetics of Cybernetics (2nd edition)*, 2nd edn. (FutureSystems Inc.).
- [43] Wang, P. (2004). Toward a unified artificial intelligence, in *In Papers from the 2004 AAAI Fall Symposium on Achieving Human-Level Intelligence through Integrated Research and Systems*, pp. 83–90.
- [44] Wang, P. (2005). Experience-grounded semantics: A theory for intelligent systems, in *Cognitive Systems Research* (Springer-Verlag), pp. 282–302.
- [45] Wang, P. (2006). *Rigid Flexibility: The Logic of Intelligence* (Springer).
- [46] Zitterbart, M. and De Meer, H. (eds.) (2011). *International Workshop on self-organizing systems (IWSOS 2011)* (Springer, New York, NY, USA).

Chapter 10

Towards an Actual Gödel Machine Implementation: A Lesson in Self-Reflective Systems

Bas R. Steunebrink and Jürgen Schmidhuber
IDSIA & University of Lugano, Switzerland

`{bas,juergen}@idsia.ch`

Recently, interest has been revived in self-reflective systems in the context of Artificial General Intelligence (AGI). An AGI system should be intelligent enough to be able to reason about its own program code, and make modifications where it sees fit, improving on the initial code written by human programmers. A pertinent example is the Gödel Machine, which employs a proof searcher—in parallel to its regular problem solves duties—to find a self-rewrite of which it can prove that it will be beneficial. Obviously there are technical challenges involved in attaining such a level of self-reflection in an AGI system, but many of them are not widely known or properly appreciated. In this chapter we go back to the theoretical foundations of self-reflection and examine the (often subtle) issues encountered when embarking on actually implementing a self-reflective AGI system in general and a Gödel Machine in particular.

10.1 Introduction

An Artificial General Intelligence (AGI) system is likely to require the ability of self-reflection; that is, to inspect and reason about its own program code and to perform comprehensive modifications to it, while the system itself is running. This is because it seems unlikely that a human programmer can come up with a completely predetermined program that satisfies sufficient conditions for general intelligence, without requiring adaptation. Of course self-modifications can take on different forms, ranging from simple adaptation of a few parameters through a machine learning technique, to the system having complete read and write access to its own currently running program code [1]. In this chapter we consider the technical implications of approaching the latter extreme, by building towards a pro-

programming language plus interpreter that allows for complete inspection and manipulation of its own internals in a safe and easily understandable way.

The ability to inspect and manipulate one's own program code is not novel; in fact, it was standard practice in the old days when computer memory was very limited and expensive. In recent times, however, programmers are discouraged of using self-modifying code because it is very hard for human programmers to grasp all consequences (especially over longer timespans) of self-modifications and thus this practice is considered error-prone. Consequently, many modern (high-level) programming languages severely restrict access to internals (such as the call stack) or hide them altogether. There are two reasons, however, why we should not outlaw writing of self-modifying code forever. First, it may yet be possible to come up with a programming language that allows for writing self-modifying code in a safe and easy-to-understand way. Second, now that automated reasoning systems are becoming more mature, it is worthwhile investigating the possibility of letting the machine—instead of human programmers—do all the self-modifications based on automated reasoning about its own programming.

As an example of a system embodying the second motivation above we will consider the Gödel Machine [2–5], in order to put our technical discussion of self-reflection in context. The fully self-referential Gödel Machine is a universal artificial intelligence that is theoretically optimal in a certain sense. It may interact with some initially unknown, partially observable environment to solve arbitrary user-defined computational tasks by maximizing expected cumulative future utility. Its initial algorithm is not hardwired; it can completely rewrite itself without essential limits apart from the limits of computability, provided a proof searcher embedded within the initial algorithm can first prove that the rewrite is useful, according to its formalized utility function taking into account the limited computational resources. Self-rewrites due to this approach can be shown to be *globally optimal* with respect to the initial utility function (e.g., a Reinforcement Learner's reward function), relative to Gödel's well-known fundamental restrictions of provability [6].

In the next section we provide an outline of the specification of the Gödel Machine concept, which then provides the context for a subsequent discussion of self-reflective systems. As alluded to above, a Gödel Machine implementation calls for a system with the ability to make arbitrary changes to its currently running program. But what does that mean? What is a change, how arbitrary can a change be, and what does a running program actually look like? We will see that these are nontrivial questions, involving several subtle but important

issues. In this chapter we provide an overview of the issues involved and ways to overcome them.

10.2 The Gödel Machine Concept

One can view a Gödel Machine as a program consisting of two parts. One part, which we will call the *solver*, can be any problem-solving program. For clarity of presentation, we will pretend the *solver* is a Reinforcement Learning (RL) [7] program interacting with some external environment. This will provide us with a convenient way of determining *utility* (using the RL program's reward function), which will be an important topic later on. But in general, no constraints are placed on the *solver*. The second part of the Gödel Machine, which we will call the *searcher*, is a program that tries to improve the entire Gödel Machine (including the *searcher*) in a provably optimal way. This *searcher* will be the main topic of this section.

A Gödel Machine's *searcher* is said to hold a function computing the machine's expected cumulative future utility. This utility function is then used to construct a *target theorem*; that is, the Gödel Machine is *only* allowed to perform a self-modification if it is provably useful to do so. As we will see below, the *target theorem* specifies precisely when a rewrite is considered "useful."

Basically, the function for determining the expected cumulative future utility, shown as u below, sums all rewards for all future time steps. Here "time steps" actually means not clock ticks, but execution of elementary instructions. Indeed, each instruction takes time to execute, so if we can find a way to explicitly represent the instructions that are going to be executed in the future, we automatically have a window into a future time. An obvious choice of such a representation is the *continuation*, which is a well-studied concept in light of λ -calculus-based programming languages (e.g., Lisp, Scheme) [8]. Intuitively, a continuation can be seen as the opposite of a call stack; instead of showing "where we came from," a continuation explicitly shows "what is going to happen next." Note that in all but the simplest cases, a continuation will only be partially expanded. For example, suppose the current continuation is `{ A(); if B() then C() else D() }`; this continuation specifies that the next thing to be done is expanding `A` and executing its body, and then the conditional statement will be executed, which means that first `B` will be expanded and depending on its result, either `C` or `D` will be expanded. Note that before executing `B`, it is

not clear yet whether C or D will be executed in the future; so it makes no sense to expand either of them before we know the result of B.

In what follows we consistently use subscripts to indicate where some element is encoded. u is a function of two parameters, $u_{\bar{s}}(s, c)$, which represents the expected cumulative future utility of running continuation c on state s . Here \bar{s} represents the evaluating state (where u is encoded), whereas s is the evaluated state. The reason for this separation will become clear when considering the specification of u :

$$u_{\bar{s}}(s, c) = \mathbb{E}_{\mu_s, M_s} [u'] \quad \text{with } u'(env) = r_{\bar{s}}(s, env) + \mathbb{E}_{\kappa_c, K_c} [u_{\bar{s}} | env] \quad (10.1)$$

As indicated by subscripts, the representation M of the external environment is encoded inside s , because all knowledge a Gödel Machine has must be encoded in s . For clarity, let M be a set of bitstrings, each constituting a representation of the environment held possible by the Gödel Machine. μ is a mapping from M to probabilities, also encoded in s . c encodes not only a (partially expanded) representation of the instructions that are going to be executed in the future, but also a set K of state–continuation pairs representing which possible next states and continuations can result from executing the first instruction in c , and a mapping κ from K to probabilities. So μ and κ are (discrete) probability distributions on sample spaces M and K , respectively. $r_{\bar{s}}(s, env)$ determines whether state s is rewarding given environment env . For example, in the case where *solver* (which is part of s) is an RL program, $r_{\bar{s}}(s, env)$ will be nonzero only when s represents a state just after performing an input receiving instruction. Finally, the term $\mathbb{E}_{\kappa_c, K_c} [u_{\bar{s}} | env]$ recurses on u with the state and continuation following from executing the next instruction in continuation c .

It is *crucial* to note that u and r are taken from the evaluating state \bar{s} , *not* from the state s under evaluation. Doing the latter would break the global optimality [5] of the Gödel Machine, because then it would be capable of rewriting its utility function in arbitrary, even detrimental ways. Therefore, the original, unchanged utility function of \bar{s} must be used to first show that a rewrite is useful. In contrast, μ and M are taken from s , because for any interesting Gödel Machine, c will both contain instructions that affect the external environment (e.g., output to actuators) and instructions that update the internal model of the environment (encoded in s) accordingly. To make use of these updated models, u must extract them from s . Likewise, κ and K are taken from c , because after each instruction the continuation necessarily changes, resulting in new κ and K .

It is also important to notice that the function u is recursive. In formula (10.1), a Gödel Machine's life ends when the continuation sample space K_c is empty. In that case, the term $\mathbb{E}_{\kappa_c, K_c} [u_{\bar{s}} | env]$ is zero and recursion on u stops. In the most common case, how-

ever, an instruction will be deterministic, e.g., $K_c = \{(s', c')\}$. Then formula (10.1) reduces to $u_{\bar{s}}(s, c) = \mathbb{E}_{\mu_s, M_s} [r_{\bar{s}}(s, \cdot)] + u_{\bar{s}}(s', c')$. So for chains of deterministic state transitions, the expected cumulative future utility reduces to a summation of expected rewards in the successive states. But in e.g. RL settings, even those expected rewards are usually zero, because most instructions are not involved with receiving information from the environment (which in RL is usually the source of rewards).

We are now in a position to introduce the *target theorem*, which is based on the utility function u . Let s be an expected future state *before* some time t . Furthermore, let $c^* = \{\text{wait_until}(t); \text{switchprog}()\}$ denote the continuation that waits until time t (e.g., by continually calling a primitive clock function until the returned time tests as greater or equal to t) and then executes *switchprog*, which may inspect and modify any part of s . Now $u_{\bar{s}}(s, c^*)$ tells us what is the expected cumulative future utility of running *switchprog* on s at time t . If this utility compares positively against the expected cumulative future utility of *not* switching, then the Gödel Machine must perform the switch at time t . Thus, the *searcher* is to find a state s , a time t , and a nullary procedure *switchprog* such that $\text{tt}_{\bar{s}}(s, t, \text{switchprog})$ holds, where:

$$\text{tt}_{\bar{s}}(s, t, \text{switchprog}) = [u_{\bar{s}}(s, \{\text{wait_until}_s(t); \text{switchprog}()\}) > u_{\bar{s}}(s, \{\text{scheduler}_s()\})] \quad (10.2)$$

where *scheduler* is the procedure that interleaves the Gödel Machine's *solver* and *searcher*. Thus $\{\text{scheduler}_s()\}$ denotes the continuation that “proceeds as usual” from s (including searching for new *switchprogs*). Note that, since *switchprog* may overwrite the scheduler, it must as its last instruction invoke a new scheduler (if needed), or simply invoke the old scheduler if it has been left intact.

There are two ways in which this target theorem can be used to find provably useful self-rewrites: (1) generate a candidate *switchprog* (e.g., a program performing optimization techniques also applied by state-of-the-art compilers), generate a future switch time t at least as far ahead as the time the evaluation of the utility of running *switchprog* is expected to take, and make s a copy of the current state \bar{s} ; then run both continuations specified by tt on s (this is safe because s acts like a sandbox) until the inequality can be shown to be true or false, or until the current time exceeds t ; if the inequality is false or time has run out, generate a new *switchprog* and start over; else, swap the current state for s , wait until time t and execute *switchprog*. (2) Start with an axiomatic system with axiom and rules, and generate theorems (or generate programs that calculate theorems [5]), testing them for equivalence with tt . The first method can be seen as a top-down approach to proving tt and

the latter as bottom-up. Hybrid approaches are also possible, so this specification of the target theorem allows for a *family* of possible Gödel Machine implementations [9].

To make sure the Gödel Machine is at least *asymptotically* optimal even before the first self-rewrite, it may be initialized with Hutter’s non-self-referential but *asymptotically fastest algorithm for all well-defined problems* HSEARCH [10], which uses a hardwired brute force proof searcher and ignores the costs of proof search. Given some problem, the Gödel Machine may decide to replace HSEARCH by a faster method suffering less from large constant overhead, but even if it does not, its performance will not be less than asymptotically optimal.

Having explained an example of a system needing self-reflection, we will now move on to explaining the technical aspects of attaining self-reflection.

10.3 The Theoretical Foundations of Self-Reflective Systems

In the mid-eighties, there was a fashion for reflective interpreters, a fad that gave rise to a remarkable term: “reflective towers.” Just imagine a marsh shrouded in mist and a rising tower with its summit lost in gray and cloudy skies—pure Rackham! (...) Well, who hasn’t dreamed about inventing (or at least having available) a language where anything could be redefined, where our imagination could gallop unbridled, where we could play around in complete programming liberty without trammel nor hindrance? [8]

The reflective tower that Queinnee is so poetically referring to, is a visualization of what happens when performing self-reflection.¹ A program running on the n th floor of the tower is the effect of an evaluator running on the $(n - 1)$ th floor. When a program running on the n th floor performs a reflective instruction, this means it gains access to the state of the program running at the $(n - 1)$ th floor. But the program running on the n th floor can also invoke the evaluator function, which causes a program to be run on the $(n + 1)$ th floor. If an evaluator evaluates an evaluator evaluating an evaluator etc. etc., we get the image of “a rising tower with its summit lost in gray and cloudy skies.” If a program reflects on a reflection of a reflection etc. etc., we get the image of the base of the tower “shrouded in mist.” What happens were we to stumble upon a ground floor? In the original vision of the reflective tower, there is none, because it extends infinitely in both directions (up and down). Of course in practice such infinities will have to be relaxed, but the point is that it will be of great interest to see exactly when, where, and how problems will surface, which is precisely the point of this section.

¹*Reflection* can have different meanings in different contexts, but here we maintain the meaning defined in the introduction: the ability to inspect and modify one’s own currently running program.

According to Queinnec, there are two things that a reflective interpreter must allow for. First:

Reflective interpreters should support introspection, so they must offer the programmer a means of grabbing the computational context at any time. By “computational context,” we mean the lexical environment and the continuation [8].

The *lexical environment* is the set of bindings of variables to values, for all variables in the lexical scope² of the currently running program. Note that the lexical environment has to do only with the internal state of a program; it has nothing to do with the *external environment* with which a program may be interacting through actuators. The *continuation* is a representation of future computations, as we have seen in the previous section. And second:

A reflective interpreter must also provide means to modify itself (a real thrill, no doubt), so (...) that functions implementing the interpreter are accessible to interpreted programs [8].

In this section we will explore what these two requirements for self-reflection mean technically, and to what extent they are realizable theoretically and practically.

10.3.1 Basic λ -calculus

Let us first focus on representing a computational context; that is, a lexical environment plus a continuation. The most obvious and well-studied way of elucidating these is in λ -calculus, which is a very simple (theoretical) programming language. The notation of expressions in λ -calculus is a bit different from usual mathematical notation; for example, parentheses in function application are written on the outside, so that we have $(f\ x)$ instead of the usual $f(x)$. Note also the rounded parentheses in $(f\ x)$; they are part of the syntax and always indicate function application, i.e., they are not allowed freely. Functions are made using *lambda abstraction*; for example, the identity function is written as $\lambda x.x$. So we write a λ symbol, then a variable, then a period, and finally an expression that is the *body* of the function. Variables can be used to name expression; for example, applying the identity function to y can be written as “ $g(y)$ where $g(x) = x$,” which is written in λ -calculus as $(\lambda g.(g\ y)\ \lambda x.x)$.

Formally, the language (syntax) Λ_1 of basic λ -calculus is specified using the following recursive grammar.

$$\Lambda_1 ::= v \mid \lambda v.\Lambda_1 \mid (\Lambda_1\ \Lambda_1) \tag{10.3}$$

²For readers unfamiliar with the different possible scoping methods, it suffices to know that lexical (also called static) scoping is the intuitive, “normal” method found in most modern programming languages.

This expresses succinctly that (1) if x is a variable then x is an expression, (2) if x is a variable and M is an expression then $\lambda x.M$ is an expression (*lambda abstraction*), and (3) if M and N are expressions then $(M N)$ is an expression (*application*).

In pure λ -calculus, the only “operation” that we can perform that is somewhat akin to evaluation, is β -reduction. β -reduction can be applied to a Λ_1 (sub)expression if that (sub)expression is an application with a lambda abstraction in the operator position. Formally:

$$(\lambda x.M N) \xrightarrow{\beta} M[x \leftarrow N] \quad (10.4)$$

So the “value” of supplying an expression N to a lambda abstraction $\lambda x.M$ is M with all occurrences of x replaced by N . Of course we can get into all sorts of subtle issues if several nested lambda abstractions use the same variable names, but let’s not go into that here, and assume a unique variable name is used in each lambda abstraction.

At this point one might wonder how to *evaluate* an arbitrary Λ_1 expression. For example, a variable should evaluate to its binding, but how do we keep track of the bindings introduced by lambda abstractions? For this we need to introduce a *lexical environment*, which contains the bindings of all the variables in scope. A lexical environment is historically denoted using the letter ρ and is represented here as a function taking a variable and returning the value bound to it. We can then specify our first evaluator \mathcal{E}_1 as a function taking an expression and a lexical environment and returning the value of the expression. In principle this evaluator can be specified in any formalism, but if we specify it in λ -calculus, we can easily build towards the infinite reflective tower, because then the computational context will have the same format for both the evaluator and evaluated expression.

There are three syntactic cases in language Λ_1 and \mathcal{E}_1 splits them using double bracket notation. Again, we have to be careful not to confuse the specification language and the language being specified. Here they are both λ -calculus in order to show the reflective tower at work, but we should not confuse the different floors of the tower! So in the specification below, if the Λ_1 expression between the double brackets is floor n , then the right-hand side of the equal sign is a Λ_1 expression on floor $n - 1$. \mathcal{E}_1 is then specified as follows.

$$\mathcal{E}_1[[x]] = \lambda \rho. (\rho x) \quad (10.5)$$

$$\mathcal{E}_1[[\lambda x.M]] = \lambda \rho. \lambda \varepsilon. (\mathcal{E}_1[[M]] \rho[x \leftarrow \varepsilon]) \quad (10.6)$$

$$\mathcal{E}_1[[(M N)]] = \lambda \rho. ((\mathcal{E}_1[[M]] \rho) (\mathcal{E}_1[[N]] \rho)) \quad (10.7)$$

It should first be noted that all expression on the right-hand side are lambda abstractions expecting a lexical environment ρ , which is needed to look up the values of variables. To

start evaluating an expression, an empty environment can be provided. According to the first case, the value of a variable x is its binding in ρ . According to the second case, the value of a lambda abstraction is itself a function, waiting for a value ε ; when received, M is evaluated in ρ extended with a binding of x to ε . According to the third case, the value of an application is the value of the operator M applied to the value of the operand N , assuming the value of M is indeed a function. Both operator and operand are evaluated in the same lexical environment.

For notational convenience, we will abbreviate nested applications and lambda abstractions from now on. So $((f x) y)$ will be written as $(f x y)$ and $\lambda x.\lambda y.M$ as $\lambda xy.M$.

We have now introduced an explicit lexical environment, but for a complete computational context, we also need a representation of the continuation. To that end, we rewrite the evaluator \mathcal{E}_1 in *continuation-passing style* (CPS). This means that a continuation, which is a function historically denoted using the letter κ , is extended whenever further evaluation is required, or invoked when a value has been obtained. That way, κ always expresses the future of computations—although, as described before, it is usually only partially expanded. The new evaluator \mathcal{E}'_1 works explicitly with the full computational context by taking both the lexical environment ρ and the continuation κ as arguments. Formally:

$$\mathcal{E}'_1[[x]] = \lambda\rho\kappa.(\kappa(\rho x)) \quad (10.8)$$

$$\mathcal{E}'_1[[\lambda x.M]] = \lambda\rho\kappa.(\kappa\lambda\varepsilon\kappa'.(\mathcal{E}'_1[[M]]\rho[x\leftarrow\varepsilon]\kappa')) \quad (10.9)$$

$$\mathcal{E}'_1[[M N]] = \lambda\rho\kappa.(\mathcal{E}'_1[[M]]\rho\lambda f.(\mathcal{E}'_1[[N]]\rho\lambda x.(f x \kappa))) \quad (10.10)$$

In the first and second case, the continuation is immediately invoked, which means that the future of computations is reduced. In these cases this is appropriate, because there is nothing *extra* to be evaluated. In the third case, however, two things need to be done: the operator and the operand need to be evaluated. In other words, while the operator (M) is being evaluated, the evaluation of the operand (N) is a computation that lies in the future. Therefore the continuation, which represents this future, must be extended. This fact is now precisely represented by supplying to the evaluator of the operator $\mathcal{E}'_1[[M]]$ a new continuation which is an extension of the supplied continuation κ : it is a function waiting for the value f of the operator. As soon as f has been received, this extended continuation invokes the evaluator for the operand $\mathcal{E}'_1[[N]]$, but *again* with a new continuation: $\lambda x.(f x \kappa)$. This is because, while the operand N is being evaluated, there is still a computation lying in the future; namely, the actual invocation of the value (f) of the operator on the value (x) of the operand. At the moment of this invocation, the future of computations is exactly the same as the future before evaluating both M and N ; therefore, the continuation κ of that moment

must be passed on to the function invocation. Indeed, in (10.9) we see what will be done with this continuation: a lambda abstraction evaluates to a binary function, where ε is the value of its operand and κ' is the continuation at the time when the function is actually invoked. This is then also the continuation that has to be supplied to the evaluator of the function's body ($\mathcal{E}'_1[[M]]$).

For the reader previously unfamiliar with CPS it will be very instructive to carefully compare (10.5)–(10.7) with (10.8)–(10.10), especially the application (third) case.

Let's return now to the whole point of explicitly representing the lexical environment and the continuation. Together they constitute the computational context of a program under evaluation; for this program to have self-reflective capabilities, it must be able to “grab” both of them. This is easily achieved now, by adding two special constructs to our language, `grab-r` and `grab-k`, which evaluate to the current lexical environment and current continuation, respectively.

$$\mathcal{E}'_1[[\text{grab-r}]] = \lambda\rho\kappa.(\kappa\rho) \quad (10.11)$$

$$\mathcal{E}'_1[[\text{grab-k}]] = \lambda\rho\kappa.(\kappa\kappa) \quad (10.12)$$

These specifications look deceptively simple, and indeed they are. Because although we now have a means to grab the computational context at any time, there is little we can do with it. Specifically, there is no way of inspecting or modifying the lexical environment and continuation after obtaining them. So they are as good as black boxes.

Unfortunately there are several more subtle issues with the evaluator \mathcal{E}'_1 , which are easily overlooked. Suppose we have a program π in language Λ_1 and we want to evaluate it. For this we would have to determine the value of $(\mathcal{E}'_1[[\pi]] \rho_0 \lambda x.x)$, supplying the evaluator with an initial environment and an initial continuation. The initial continuation is easy: it is just the identity function. The initial environment ρ_0 and the extension of an environment (used in (10.9) but not specified yet) can be specified as follows.

$$\rho_0 = \lambda x.\perp \quad (10.13)$$

$$\rho[x \leftarrow \varepsilon] = \lambda y.\text{if } (= x y) \varepsilon (\rho y) \quad (10.14)$$

So the initial environment is a function failing on every binding lookup, whereas an extended environment is a function that tests for the new binding, returning either the bound value (ε) when the variables match ($(= x y)$), or the binding according to the unextended environment ((ρy)). But here we see more problems of our limited language Λ_1 surfacing: there are no conditional statements (`if`), no primitive functions like `=`, and no constants like \perp , so we are not allowed to write the initial and extended lexical environment as above.

Also the language does not contain numbers to do arithmetics (which has also caused our examples of Λ_1 expressions to be rather abstract). Admittedly, conditionals, booleans, and numbers *can* be represented in pure λ -calculus, but that is more of an academic exercise than a practical approach. Here we are interested in building towards practically feasible self-reflection, so let's see how far we can get by extending our specification language to look more like a “real” programming language.

10.3.2 Constants, Conditionals, Side-effects, and Quoting

Let's extend our very austere language Λ_1 and add constructs commonly found in programming languages, and in Scheme [11] in particular. Scheme is a dialect of Lisp, is very close to λ -calculus, and is often used to study reflection in programming [8, 12, 13]. The Scheme-like language Λ_2 is specified using the following recursive grammar.

$$\Lambda_2 ::= v \mid \lambda v. \Lambda_2 \mid (\Lambda_2 \Lambda_2) \mid c \mid \text{if } \Lambda_2 \Lambda_2 \Lambda_2 \mid \text{set! } v \Lambda_2 \mid \text{quote } \Lambda_2 \quad (10.15)$$

where constants (c) include booleans, numbers, and, notably, primitive functions. These primitive functions are supposed to include operators for doing arithmetics and for inspecting and modifying data structures (including environments and continuations!), as well as IO interactions. The `if` construct introduces the familiar *if-then-else* expression, `set!` introduces assignment (and thereby side-effects), and `quote` introduces quoting (which means treating programs as data).

In order to appropriately model side-effects, we need to introduce the storage, which is historically denoted using the letter σ . From now on the lexical environment (ρ) does not bind variables to values, but to addresses. The storage, then, binds addresses to values. This setup allows `set!` to change a binding by changing the storage but not the lexical environment. The new evaluator \mathcal{E}_2 is then specified as follows.

$$\mathcal{E}_2[[x]] = \lambda \rho \sigma \kappa. (\kappa \sigma (\sigma (\rho x))) \quad (10.16)$$

$$\mathcal{E}_2[[\lambda x.M]] = \lambda \rho \sigma \kappa. (\kappa \sigma \lambda \varepsilon \sigma' \kappa'. (\mathcal{E}_2[[M]] \rho [x \leftarrow \alpha] \sigma' [\alpha \leftarrow \varepsilon] \kappa')) \quad (10.17)$$

$$\mathcal{E}_2[[(M N)]] = \lambda \rho \sigma \kappa. (\mathcal{E}_2[[M]] \rho \sigma \lambda \sigma' f. (\mathcal{E}_2[[N]] \rho \sigma' \lambda \sigma'' x. (f x \sigma'' \kappa))) \quad (10.18)$$

where α is a fresh address. Again, it will be very instructive to carefully compare the familiar (10.8)–(10.10) with (10.16)–(10.18). The main difference is the storage which is being passed around, invoked (10.16), and extended (10.17). The new cases are:

$$\mathcal{E}_2[[c]] = \lambda \rho \sigma \kappa. (\kappa \sigma c) \quad (10.19)$$

$$\mathcal{E}_2[[\text{if } C T F]] = \lambda \rho \sigma \kappa. (\mathcal{E}_2[[C]] \rho \sigma \lambda \sigma' c. (\text{if } c \mathcal{E}_2[[T]] \mathcal{E}_2[[F]] \rho \sigma' \kappa)) \quad (10.20)$$

$$\mathcal{E}_2[\text{set! } x M] = \lambda \rho \sigma \kappa. (\mathcal{E}_2[M] \rho \sigma \lambda \sigma' \varepsilon. (\kappa \sigma' [(\rho x) \leftarrow \varepsilon] \varepsilon)) \quad (10.21)$$

$$\mathcal{E}_2[\text{quote } M] = \lambda \rho \sigma \kappa. (\kappa \sigma M) \quad (10.22)$$

Constants simply evaluate to themselves. Similarly, quoting an expression M returns M unevaluated. Conditional statements force a choice between evaluating the *then*-body ($\mathcal{E}_2[[T]]$) and the *else*-body ($\mathcal{E}_2[[F]]$). Note that in (10.21) only the storage (σ') is changed, such that variable x retains the address that is associated with it in the lexical environment, causing future look-ups (see (10.16)) to return the new value (ε) for x .

As an example of a primitive function, the binary addition operator can be specified as follows.

$$+ = \lambda x \sigma \kappa. (\kappa \sigma \lambda y \sigma' \kappa'. (\kappa' \sigma' (+ x y))) \quad (10.23)$$

Note the recursion; $+$ is specified in terms of the $+$ operator used one floor lower in the reflective tower. The same holds for `if` in (10.20). Where does it bottom out? The marsh is still very much shrouded in mist.

The reason that we cannot specify primitive procedures and constructs nonrecursively at this moment is because the specifications so far say nothing about the data structures used to represent the language constructs. Theoretically this is irrelevant, because the ground floor of the reflective tower is infinitely far away. But the inability to inspect and modify data structures makes it hard to comply with Queinnec's second condition for self-reflection, namely that everything—including the evaluator—should be modifiable. There are now (at least) two ways to proceed: (1) set up a recursive loop of evaluators with shared state to make the reflective tower “float” without a ground floor, or (2) let go of the circular language specification and retire to a reflective bungalow where everything happens at the ground floor. Both options will be discussed in the next two sections, respectively.

10.4 Nested Meta-Circular Evaluators

Using the well-studied technique of the *meta-circular evaluator* [12], it is possible to attain self-reflectivity in any (Turing-complete) programming language. A meta-circular evaluator is basically an interpreter for the same programming language as the one in which the interpreter is written. Especially suitable for this technique are homoiconic languages such as Lisp and in particular its dialect Scheme [11], which is very close to λ -calculus and is often used to study meta-circular evaluators and self-reflection in programming in general [8, 12–19]. So a meta-circular Scheme evaluator is a program written in Scheme

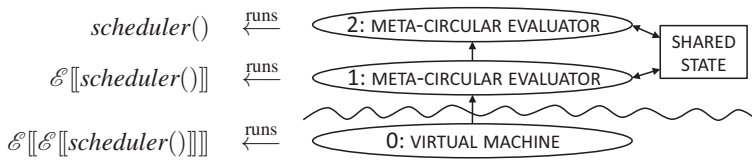


Fig. 10.1 The self-inspection and self-modification required for a Gödel Machine implementation can be attained by having a double nesting of meta-circular evaluators run the Gödel Machine’s *scheduler*. Every instruction is grounded in some virtual machine running “underwater,” but the nested meta-circular evaluators can form a loop of self-reflection without ever “getting their feet wet.”

which can interpret programs written in Scheme. There is no problem with circularity here, because the program running the meta-circular Scheme evaluator itself can be written in any language. For clarity of presentation let us consider how a meta-circular Scheme evaluator can be used to obtain the self-reflectivity needed for e.g. a Gödel Machine.

In what follows, let $\mathcal{E}[\pi]$ denote a call to an evaluator with as argument program π . As before, the double brackets mean that program π is to be taken literally (i.e., unevaluated), for it is the task of the evaluator to determine the value of π . Now for a meta-circular evaluator, $\mathcal{E}[\pi]$ will give the same value as $\mathcal{E}[\mathcal{E}[\pi]]$ and $\mathcal{E}[\mathcal{E}[\mathcal{E}[\pi]]]$ and so on. Note that $\mathcal{E}[\mathcal{E}[\pi]]$ can be viewed as the evaluator reading and interpreting its own source code and determining how the program constituting that source code evaluates π . A very clear account of a complete implementation (in Scheme) of a simple reflective interpreter (for Scheme) is provided by Jefferson *et al.* [13], of which we shall highlight one property that is very interesting in light of our goal to obtain a self-reflective system. Namely, in the implementation by Jefferson *et al.*, no matter how deep one would nest the evaluator (as in $\mathcal{E}[\mathcal{E}[\dots\pi\dots]]$), all levels will share the same *global environment*³ for retrieving and assigning procedures and data. This implies that self-reflection becomes possible when running a program (in particular, a Gödel Machine’s *scheduler*) in a doubly nested meta-circular interpreter with a shared global environment (see figure 10.1). Note that this setup can be attained regardless of the hardware architecture and without having to invent new techniques.

Consider again the quote at the start of section 10.3. It is interesting to note that this paragraph is immediately followed by equally poetic words of caution: “However, we pay

³A global environment can be seen as an extension of the lexical environment. When looking up the value of a variable, first the lexical environment is searched; if no binding is found, the global environment is searched. Similarly for assignment. The global environment is the same in every lexical context. Again, all this has nothing to do with the *external environment*, which lies outside the system.

for this dream with exasperatingly slow systems that are almost incompilable and plunge us into a world with few laws, hardly even any gravity.” We can now see more clearly this world without gravity in figure 10.1: the nested meta-circular evaluators are floating above the water, seemingly without ever getting their feet wet.

But this is of course an illusion. Consider again the third quote, stating that in a self-reflective system the functions implementing the interpreter must be modifiable. But what are those “functions implementing the interpreter?” For example, in the shared global environment, there might be a function called `evaluate` and helper functions like `lookup` and `allocate`. These are all modifiable, as desired. But all these functions are composed of *primitive* functions (such as Scheme’s `cons` and `cdr`) and of syntactical compositions like function applications, conditionals, and lambda abstractions. Are those functions and constructs also supposed to be modifiable? All the way down the “functions implementing the interpreter” can be described in terms of machine code, but we cannot change the instruction set of the processor. The regress in self-modifiability has to end *somewhere*.⁴

10.5 A Functional Self-Reflective System

Taking the last sentence above to its logical conclusion, let us now investigate the consequences of ending the regress in self-modifiability already at the functions implementing the interpreter. If the interpreter interprets a Turing-complete instruction set, then having the ability to inspect and modify the lexical environment and the continuation at any point is already enough to attain *functionally* complete reflective control.

The reason that Queinnec mentions modifiability of the functions implementing the interpreter is probably not for functional reasons, but for timing and efficiency reasons. As we have seen, the ability to “grab the computational context” is enough to attain functional reflection, but it does not allow a program to speed up its own interpreter (*if* it would know how). In this section we will introduce a new interpreter for a self-reflective system that takes a middle road. The functions implementing this interpreter are primitives (i.e., black boxes); however, they are also (1) very few in number, (2) very small, and (3) fast

⁴Unless we consider systems that are capable of changing their own hardware. But then still there are (probably?) physical limits to the modifiability of hardware.

in execution.⁵ We will show that a program being run by this interpreter can inspect and modify itself (including speed upgrades) in a sufficiently powerful way to be self-reflective.

First of all, we need a programming language. Here we use a syntax that is even simpler than classical λ -calculus, specified by the following recursive grammar.

$$\Lambda_3 ::= c \mid n \mid (\Lambda_3 \Lambda_3) \quad (10.24)$$

where c are constants (symbols for booleans and primitive functions) and n are numbers (sequences of digits). There are no special forms (such as `quote`, `lambda`, `set!`, and `if` in Scheme), just function application, where all functions are unary. Primitive binary functions are invoked as, e.g., $((+ 1) 2)$. Under the hood, the only compound data structure is the pair⁶ and an application $(f x)$ is simply represented as a pair $f:x$. An instance of a number takes as much space as one pair (say, 64 bits); constants do not take any space.

For representing what happens under the hood, we extensively use the notation $a:d$, meaning a pair with head a and tail d . The colon associates from right to left, so $a:ad:dd$ is the same as $(a:(ad:dd))$. \perp is a constant used to represent false or an error or undefined value; for example, the result of an integer division by zero is \perp (programs are *never* halted due to an error). The empty list (Scheme: `'()`) is represented by the constant \emptyset .

To look up a value stored in a storage σ at location p , we use the notation $\sigma[p \rightarrow n]$ for numbers and $\sigma[p \rightarrow a:d]$ for pairs. Allocation is denoted as $\sigma[q \stackrel{*}{\leftarrow} n]$ for numbers and $\sigma[q \stackrel{*}{\leftarrow} a:d]$ for pairs, where q is a fresh location where the allocated number or pair can be found. \mathbb{N}_σ is used to denote the set of locations where numbers are stored; all other locations store pairs. The set of constants is denoted as \mathbb{P} . We refer the reader to the appendix for more details on notation and storage whenever something appears unclear.

The crux of the interpreter to be specified in this section is that not only programs are stored as pairs, but also the lexical environment and the continuation. Since programs can build (`cons`), inspect (`car`, `cdr`),⁷ and modify (`set-car!`, `set-cdr!`) pairs, they can then also build, inspect, and modify lexical environments and continuations using these same functions. That is, provided that they are accessible. The interpreter presented below will see to that.

⁵The last point of course depends on the exact implementation, but since they are so small, it will be clear that they can be implemented very efficiently. In our not-so-optimized reference implementation written in C++, one primitive step of the evaluator takes about twenty nanoseconds on a 2009-average PC. Since reflection is possible after every such step, there are about fifty million chances per second to interrupt evaluation, grab the computational context, and inspect and possibly modify it.

⁶Our reference implementation also supports vectors (fixed-size arrays) though.

⁷In Scheme and Λ_3 , `car` and `cdr` are primitive unary functions that return the head and tail of a pair, respectively.

As is evident from (10.24), the language lacks variables, as well as a construct for lambda abstraction. Still, we stay close to λ -calculus by using the technique of De Bruijn indices [20]. Instead of named variables, numbers are used to refer to bound values. For example, where in Scheme the λ -calculus expression $\lambda x.\lambda y.x$ is written as $(\text{lambda } (x\ y)\ x)$, with De Bruijn indices one would write $(\text{lambda } (\text{lambda } 1))$. That is, a number n evaluates to the value bound by the n th enclosing lambda operator (counting the first as zero). A number has to be quoted to be used as a number (e.g., to do arithmetics). But as we will see shortly, neither `lambda` nor `quote` exists in Λ_3 ; however, a different, more general mechanism is employed that achieves the same effect.

By using De Bruijn indices the lexical environment can be represented as a list,⁸ with “variable” lookup being simply a matter of indexing the lexical environment. The continuation will also be represented as a list; specifically, a list of functions, where each function specifies what has to be done as the next step in evaluating the program. So at the heart of the interpreter is a stepper function, which pops the next function from the continuation and invokes it on the current program state. This is performed by the primitive unary function `step`, which is specified as follows. (Note that every n -ary primitive function takes $n + 2$ arguments, the extra two being the continuation and the storage.)

$$\text{step}(\pi)(\kappa, \sigma) = \begin{cases} \varphi(\pi)(\kappa', \sigma) & \text{if } \sigma[\kappa \rightarrow \varphi:\kappa'] \text{ and } \varphi \in \mathbb{F}_1 \\ \text{step}(\varphi')(\kappa', \sigma[\varphi' \leftarrow^* \varphi:\pi]) & \text{if } \sigma[\kappa \rightarrow \varphi:\kappa'] \text{ and } \varphi \in \mathbb{F}_2 \\ \varphi(\pi', \pi)(\kappa', \sigma) & \text{if } \sigma[\kappa \rightarrow (\varphi:\pi'):\kappa'] \text{ and } \varphi \in \mathbb{F}_2 \\ \text{step}(\perp)(\kappa', \sigma) & \text{if } \sigma[\kappa \rightarrow \varphi:\kappa'] \\ \pi & \text{otherwise} \end{cases} \quad (10.25)$$

where \mathbb{F}_1 and \mathbb{F}_2 are the sets of all primitive unary and binary functions, respectively. In the first case, the top of the continuation is a primitive unary function, and `step` supplies π to that function. In the second case, the top is a primitive binary function, but since we need a second argument to invoke that function, `step` makes a pair of the function and π as its first argument. In the third case, where such a pair with a binary function is found at the top of the continuation, π is taken to be the second argument and the binary function is invoked. In the fourth case, the top of the continuation is found not to be a function (because the first three cases are tried first); this is an error, so the non-function is popped from the continuation and `step` continues with \perp to indicate the error. Finally, in the fifth

⁸In Scheme and Λ_3 , a *list* is a tail-linked, \emptyset -terminated sequence of pairs, where the elements of the list are held by the heads of the pairs.

case, when the continuation is not a pair at all, `step` cannot continue with anything and π is returned as the final value.

It is crucial that all primitive functions invoke `step`, for it is the heart of the evaluator, processing the continuation. For example, the primitive unary function `cdr`, which takes the second element of a pair, is specified as follows.

$$\text{cdr}(p)(\kappa, \sigma) = \begin{cases} \text{step}(d)(\kappa, \sigma) & \text{if } \sigma[p \rightarrow a:d] \\ \text{step}(\perp)(\kappa, \sigma) & \text{otherwise} \end{cases} \quad (10.26)$$

Henceforth we will not explicitly write the “`step`(\perp)(κ, σ) otherwise” part anymore. As a second example, consider addition, which is a primitive binary function using the underlying implementation’s addition operator.

$$+(n, n')(\kappa, \sigma) = \text{step}(n'')(\kappa, \sigma[n'' \stackrel{*}{\leftarrow} (m \stackrel{64}{+} m')]) \quad \text{if } n, n' \in \mathbb{N}_\sigma, \sigma[n \rightarrow m] \text{ and } \sigma[n' \rightarrow m'] \quad (10.27)$$

Notice that, like every expression, n and n' are mere indices of the storage, so first their values (m and m' , respectively) have to be retrieved. Then a fresh index n'' is allocated and the sum of m and m' is stored there. Here $\stackrel{64}{+}$ may be the 64-bit integer addition operator of the underlying implementation’s language.

Now we turn to the actual evaluation function: `eval`. It is a unary function taking a *closure*, denoted here using the letter θ . A closure is a pair whose head is a lexical environment and whose tail is an (unevaluated) expression. Given these facts we can immediately answer the question of how to start the evaluation of a Λ_3 program π stored in storage σ : initialize $\sigma' = \sigma[\theta \stackrel{*}{\leftarrow} \emptyset; \pi][\kappa \stackrel{*}{\leftarrow} \text{eval}:\emptyset]$ and determine the value of `step`(θ)(κ, σ'). So we form a pair (closure) of the lexical environment (which is initially empty: \emptyset) and the program π to be evaluated in that environment. What `step`(θ)(κ, σ') will do is go to the first case in (10.25), namely to call the primitive unary function `eval` with θ as argument. Then `eval` must distinguish three cases because a program can either be a primitive constant (i.e., \perp , \emptyset , or a function), a number, or a pair. Constants are self-evaluating, numbers are treated as indices of the lexical environment, and pairs are treated as applications. `eval` is then specified as follows.

$$\text{eval}(\theta)(\kappa, \sigma) = \begin{cases} \text{step}(\pi)(\kappa, \sigma) & \text{if } \sigma[\theta \rightarrow \rho:\pi] \text{ and } \pi \in \mathbb{P} \\ \text{step}(\rho \downarrow_\sigma n)(\kappa, \sigma) & \text{if } \sigma[\theta \rightarrow \rho:\pi] \text{ and } \pi \in \mathbb{N}_\sigma \text{ and } \sigma[\pi \rightarrow n] \\ \text{step}(\theta_1)(\kappa', \sigma') & \text{if } \sigma[\theta \rightarrow \rho:\pi:\pi'] \end{cases} \quad (10.28)$$

where $\sigma' = \sigma[\theta_1 \stackrel{*}{\leftarrow} \rho:\pi][\theta_2 \stackrel{*}{\leftarrow} \rho:\pi'][\kappa' \stackrel{*}{\leftarrow} \text{eval}:(\text{next}:\theta_2):\kappa]$. In the first case the constant is simply extracted from the closure (θ). In the second case, the De Bruijn index inside the

closure is used as an index of the lexical environment ρ , also contained in θ . (Definitions of \mathbb{P} , \mathbb{N} , and \downarrow are provided in the appendix). In the third case, `eval` makes two closures: θ_1 is the operator, to be evaluated immediately; θ_2 is the operand, to be evaluated “next.” Both closures get the same lexical environment (ρ). The new continuation κ' reflects the order of evaluation of first π then π' : `eval` itself is placed at the top of the continuation, followed by the primitive binary function `next` with its first argument (θ_2) already filled in. The second argument of `next` will become the value of π , i.e., the operator of the application. `next` then simply has to save the operator on the continuation and evaluate the operand, which is its first argument (a closure). This behavior is specified in the *second* case below.

$$\text{next}(\theta, \varphi)(\kappa, \sigma) = \begin{cases} \text{step}(\theta)(\kappa', \sigma[\kappa' \stackrel{*}{\leftarrow} \varphi':\kappa]) & \text{if } \sigma[\varphi \rightarrow \text{quote2}:\varphi'] \\ \text{step}(\theta)(\kappa', \sigma[\kappa' \stackrel{*}{\leftarrow} \text{eval}:\varphi:\kappa]) & \text{otherwise} \end{cases} \quad (10.29)$$

What does the first case do? It handles a generalized form of quoting. The reason that `quote`, `lambda`, `set!`, and `if` are special forms in Scheme (as we have seen in Λ_2) is because their arguments are not to be evaluated immediately. This common theme can be handled by just one mechanism. We introduce the binary primitive function `quote2` and add a hook to `next` (the first case above) to prevent `quote2`'s second argument from being evaluated. Instead, the closure (θ , which represented the unevaluated second argument of `quote2`) is supplied to the first argument of `quote2` (which must therefore be a function).⁹

So where in Scheme one would write `(quote (1 2))`, here we have to write `((quote2 cdr) (1 2))`. Since `quote2` supplies a closure (an environment–expression pair) to its first argument, `cdr` can be used to select the (unevaluated) expression from that closure. Also, we have to quote numbers explicitly, otherwise they are taken to be De Bruijn indices. For example, Scheme's `(+ 1 2)` here becomes `((+ ((quote2 cdr) 1)) ((quote2 cdr) 2))`. This may all seem cumbersome, but it should be kept in mind that the Scheme representation can easily be converted automatically, so a programmer does not have to notice any difference and can just continue programming using Scheme syntax.

What about `lambda` abstractions? For that we have the primitive binary function `lambda2`, which also needs the help of `quote2` in order to prevent its argument from being evaluated prematurely. For example, where in Scheme the λ -calculus expression $\lambda x.\lambda y.x$ is written as `(lambda (x y) x)`, here we have to write `((quote2 lambda2) ((quote2 lambda2) 1))`. `lambda2` is specified as follows

⁹`quote2` may also be seen as a kind of *fexpr* builder.

(cf. (10.17)).

$$\text{lambda2}(\theta, \varepsilon)(\kappa, \sigma) = \text{step}(\theta')(\kappa', \sigma[\theta' \stackrel{*}{\leftarrow} (\varepsilon:\rho):\pi][\kappa' \stackrel{*}{\leftarrow} \text{eval}:\kappa]) \quad \text{if } \sigma[\theta \rightarrow \rho:\pi] \quad (10.30)$$

So `lambda2` takes a closure and a value, extends the lexical environment captured by the closure with the provided value, and then signals that it wants the new closure to be evaluated by pushing `eval` onto the continuation.

As we saw earlier, conditional expressions are ternary, taking a condition, a *then*-part, and an *else*-part. In the very simple system presented here, `if` is just a primitive unary function, which pops from the continuation both the *then*-part and the *else*-part, and sets up one of them for evaluation, depending on the value of the condition.

$$\text{if}(\varepsilon)(\kappa, \sigma) = \begin{cases} \text{step}(\theta)(\kappa'', \sigma[\kappa'' \stackrel{*}{\leftarrow} \text{eval}:\kappa']) & \text{if } \varepsilon \neq \perp \text{ and } \sigma[\kappa \rightarrow (\text{next}:\theta):\varphi:\kappa'] \\ \text{step}(\theta)(\kappa'', \sigma[\kappa'' \stackrel{*}{\leftarrow} \text{eval}:\kappa']) & \text{if } \varepsilon = \perp \text{ and } \sigma[\kappa \rightarrow \varphi:(\text{next}:\theta):\kappa'] \end{cases} \quad (10.31)$$

So any value other than \perp is interpreted as “true.” Where in Scheme one would write `(if c t f)`, here we simply write `((if c) t) f`.

An implementation of Scheme’s `set!` in terms of `quote2` can be found in the appendix.

Now we have seen that the lexical environment and the program currently under evaluation are easily obtained using `quote2`. What remains is reflection on the continuation. Both inspection and modification of the continuation can be achieved with just one very simple primitive: `swap-continuation`. It simply swaps the current continuation with its argument.

$$\text{swap-continuation}(\kappa')(\kappa, \sigma) = \text{step}(\kappa)(\kappa', \sigma) \quad (10.32)$$

The interested reader can find an implementation of Scheme’s `call-with-current-continuation` in terms of `swap-continuation` in the appendix.

This was the last of the core functions implementing the interpreter for Λ_3 . All that is missing is some more (trivial) primitive function like `cons`, `pair?`, `*`, `eq?`, and `function(s)` for communicating with the external environment.

10.6 Discussion

So is Λ_3 the language a self-reflective Gödel Machine can be programmed in? It certainly is a suitable core for one. It is simple and small to the extreme (so it is easy to reason about), yet it allows for full functional self-reflection. It is also important to note that we

have left nothing unspecified; it is exactly known how programs, functions, and numbers are represented structurally and in memory. Even the number of memory allocations that each primitive function performs is known in advance and predictable. This is important information for self-reasoning systems such as the Gödel Machine. In that sense Λ_3 's evaluator solves all the issues involved in self-reflection that we had uncovered while studying the pure λ -calculus-based tower of meta-circular evaluators.

We are currently using a Λ_3 -like language for our ongoing actual implementation of a Gödel Machine. The most important extension that we have made is that, instead of maintaining one continuation, we keep a stack of continuations, reminiscent of the poetically beautiful reflective tower with its summit lost in gray and cloudy skies—except this time with solid foundations and none of those misty marshes! Although space does not permit us to go into the details, this construction allows for an easy and efficient way to perform interleaved computations (including critical sections), as called for by the specification of the Gödel Machine's *scheduler*.

Appendix: Details of Notation Used

After parsing and before evaluation, a Λ_3 program has the following structure:

$$\begin{array}{ll}
 \#f, error \implies \perp & (\pi \pi') \implies \pi:\pi' \\
 \#t, '() \implies \emptyset & \text{lambda } \pi \implies (\text{quote2:lambda2}):\pi \\
 x \implies x \text{ for } x \in \mathbb{P} \cup \mathbb{F} \cup \mathbb{N} & \text{quote } \pi \implies (\text{quote2:cdr}):\pi
 \end{array}$$

For historical reasons [8], π is a typical variable denoting a program or expression (same thing), ρ is a typical variable denoting an environment, κ is a typical variable denoting a continuation (or 'call stack'), and σ is a typical variable denoting a storage (or working memory).

A storage $\sigma = \langle \mathbb{S}, \mathbb{N} \rangle$ consists of a fixed-size array \mathbb{S} of 64 bit chunks and a set \mathbb{N} of indices indicating at which positions numbers are stored. All other positions store pairs, i.e., two 32 bit indices to other positions in \mathbb{S} . To look up the value in a storage σ at location p , we use the notation $\sigma[p \rightarrow n]$ or $\sigma[p \rightarrow a:d]$. The former associates n with all 64 bits located at index p , whereas the latter associates a with the least significant 32 bits and d with the most significant 32 bits. We say that the value of p is the number n if $\sigma[p \rightarrow n]$ and $p \in \mathbb{N}_\sigma$ or the pair with head a and tail d if $\sigma[p \rightarrow a:d]$. Whenever we write $\sigma[p \rightarrow a:d]$, we tacitly assume that $p \notin \mathbb{N}_\sigma$.

A new pair is allocated in the storage using the notation $\sigma[p \stackrel{*}{\leftarrow} a:d]$, where a and d are locations already in use and p is a fresh variable pointing to a previously unused location in σ . Similarly, $\sigma[p \stackrel{*}{\leftarrow} n]$ is used to allocate the number n in storage σ , after which it can be referred to using the fresh variable p . Note that we do not specify where in \mathbb{S} new pairs and numbers are allocated, nor how and when unreachable ones are garbage collected. We only assume the existence of a function $free(\sigma)$ indicating how many free places are left (after garbage collection). If $free(\sigma) = 0$, allocation returns the error value \perp . Locating a primitive in a storage also returns \perp . These last two facts are formalized respectively as follows:

$$\sigma[\perp \stackrel{*}{\leftarrow} x] \text{ iff } free(\sigma) = 0 \quad (10.33)$$

$$\sigma[c \rightarrow \perp] \text{ for all } c \in \mathbb{P} \quad (10.34)$$

where the set of primitives \mathbb{P} is defined as follows:

$$\mathbb{P} = \mathbb{B} \cup \mathbb{F}, \quad \mathbb{B} = \{\perp, \emptyset\}, \quad \mathbb{F} = \mathbb{F}_1 \cup \mathbb{F}_2, \quad (10.35)$$

$$\mathbb{F}_1 = \{\text{step, eval, if, car, cdr, pair?, number?, swap-continuation}\}, \quad (10.36)$$

$$\mathbb{F}_2 = \{\text{next, lambda2, quote2, cons, set-car!, set-cdr!, eq?, =, +, -, *, /, >\} \quad (10.37)$$

These are the basic primitive functions. More can be added for speed reasons (for common operations) or introspection or other external data (such as amount of free memory in storage, current clock time, IO interaction, etc.). Note that `quote2` is just a dummy binary function, i.e., $\text{quote2}(\varphi, \varepsilon)(\kappa, \sigma) = \text{step}(\perp)(\kappa, \sigma)$.

For convenience we often shorten the allocation and looking up of nested pairs:

$$\sigma[p \stackrel{*}{\leftarrow} a:(ad:dd)] \stackrel{\text{def}}{=} \sigma[p' \stackrel{*}{\leftarrow} ad:dd][p \stackrel{*}{\leftarrow} a:p'] \quad (10.38)$$

$$\sigma[p \stackrel{*}{\leftarrow} (aa:da):d] \stackrel{\text{def}}{=} \sigma[p' \stackrel{*}{\leftarrow} aa:da][p \stackrel{*}{\leftarrow} p':d] \quad (10.39)$$

$$\sigma[p \rightarrow a:(ad:dd)] \stackrel{\text{def}}{=} (\sigma[p \rightarrow a:p'] \text{ and } \sigma[p' \rightarrow ad:dd]) \quad (10.40)$$

$$\sigma[p \rightarrow (aa:da):d] \stackrel{\text{def}}{=} (\sigma[p \rightarrow p':d] \text{ and } \sigma[p' \rightarrow aa:da]) \quad (10.41)$$

Note that the result of an allocation is always the modified storage, whereas the result of a lookup is true or false.

Taking the n th element of list ρ stored in σ is defined as follows.

$$\rho \downarrow_{\sigma} n = \begin{cases} \varepsilon & \text{if } n = 0 \text{ and } \sigma[\rho \rightarrow \varepsilon:p'] \\ \rho' \downarrow_{\sigma} (n-1) & \text{if } n > 0 \text{ and } \sigma[\rho \rightarrow \varepsilon:p'] \\ \perp & \text{otherwise} \end{cases} \quad (10.42)$$

Scheme's `set!` and `call-with-current-continuation` can be implemented in Λ_3 as follows. They are written in Scheme syntax instead of Λ_3 for clarity, but this is no problem because there is a simple procedure for automatically converting almost any Scheme program to the much simpler Λ_3 .

```

1 (define set!
2   (quote2 (lambda (env-n)
3             (lambda (x)
4               (set-car! (list-tail (car env-n) (cdr env-n)) x))))))
5
6 (define (call-with-current-continuation f)
7   (get-continuation
8     (lambda (k)
9       (set-continuation (cons f k) (set-continuation k))))))
9
10 (define (get-continuation f)
11   (swap-continuation (list f)))
12
13 (define (set-continuation k x)
14   (swap-continuation (cons (lambda (old_k) x) k)))

```

where the functions `cons`, `list`, `car`, `cdr`, `set-car!`, and `list-tail` work as in Scheme [11].

Bibliography

- [1] T. Schaul and J. Schmidhuber, Metalearning, *Scholarpedia*. **6**(5), 4650, (2010).
- [2] J. Schmidhuber. Gödel machines: Self-referential universal problem solvers making provably optimal self-improvements. Technical Report IDSIA-19-03, arXiv:cs.LO/0309048 v2, IDSIA, (2003).
- [3] J. Schmidhuber. Gödel machines: Fully self-referential optimal universal self-improvers. In eds. B. Goertzel and C. Pennachin, *Artificial General Intelligence*, pp. 199–226. Springer Verlag, (2006). Variant available as arXiv:cs.LO/0309048.
- [4] J. Schmidhuber. Completely self-referential optimal reinforcement learners. In eds. W. Duch, J. Kacprzyk, E. Oja, and S. Zadrozny, *Artificial Neural Networks: Biological Inspirations - ICANN 2005, LNCS 3697*, pp. 223–233. Springer-Verlag Berlin Heidelberg, (2005). Plenary talk.
- [5] J. Schmidhuber, Ultimate cognition à la Gödel, *Cognitive Computation*. **1**(2), 177–193, (2009).
- [6] K. Gödel, Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I, *Monatshfte für Mathematik und Physik*. **38**, 173–198, (1931).
- [7] L. P. Kaelbling, M. L. Littman, and A. W. Moore, Reinforcement learning: a survey, *Journal of AI research*. **4**, 237–285, (1996).
- [8] C. Queinnec, *Lisp in Small Pieces*. (Cambridge University Press, 1996).
- [9] B. R. Steunebrink and J. Schmidhuber. A family of Gödel Machine implementations. In *Proceedings of the 4th Conference on Artificial General Intelligence (AGI-11)*. Springer, (2011).
- [10] M. Hutter, The fastest and shortest algorithm for all well-defined problems, *International Journal of Foundations of Computer Science*. **13**(3), 431–443, (2002).

-
- [11] R. Kelsey, W. Clinger, J. Rees, and (eds.), Revised⁵ report on the algorithmic language Scheme, *Higher-Order and Symbolic Computation*. **11**(1) (August, 1998).
 - [12] H. Abelson, G. J. Sussman, and J. Sussman, *Structure and Interpretation of Computer Programs*. (MIT Press, 1996), second edition.
 - [13] S. Jefferson and D. P. Friedman, A simple reflective interpreter, *LISP and Symbolic Computation*. **9**(2-3), 181–202, (1996).
 - [14] B. C. Smith. Reflection and semantics in LISP. In *Principles of programming languages (POPL84)*, (1984).
 - [15] J. des Rivières and B. C. Smith. The implementation of procedurally reflective languages. In *1984 ACM Symposium on LISP and functional programming*, (1984).
 - [16] D. P. Friedman and M. Wand. Reification: Reflection without metaphysics. In *Proceedings of ACM Symposium on Lisp and Functional Programming*, (1984).
 - [17] M. Wand and D. P. Friedman. The mystery of the tower revealed: A non-reflective description of the reflective tower. In *Proceedings of ACM Symposium on Lisp and Functional Programming*, (1986).
 - [18] O. Danvy and K. Malmkjær. Intensions and extensions in a reflective tower. In *Lisp and Functional Programming (LFP'88)*, (1988).
 - [19] A. Bawden. Reification without evaluation. In *Proceedings of the 1988 ACM conference on LISP and functional programming*, (1988).
 - [20] N. de Bruijn, Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, *Indagationes Mathematicae*. **34**, 381–392, (1972).

Chapter 11

Artificial General Intelligence Begins with Recognition: Evaluating the Flexibility of Recognition

Tsvi Achler

Los Alamos National Labs, Los Alamos, USA

achler@gmail.com

Many types of supervised recognition algorithms have been developed over the past half-century. However, it remains difficult to compare their flexibility and ability to reason. Part of the difficulty is the need of a good definition of flexibility. This chapter is dedicated to defining and evaluating flexibility in recognition.

Artificial Intelligence and even more so Artificial General Intelligence are inseparable from the context of recognition. Recognition is an essential foundation on top of which virtually every function of intelligence is based e.g.: memory, logic, internal understanding, and reasoning. Many logic and reasoning questions can be directly answered by reasoning based on recognition information. Thus it is important to understand forms of flexible recognition structures in order to know how to store and reason based on flexible information.

The first section describes various methods that perform recognition. The second section proposes tests and metrics to evaluate flexibility, and the third provides an example of applying the tests to these methods.

11.1 Introduction

Although many recognition algorithms have been developed over the past half century, arguably the most prevalent method of classification is based on learned feedforward weights \mathbf{W} that solve the recognition relationship:

$$\vec{Y} = \mathbf{W}\vec{X} \text{ or } \vec{Y} = f(\mathbf{W}, \vec{X}) \quad (11.1)$$

Vector \vec{Y} represents the activity of a set of labeled nodes, called neurons or outputs in different literatures and individually written as $\vec{Y} = (Y_1, Y_2, Y_3, \dots, Y_H)^T$. They are considered supervised because the nodes can be labeled for example: Y_1 represents “dog”, Y_2 represents “cat”, and so on. Vector \vec{X} represents sensory nodes that sample the environment, or input

space to be recognized, and are composed of individual features $\vec{\mathbf{X}} = (X_1, X_2, X_3, \dots, X_N)^T$. The input features can be sensors that detect edges, lines, frequencies, kernel features, and so on. \mathbf{W} represents a matrix of weights or parameters that associates inputs and outputs.

Learning weights \mathbf{W} may require error propagation and comparison of inputs and outputs, but once \mathbf{W} is defined, recognition is a feedforward process. Thus the direction of information flow during recognition is feedforward: one-way from inputs to the outputs. Variations on this theme can be found within different algorithm optimizations, for example: Perceptrons (Rosenblatt, 1958), Neural Networks (NN) with nonlinearities introduced into calculation of $\vec{\mathbf{Y}}$ (Rumelhart and McClelland, 1986), and Support Vector Machines (SVM) with nonlinearities introduced into the inputs through the kernel trick (Vapnik, 1995). Although these algorithms vary in specifics such as nonlinearities determining the function f , they share the commonality in that recognition involves a feedforward transformation using \mathbf{W} .

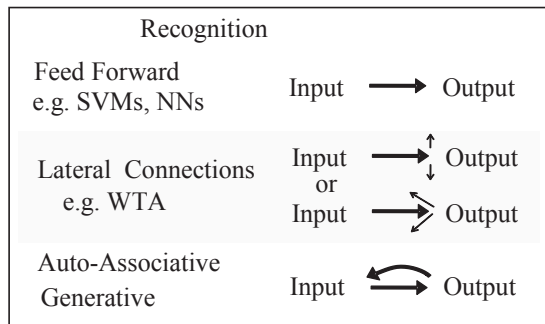


Fig. 11.1 Comparison of possible architectures used during recognition. In feedforward methods connections flow from inputs to outputs (top). After feedforward processing lateral connections may connect between outputs or back to inputs of a different node (middle). Generative or Auto-Associative connections are symmetrical and each node projects back to their own inputs (bottom).

Some recognition models implement lateral connections for competition between output nodes $\vec{\mathbf{Y}}$, such as: one-vs-all, winner-take-all, all-vs-all. However such competition methods rely on initially calculating $\vec{\mathbf{Y}}$ node activities based on the feedforward transformation \mathbf{W} .

A variation of feedforward algorithms are recurrent networks. Recurrent networks are feedforward networks in a hierarchy where a limited number of outputs are also used as inputs. This allows the processing of time. These networks can be unfolded into a recursive

feedforward network e.g. (Schmidhuber 1992; Williams & Zipser 1994; Boden 2006). Thus they also fall into a feedforward category.

In auto-associative networks, e.g. (Hopfield, 1982), all outputs feed back to their own inputs. When part of an input pattern is given, the network completes the whole pattern.

Generative models are a variation of auto-associative networks. The difference between generative and auto-associative networks is that in generative models, the auto-associative patterns are compared to, or subtracted from, the inputs. I focus on supervised generative models because they can have the same fixed points or solutions as feedforward models. Thus supervised generative and feedforward models can be directly compared.

Mathematically, generative models can be described by taking the inverse of equation (11.1):

$$\mathbf{W}^{-1} \vec{\mathbf{Y}} = \vec{\mathbf{X}} \quad (11.2)$$

Let's define \mathbf{M} as the inverse or pseudoinverse of \mathbf{W} . The relation becomes:

$$\mathbf{M} \vec{\mathbf{Y}} - \vec{\mathbf{X}} = \mathbf{0} \quad (11.3)$$

Using this equation can be called a generative process because it reconstructs the input based on what the network has previously learned. The term $\mathbf{M} \vec{\mathbf{Y}}$ is an internal prototype that best matches $\vec{\mathbf{X}}$ constructed using previously learned information. The values of $\vec{\mathbf{Y}}$ are the solution to the system. The fixed-points or solutions of equations (11.3) and (11.1) are identical, so the same $\vec{\mathbf{Y}}$ values also match the feedforward equation $\vec{\mathbf{Y}} = \mathbf{W} \vec{\mathbf{X}}$.

Equation (11.3) describes the solution but does not provide a way to project input information to the outputs. Thus dynamical networks are used that converge to equation (11.3).

One method is based on Least Squares which minimizes the energy function: $\mathbf{E} = \frac{1}{2} \|\vec{\mathbf{X}} - \mathbf{M} \vec{\mathbf{Y}}\|^2$. Taking the derivative relative to $\vec{\mathbf{Y}}$ the dynamic equation is:

$$\frac{d\vec{\mathbf{Y}}}{dt} = \mathbf{M}^T (\mathbf{M} \vec{\mathbf{Y}} - \vec{\mathbf{X}}) \quad (11.4)$$

This equation can be iterated until $d\vec{\mathbf{Y}}/dt = 0$ resulting in the fixed point solution that is equivalent to $\vec{\mathbf{Y}} = \mathbf{W} \vec{\mathbf{X}}$. $\mathbf{M} \vec{\mathbf{Y}}$ represents top-down feedback connections that convert $\vec{\mathbf{Y}}$ to $\vec{\mathbf{X}}$ domain. $\mathbf{M}^T \vec{\mathbf{X}}$ represents feedforward connections that convert $\vec{\mathbf{X}}$ to $\vec{\mathbf{Y}}$ domains. Both feedforward and feedback connections are determined by \mathbf{M} and together emulate feedforward weights \mathbf{W} .

Another way to converge to equation (11.3) is using Regulatory Feedback (RF). The equation can be written as:

$$\frac{d\vec{\mathbf{Y}}}{dt} = \vec{\mathbf{Y}} \left(\frac{1}{V} \mathbf{M}^T \left(\frac{\vec{\mathbf{X}}}{\mathbf{M} \vec{\mathbf{Y}}} \right) - 1 \right) \quad \text{where } V = \sum_{j=1}^N M_{ji} \quad (11.5)$$

Alternatively, using expanded notation it can be written as:

$$\frac{dY_i}{dt} = \frac{Y_i}{\sum_{j=1}^N M_{ji}} \sum_{k=1}^N M_{ki} \left(\frac{X_k}{\sum_{h=1}^H M_{kh} Y_h} \right) - Y_i \quad (11.6)$$

assuming $M_{N \times H}$ dimensions for \mathbf{M} . Both generative-type models have the identical fixed points (Achler & Bettencourt, 2011).

Generative models, have roots in Independent Component Analysis (ICA), are commonly unsupervised and determine \mathbf{M} so that sparseness is maximized e.g. (Olshausen & Field, 1996). The unsupervised paradigm does not have supervised labels, so ultimately a supervised feedforward method is still used for classification e.g. (Zieler *et al.*, 2010). Since the goal is to compare alternate structures that perform supervised recognition and \mathbf{W} is supervised, then \mathbf{M} must be supervised and sparseness is not implemented.

Some supervised generative models use a restricted Boltzmann machine to help training which is limited to binary activation e.g. (Hinton & Salakhutdinov, 2006). However in effect, the testing configuration remains feedforward. The difference between the approach taken here and other approaches is that it is assumed that \mathbf{M} is already learned. \mathbf{M} remains fixed throughout the calculations. \mathbf{M} is easier to learn than \mathbf{W} because it represents fixed-points (Achler 2012, in press).

Thus supervised generative models using equations (11.5), (11.6) are evaluated using a fixed \mathbf{M} during testing. Equation (11.4) can be used as well but may require an additional parameter to converge. In-depth comparisons between supervised generative models, including differences between equations (11.4) and (11.5), are beyond the scope of this chapter and will be addressed in future work.

11.2 Evaluating Flexibility

As with intelligence, evaluating robustness in recognition algorithms is not straight forward. It is also not clear what is the best metric. For example, an algorithm that is designed for a specific task and performs superbly on that task is not necessarily robust. An algorithm that shows fair performance in multiple domains may be more robust.

In order to move the field forward, particularly with the introduction of completely new methods, it is necessary to objectively evaluate and compare algorithm performance. This requires creating a benchmark. Cognitive psychologists have struggled with such metrics to

evaluate human performance for almost a century and came up with IQ metrics. A similar philosophy has is suggested for AI, e.g. (Legg & Hutter, 2007).

However testing for robustness in a field such as AI poses a catch-22. Once a test is defined, it becomes a benchmark. Benchmarks commonly generate a competition for narrow algorithms that focus on the benchmark, as is the case with tests in the AI and recognition fields (e.g. Caltech-256 Object Category Dataset). Subsequently the algorithms that may perform well on benchmarks are not necessary the most flexible. The proposed workaround of the catch-22 is to focus on combinatorial problems combined with measurements of resources.

The proposed tests for robustness should not reward algorithms that are over-trained for the test. Thus the majority of the tests within the battery are designed to present a combinatorial explosion for a brute-force method that tries to achieve good performance only by learning specific instances of the testing suite. However multiple over-trained algorithms may be implemented as one algorithm to overcome the multiple tests. This is why the evaluation of resources is essential. The measurement of parameters, training, and setup costs of algorithms serves as a measure of “narrowness”.

The proposed evaluation compares performance on the test set with the total number of parameters and training required. This is an Occam’s razor-like metric rewarding the simplest solution with the most functionality.

The test battery can be updated periodically (e.g. every several years). The overall goals of evaluation are to: 1) Design tests where performance cannot be improved by learning instances from the test suite. 2) Provide a framework where performance can be compared across algorithms. 3) Re-evaluate algorithm performance and update the tests periodically to ensure flexibility and promote progress.

11.2.1 *The Testing Paradigm*

For testing the contending recognition algorithms are treated as a black box and given the same patterns. Let’s define supervised input-label patterns $A \rightarrow Z$. Thus if \vec{X}_A is presented to the network, then it is expected to recognize it and Y_A should go to one. Each network to be tested is given the same input to label associations for training (A, B, C, \dots, Z) . The actual patterns can be random patterns.

Testing is conducted by presenting patterns and pattern manipulations to the input layer \vec{X} and evaluating based on expectations the output layer \vec{Y} responses. The performance is pooled across tests and the number of parameters counted in the evaluation.

The majority of tests of classifiers in the literature do not test beyond the single \vec{X} e.g. ($\vec{X}_{\text{test}} = \vec{X}_A, \vec{X}_B, \dots, \vec{X}_Z$). Thus robustness in recognizing mixtures is not explicitly evaluated in the literature. In this work both single \vec{X} and the sensitivity to the manipulation of mixtures of \vec{X} are evaluated e.g. ($\vec{X}_{\text{test}} = \vec{X}_A + \vec{X}_B, \vec{X}_A \cup \vec{X}_B \dots$).

Three types of problems are drawn upon to produce combinatorial explosions: superposition catastrophe (problems with mixtures of patterns), the binding problem (problems of grouping components of patterns), and numerosity (estimating the number patterns without individually counting each one). Unless algorithms have a powerful method of generalizing what they have learned, the training required can increase combinatorially with network size.

11.2.2 *Combinatorial Difficulties of Superposition or Mixes*

A simple way to create a combinatorial difficulty is with mixtures of patterns. For example if a network can process 5 000 patterns, there are about 12 million possible two pattern combinations of those patterns, 20 billion possible three pattern combinations, and so on.

In methods that over-rely on learning, the mixtures must be learned. This can quickly overcome the number of available variables. We refer to this property as a ‘‘combinatorial’’ explosion regardless if it is technically exponential or another function.

The mixture test evaluates algorithm performance when features of patterns are mixed or added together forming a ‘superposition’ (von der Malsburg, 1999; Rosenblatt, 1962).

In the superposition test, the networks are tested on patterns formed from mixtures of input vectors, for example $\vec{X}_{\text{mix}} = \vec{X}_A + \vec{X}_B$, $\vec{X}_{\text{mix}} = \vec{X}_A + \vec{X}_B + \vec{X}_C$ and so on. Let k represent the number of patterns mixtures superimposed in \vec{X}_{mix} . Let n represent the number of individual patterns the network knows, then the possible number of combinations increases exponentially and is given by:

$$\text{number_of_combinations} = \binom{n}{k} = \frac{n!}{k!(n-k)!} \quad (11.7)$$

During testing, for each network the top k y -values are selected and their identities are compared to the patterns in \vec{X}_{mix} . If the top k nodes match the patterns that were used to compose \vec{X}_{mix} , then a correct classification for that combination is recorded. This is repeated for all possible combinations.

As the number of simultaneous patterns increases the number of possible combinations increases combinatorially. When the networks are presented with 26 patterns, and one

is chosen, $k = 1$ (i.e. $\vec{\mathbf{X}}_{\text{test}} = \vec{\mathbf{X}}_A, \vec{\mathbf{X}}_B, \vec{\mathbf{X}}_C \dots$), there are 26 possible patterns. When networks are presented with two patterns simultaneously $k = 2$, there are 325 possible combinations of non-repeating patterns (e.g. $\vec{\mathbf{X}}_{\text{mix}} = \vec{\mathbf{X}}_A + \vec{\mathbf{X}}_B, \vec{\mathbf{X}}_A + \vec{\mathbf{X}}_C, \vec{\mathbf{X}}_A + \vec{\mathbf{X}}_D, \dots$). When networks are presented with three pattern combinations, $k = 3$ (e.g. $\vec{\mathbf{X}}_{\text{mix}} = \vec{\mathbf{X}}_A + \vec{\mathbf{X}}_B + \vec{\mathbf{X}}_C, \vec{\mathbf{X}}_A + \vec{\mathbf{X}}_B + \vec{\mathbf{X}}_D, \vec{\mathbf{X}}_A + \vec{\mathbf{X}}_B + \vec{\mathbf{X}}_E \dots$) there are 2,600 possible combinations. For eight simultaneous patterns (i.e. $k = 8$), there are 1,562,275 combinations. If networks cannot generalize for superpositions, they must train for most of these combinations, i.e. most of the 1.5 million combinations. Learning $k = 8$ does not guarantee good performance with other k values. Thus potentially the networks must be trained on all k 's.

In this test the superpositions are combined “noiselessly”. If two patterns say $\vec{\mathbf{X}}_A$ and $\vec{\mathbf{X}}_B$ have the same feature X_1 and each pattern has $X_1 = 1$, then if both patterns are superimposed (pattern1 + pattern2) the value of that feature is $X_1 = 2$ in the superposition. Next we evaluate combinations with information loss.

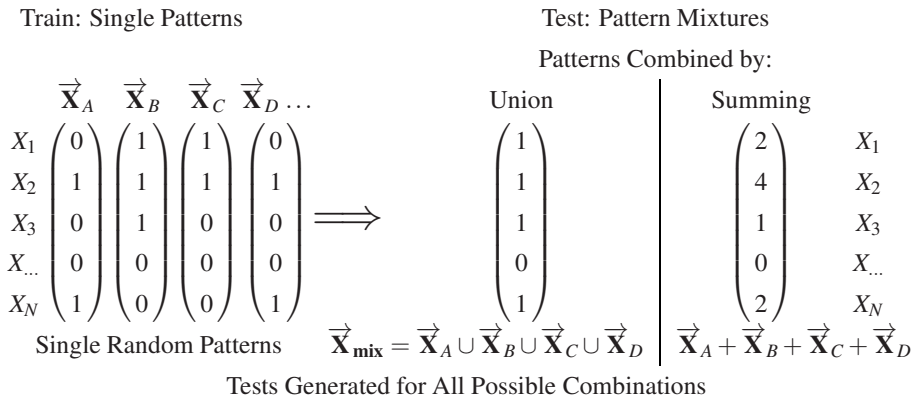


Fig. 11.2 Training with single patterns, testing with pattern mixtures. Test patterns are superpositions of single patterns combined by summing or union. Only 5 input features shown.

11.2.3 “Occluding” Superpositions

Flexible algorithms must function in multiple scenarios, even if fundamental assumptions of the scenarios change. For example, instead summing, overlapping features may “block” each other and their overlap may be better represented as a *union* or *min* of features.

In the union or min superposition test, the networks are tested on vectors that are combined using unions ($\vec{X}_A \cup \vec{X}_B$, $\vec{X}_A \cup \vec{X}_B \cup \vec{X}_C$, etc.). Not only do the same combinatorial problems in training occur with this scenario, a method suitable for superposition using addition may not apply to union superposition. The symbol \cup will be used to represent a *union*. Note that if values within the matrixes are non-binary then the union can be represented by the minimal value of the intersection, a *min* function. In this test the superpositions are combined with information loss. Suppose two patterns say \vec{X}_A and \vec{X}_B have the same feature X_1 and each pattern has $X_1 = 1$. If both patterns are superimposed ($\text{pattern1} \cup \text{pattern2}$) the value of that feature is $X_1 = 1$ in the superposition. Significant information can be lost in a union. For example, repeats cannot be decoded: $\vec{X}_{\text{mix}} = \vec{X}_A \cup \vec{X}_A$, will be identical to \vec{X}_A , $\vec{X}_A \cup \vec{X}_A \cup \vec{X}_A$ etc. However this is a useful test for the flexibility of algorithms. Since the comparison is between algorithms, any systematic information loss will affect the tested algorithms equally. Thus un-decodable aspects of test cases may reduce the number correct but will not affect overall ranking of algorithms since the algorithms are compared against each other. An example of the Union Superposition test is found in (Achler, Omar and Amir, 2008).

11.2.4 Counting Tests

The Superposition Catastrophe test has no more than one instance per pattern, without repeats. However, repeats (e.g. $\vec{X}_{\text{mix}} = \vec{X}_A + \vec{X}_A$) also provide important test cases. The ability to generalize without training specifically for the repeats is important for the ability to process or count simultaneous patterns without counting them one by one. Babies and animals have an inherent ability to estimate amounts or subitize, even while demonstrating an impoverished ability to individually count (Feigenson, Dehaene & Spelke, 2004). This ability may be important to quickly estimate the most abundant food resources and make decisions relevant to survival.

The counting test is designed to evaluate the ability to estimate amounts without individually counting. A network that can automatically process a superposition mixture such as $\vec{X}_{\text{mix}} = \vec{X}_A + \vec{X}_K + \vec{X}_J + \vec{X}_E + \vec{X}_J + \vec{X}_K + \vec{X}_K + \vec{X}_G$ can evaluate whether there are more K 's than J 's without individually counting the patterns. In the counting test, the amplitude value of y 's are evaluated. For the \vec{X}_{mix} above it is expected that \vec{Y} will show $Y_A = 1, Y_E = 1, Y_G = 1, Y_J = 2, Y_K = 3$, all other Y 's = 0. In the count test, instead of selecting the top k Y -values and comparing their identity to patterns in \vec{X}_{mix} , the amplitude value

of Y 's are compared to the number of repeats of the corresponding patterns within \vec{X}_{mix} .

$$\text{number_of_combinations} = \frac{n^k}{k!} \quad (11.8)$$

The number of possibilities in the count test increase even more than the superposition test. An example of the superposition and counting test is found in (Achler, Vural & Amir, 2009).

11.2.5 Binding Tests

The binding problem represents another scenario with a combinatorial explosion. However this often has different meanings in neuroscience, cognitive science, and philosophy. Thus we briefly review some definitions and usage.

According to “the neural binding” hypothesis, neurons within different neuronal assemblies fire in synchrony to bind different features of neuronal representations together (Gray *et al.*, 1989; von der Malsburg, 1999). This defines binding as a mechanism of attention and represents the internal workings of a specific mechanism. This type of binding is not within the scope of this paper since algorithms are treated as black boxes.

Another definition of binding is a “unity of perception”. This idea is somewhat metaphysical since it is hard to objectively define and measure unity from a human’s report. However there is a rich literature on errors of perception. Binding problems occur in humans when image features can be interpreted through more than one representation. An intuitive way to describe this problem is through visual illusions. For example, a local feature can support different representations based on the overall interpretation of the picture. In the old woman/young woman illusion of Fig 11.3, the young woman’s cheek





Fig. 11.3

is the old woman’s nose. Though the features are exactly the same, the interpretation is different. In humans, this figure forms an illusion because all features in the image can fit into two representations. Classifiers have similar difficulties but with simpler patterns. If a pattern can be part of two representations then the networks must determine to which it belongs. Training is used to find optimal interconnection weights for each possible scenario. However, this is not trivial for combinations of patterns and training can grow exponentially (Rosenblatt, 1962). We refine this definition of binding using simple tests and suggest it fits in a more general mathematical framework of *set-cover*.

The most basic binding scenario is given in figure 11.4D. Suppose a larger pattern completely overlaps with a smaller pattern: activation of node Y_1 is determined by pattern $\vec{X}_1 = (1, -)$, where feature $X_1 = 1$ and feature X_2 is not relevant for node Y_1 . Activation of node Y_2 is determined by pattern $\vec{X}_2 = (1, 1)$, where features X_1 and $X_2 = 1$. When presented with the larger pattern, (features $X_1 \& X_2 = 1$) the representation of the smaller pattern should not predominate. The network should recognize Y_2 (settle on $Y_1 = 0, Y_2 = 1$). A correct classification is Y_1 when only $X_1 = 1$, but Y_2 when $X_1 \& X_2 = 1$. This satisfies the set-cover problem. The classifier should prefer the representation that covers the most inputs with the least amount of overlap in the inputs.

A more complex scenario occurs with the addition of Y_3 , see figure 11.4E. Node Y_3 is determined by pattern $\vec{X}_3 = (-, 1, 1)$, where features X_2 and $X_3 = 1$ (and X_1 is not relevant for Y_3). Set-cover still holds. Since the same basic overlap exists between Y_1 and Y_2 , the same interaction should remain given X_1 and X_2 . However if $X_1 \& X_2 \& X_3 = 1$, then activation of Y_1 and Y_3 simultaneously can completely cover these inputs. Any activation of Y_2 would be redundant because X_2 would be covered twice. Choosing Y_2 given $X_1 \& X_2 \& X_3 = 1$ is equivalent to choosing the irrelevant features for binding.

For intuition let's give nodes Y_1 , Y_2 , and Y_3 , representations of wheels, barbell, and chassis respectively. The inputs represent spatially invariant features where feature X_1 represents circles, X_3 represents the body shape and feature X_2 represents a horizontal bar. Y_1 represents wheels and thus when it is active, feature X_1 is interpreted as wheels. Y_2 represents a barbell  composed of a bar adjacent to two round weights (features X_1 and X_2). Note: even though Y_2 includes circles (feature X_1), the circles do not represent wheels (Y_1), they represent barbell weights. Thus if Y_2 is active feature X_1 is interpreted as part of the barbell. Y_3 represents a car body without wheels (features X_2 and X_3), where feature X_2 is interpreted as part of the chassis. Given an image of a car  with all features simultaneously (X_1 , X_2 and X_3), choosing the barbell (Y_2) is equivalent to a binding error within the wrong context in light of all of the inputs.

Most classifiers if not trained otherwise are as likely to choose barbell or car chassis (Achler 2009). In that case the complete picture is not analyzed in terms of the best fit given all of the information present. This training is not trivial and may represent a combinatorial problem.

A solution based on set-cover automatically classifies the components and subcomponents of this problem. If $X_1, X_2, X_3 = 1$, then Y_1 and Y_3 represents the most efficient solution. If $X_1, X_2 = 1$, then Y_2 represents the most efficient solution. Set-cover also re-

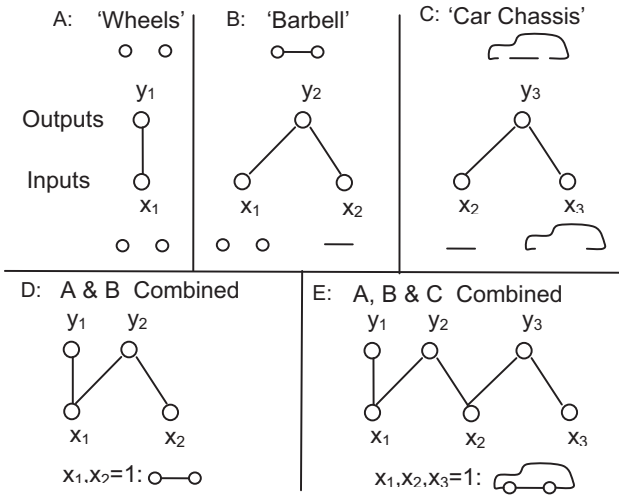


Fig. 11.4 (A–E): Modular combination of nodes Y_1 , Y_2 , Y_3 (A, B & C) display binding in combined networks (D and E). Y_1 & Y_3 represent car with wheels, Y_2 represents barbell. If $X_1, X_2 = 1$ then Y_2 should predominate (not Y_1), because it encompasses all active inputs. If $X_1, X_2, X_3 = 1$ then Y_2 should not predominate because interpreting a barbell within the car is a binding error.

solves basic recognition: if $X_1 = 1$, then Y_1 is the most efficient solution, if $X_2, X_3 = 1$, then Y_3 is the most efficient solution. Analysis of larger scenarios with infinite chains can be found in (Achler and Amir, 2008).

11.2.6 Binding and The Set-Cover Problem

The notion of set-cover can guide the design more complex binding tests. Given a universe of possible X 's of input patterns and Y 's that cover X 's. Set-cover asks what is the sub-set of Y 's that cover any arbitrary set of X but which use the fewest Y 's.

Set-cover can explain the old-young woman illusion, where the most encompassing representations mutually predominate. In that illusion there are two equal encompassing representations. Recognition settles on either the old woman or young woman, and no other combinations of features. All features are interpreted as part of either representation even if the interpretation of the individual feature can vary drastically (e.g. cheek of young woman vs. nose of old woman).

Set-cover is not a trivial problem to compute. The evaluation of set covering is NP-complete, but the optimization of set-cover is NP-hard. This means that training networks

to account for every possible cover is not practical because it may require every possibility to be trained, exposing another combinatorial explosion.

Thus networks that can “look beyond what is learned” should be able to resolve set-cover. Simple examples that require set-cover can be introduced to a test set. Examples of binding tests can be found in (Achler & Amir, 2008) and (Achler, 2009).

11.2.7 *Noise Tests*

Random noise is often used as a metric for flexibility. Random noise can be predicted and trained. Since training with noise does not necessarily result in a combinatorial explosion, this is less favored. However resistance to random noise is important and a commonly used measure. Thus it is added to the battery. Noisy stimuli can be generated using real-valued random exponential noise with mean amplitude μ , added to the prototypical stimulus. Noisy test vectors can be given to the classifiers and their result is compared to its original labels. The percent of patterns correctly identified for that noise level can be recorded. Other systematic non-random noise scenarios can be tested as well.

11.2.8 *Scoring the Tests*

The *Intelligence Quotient* (IQ) scoring system is borrowed from the field of psychology because it provides a familiar reference. Here each algorithm tested is like an individual. Performance on the battery of tests is pooled and the algorithms are ranked relative to the average of all algorithms. Performance is ranked on the Gaussian bell curve with a center value (average IQ) of 100, each standard deviation is 15 points. IQ value will be 100 for the average performing individuals (> 100 for better than average, < 100 for less than average), regardless of the actual score on a particular sub-test.

The absolute value of performance an algorithm achieves in a single test is not relevant since comparisons are the ultimate evaluation. For example, suppose one algorithm’s performance on a particular sub-test in the battery is low, say 10% correct. Since IQ values only reflect differences from average performance, if the average is 7% correct, that performance provides a better-than-average contribution to that algorithms IQ score for that test.

Another advantage of this method is that tests can be simply combined to give an aggregate for the battery. For example:

$$\text{Battery_IQ} = \text{Average}(\text{IQ}_{\text{numerosity}} + \text{IQ}_{\text{superposition}} + \text{IQ}_{\text{binding}} + \text{IQ}_{\text{noisy}} + \dots) \quad (11.9)$$

The IQ ranking system, evaluates algorithm flexibility while measuring and encouraging performance improvement. Additional tests are easy to include into the battery and scoring system.

11.2.9 *Evaluating Algorithms' Resources*

One advantage of AI testing over humans testing is the ability to evaluate the number of training trials and resources used by the algorithm. An algorithm with many parameters and extensive training may do slightly better than another algorithm that was not given as much training and does not have as many free parameters. However, a slightly better performance at the cost of more training and variables may actually be less flexible. The algorithm that performs best with the least amount of training and the least amount of free parameters, is the most desirable.

Thus we have included a measure of resources to the test score that accounts for the number of training epochs and free variables. These factors would modify the final test score to give a final AI_IQ score.

$$\text{AI_IQ} = \frac{\text{abilities}}{\text{resources}} = \frac{\text{Battery_Score}}{\text{training_epochs} + \text{variables} + \text{training}} \quad (11.10)$$

Flexible AI should require less degrees of freedom and apply to a greater number of scenarios (without extensive retraining for each scenario). The purpose of the metrics is to encourage such capabilities, limit degrees of freedom but maximize applicability. This philosophy of evaluation – rather than the exact details – is important for flexibility evaluation.

11.3 **Evaluation of Flexibility**

This section provides concrete examples of tests that compare the flexibility of feedforward methods to generative and other methods. This analysis is not complete but provides in-depth examples. We begin by defining random patterns outlined in Section 11.2.1: with 26 labels ($H = 26$) and 512 input features ($N = 512$). A SVM, Winner-take-all (WTA) algorithm, generative RF, and three different versions of NN algorithms are compared. The three versions are: naïve NN's trained only on single patterns, NNs trained on single and 2 pattern mixes, and NNs trained with WEKA.

The purpose of the publicly available *Waikato Environment for Knowledge Analysis* (WEKA) package (Witten & Frank, 2005) is to facilitate algorithm comparison and maintains up to date algorithms.

We begin by training a NN. The NNs are trained via backpropagation with momentum. 100,000 training examples are randomly chosen with replacement from the prototypes. The number of hidden units used is increased until it was sufficient to learn the input-output mapping to a component-wise tolerance of 0.05. Resulting NNs have on average 12 hidden units. The learning rate was .1, the momentum was 1 and there is a bias unit projecting to both hidden and output layers with value of 1.

Let's evaluate the number of parameters for NN: 26 output nodes, 512 input nodes, 12 hidden units, bias unit, momentum, component-wise tolerance, and learning rate. There are 551 variables so far. However during training unsuccessful NNs were created with 1-11 hidden units which were rejected due to suboptimal performance. There were also 100,000 training episodes for NN. All of those resources should be included in the evaluation. That is more than 1,200,000 episodes of training. To simplify our analysis lets suppose 100 511 represents the number of variables and resources.

After this, suppose performance on the mixtures is poor and to improve performance the network is trained on 2-pattern mixtures. That represents 325 more training patterns and more epochs. Those should be included as well.

There are 26 training episodes for RF. No training was done for combinations. There is one variable for tolerance determining when to end the simulation. This represents a total count of 565 for the number of resources: 26 outputs +512 inputs +1 tolerance variables +26 training episodes.

The winner-take-all algorithm is trained the same way as the RF algorithm however each output node has inhibitory connections to all other nodes. Since all of the output nodes are uniformly inhibitory, only one variable is added. This is a total of 566 resources. The WTA algorithm is evaluated for multiple patterns by 1) finding the most active node, 2) determining its identity, 3) inhibiting it, 4) finding the next most active node.

In the WEKA environment the number of variables and training episodes were not easily available. Thus it is assumed WEKA implemented algorithms are the most efficient possible, despite this being a generous oversimplification. SVMs do not have hidden units so the number output and input nodes should be the same as RF. However SVMs have kernels increasing the input space size and those are also governed by variables. A similar situation holds for the number of hidden variables for the WEKA neural network algorithm. Again, since WEKA is treated as a black box, these are not counted and the number of training episodes is not counted as well.

11.3.1 Superposition Tests with Information Loss

Superposition pattern mixtures were combined using a union (or min function). Some information is lost in the mixtures. Following (Achler, Omar, Amir, 2008) let's look at test performance. Two more networks that were not present in the original paper are included here: SVM and winner-take-all WTA. See table 11.1.

Table 11.1 (a) "Occluding" Superposition: evaluation of performance given mixtures with information loss. Raw scores of single patterns (left), 2-pattern mixtures, and 4-pattern mixtures (right).

number mixed:	k = 1	k = 2	k = 4
combinations:	26	325	14 950
RF	100%	100%	90%
NN naïve	100%	52%	4%
NN 2-pattern	100%	92%	32%
NN WEKA	100%	91%	28%
SVM WEKA	100%	91%	42%
WTA	100%	85%	24%

Table 11.1 (b) Total score for the 15301 occluding mixtures and analysis. Raw overall score (top row), followed by IQ values based on those scores, then the number of parameters and training episodes for each algorithm, followed by the AI_IQ scores. AI_IQ scores (bottom) are sensitive to the number of parameters. * indicates estimated number of parameters.

	RF	NN naïve	NN 2-pat	NN WEKA	SVM WEKA	WTA
Score (%)	90.2	5.2	33.4	29.5	43.1	25.4
IQ	128	83	98	96	103	93
Parameters	565	100,551	100,551	565*	565*	566
AI_IQ	126	86	86	99	105	97

RF performed well within all mixtures showing more flexibility. The number of parameters in WEKA are grossly underestimated, because it is being treated as a black box here. This demonstrates that an accurate count of parameters is critical for evaluating flexibility.

11.3.2 Superpositions without loss

Next superposition tests, where pattern mixtures are combined using an addition function without information loss, are evaluated following Achler (2009).

Table 11.2 (a) Performance for superposition mixtures without information loss. 26 single patterns (left), 2-pattern mixtures and 4-pattern mixtures (right) were tested. RF correctly recognized all patterns in all 325 $k = 2$ and 14950 $k = 4$ combinations.

number mixed: combinations:	k = 1	k = 2	k = 4
RF	100%	100%	100%
NN WEKA	100%	91%	4%
SVM WEKA	100%	94%	8%
WTA	100%	91%	42%

Table 11.2 (b) Total score for the 15301 superposition mixtures (top row), followed by IQ values based on those scores, then the number of parameters and training episodes for each algorithm. AI_IQ scores (bottom) are sensitive to the number of parameters. * indicates estimated number of parameters.

	RF	NN WEKA	SVM WEKA	WTA
Score (%)	100	6	10	43.1
IQ	121	88	90	101
parameters	565	565*	565*	566
AI_IQ	131	89	90	105

Comparing this test to the occluding test, winner-take-all performed better while *NN* and *SVM* performed worse. See table 11.2.

11.3.3 Counting Tests

This section is intended to evaluate counting tests where repeating patterns are combined in the mixtures using an addition function e.g. (Achler, Vural & Amir, 2009). Unfortunately WEKA could not test more than 15 000 tests. It is not designed for such large number of tests. This is an indication that the literature is not focusing on manipulations during recognition: primarily testing single patterns. Hopefully future versions of WEKA will allow more testing. Winner-take-all by its nature is not well suited for this test because it is not clear how to select a node several times.

The tests are based on the same 26 patterns used in the superposition. 17,576 possible combinations of 3 and 456,976 possible combinations of 4 pattern repeats were tested. RF scored well on all tests. Although this is an important part of the test battery, this test is not included in the overall score because of the limited comparisons available using WEKA.

11.3.4 *Binding Scenarios*

The “simplest” and “next simplest” binding scenarios require new patterns. In the “simplest” binding scenario there are two output nodes and two input nodes. Networks trained on the simplest scenario with two nodes perform correctly. NN WEKA, SVM WEKA, Winner-take-all, RF all score 100%. Again, the number of training epochs, hidden nodes and kernels parameters in WEKA are not evaluated since WEKA is considered a black box. Thus the performance of all methods are approximately equal.

In the “next simplest” binding scenario there are three inputs and three output nodes. Not all algorithms perform this correctly. In the case of all inputs active, RF decides on the correct two patterns that best fit the inputs. NN WEKA and SVM WEKA settle 50% on two representations that capture most of the inputs but do not cover all of the inputs in an efficient manner, a poor set-cover. The other input patterns were correctly matched.

Of the 4 possible of input patterns, RF achieved 4/4, SVM & NN WEKA and Winner-take-all achieved 3/4. Thus scores for this test are 100%, 75%, 75%, 75% respectively. IQ scores are: 123, 93, 93, 93 respectively. Resources are estimated as 3 inputs and 3 outputs for all methods but WTA which has an extra variable. AI_IQ scores based on those resource estimates are: 121, 96, 96, 86 respectively.

This is a simple demonstration of binding. Further tests are envisioned to be developed for more complex binding/set-cover scenarios.

11.3.5 *Noise Tests*

Tests were performed on the 26 patterns with random exponential noise of mean amplitude μ added to the prototypical stimulus. As part of tests NN were trained on patterns with $\mu = 0.25$ and $\mu = 0.15$, $NN \mu = 0.25$, and $NN \mu = 0.15$ respectively. Let’s look at test performance (from Achler, Omar, Amir, 2008).

The number of variables and training epochs for NN are similar to as Section 11.3.1 with 100,551 total. Again we do not apply variables and training for WEKA at this point. All NNs that were well-trained for noise performed better than RF. However RF was more noise resistant without training than NN naïve.

11.3.6 *Scoring Tests Together*

The proposed test suite includes: superposition, occluding superposition, counting, binding, and noise. However at this point the most complete results available compare

Table 11.3 (a) Comparison of performance in given random noise. NNs trained on noise improved performance. Naïve RF performs better than naïve NN, but NN WEKA performed best.

	Input Noise Level μ						
	0.1	0.15	0.2	0.25	0.3	0.35	0.4
RF	100%	100%	95%	85%	68%	47%	29%
NN naïve	100%	78%	43%	24%	12%	9%	8%
NN $\mu = 0.15$	100%	100%	100%	81%	53%	28%	17%
NN $\mu = 0.25$	100%	100%	100%	97%	79%	49%	28%
NN WEKA	100%	100%	100%	100%	98%	91%	76%

Table 11.3 (b) Total score for the 182 tests (top row), followed by IQ values based on those scores, then the number of parameters and training episodes for each algorithm. AI_IQ scores (bottom). * indicates estimated number of parameters.

	RF	NN naïve	NN $\mu = 0.15$	NN $\mu = 0.25$	NN WEKA
Score (%)	75	39	68	79	95
IQ	103	76	98	106	117
parameters	565	100,551	100,551	100,551	565*
AI_IQ	120	86	86	86	129

RF and NN WEKA in: superposition, occluding superposition, binding, and noise. The total score weighing all weighted tests equally is RF 91.3; NN WEKA 51.4*. AI_IQ score: RF 111; NN WEKA 89. Again, these numbers do not properly account for the number of parameters and training in WEKA. Ideally the all resources should be included including hidden layer variables and WEKA training. However the goal here is to indicate how flexibility can be evaluated and quantified. From this perspective this limited demonstration is sufficient. This score and demonstration shows how flexibility can be evaluated across a set of tests with combinatorial difficulties.

11.3.7 Conclusion from Tests

The test battery demonstrates fundamental differences in flexibility between a method using **M**: RF; and methods using **W**: NN, SVM. Despite the black box designation for WEKA, RF still performed better than NN and SVM. This is demonstrated whether the mixtures are summed together (Achler, 2009) or combined as a union (Achler, Omar, Amir, 2008). The mixtures can contain multiple additions of the same pattern (repeats) and the networks are able to determine the pattern mixture's numerosity (Achler, Vural,

Amir, 2009). They are also able to resolve certain binding scenarios: i.e. to determine whether subcomponent parts belong together (Achler & Amir, 2008). Thus recognition based on \mathbf{M} is quantified as more flexible.

Comparing performance between generative models, using the least-squares method, equation 11.4 versus 11.5, we obtain the same results except for the union cases and some instances of the binding cases. The instances of binding and unions that did not settle on the same solutions had input patterns outside the fixed points or linear combinations of the fixed points. In those cases both models converged, however differing results can be expected since no guarantees are made outside of the fixed points. The significance of the differences between methods is beyond the scope of this paper and will be discussed in future work.

11.4 Summary

It remains difficult to evaluate recognition flexibility using benchmark tests while discouraging methods which may improve performance by being narrowly optimized for the tests. We developed a set of metrics composed of tests and evaluation criteria that focus on flexibility.

The benchmark tests are designed to frustrate narrowly optimized methods by using tests that present combinatorial difficulties. The metrics are designed to frustrate narrowly optimized methods by penalizing performance based on the amount of resources used, such as parameters and the number of training instances.

We introduce several types of classifiers and showed how this evaluation system works. Most notably, we compared methods that utilize feedforward connections during testing with methods that use lateral competition and feedforward-feedback connections during testing.

We ran into several difficulties with the general simulation package, WEKA. It was not designed to test a large number of tests. However, despite such limitations the findings were sufficient to demonstrate how this metric system operates.

The conclusion of this study indicates that feedforward-feedback methods may be a more flexible configuration for classifiers. This is a topic that is still evolving and a work in progress.

Acknowledgments

This work was supported in part by the Notational Geospatial Agency and Los Alamos National Labs. I would like to thank Eyal Amir, Tanya Berger-Wolf, Luis Bettencourt, Garrett Kenyon, Peter Loxley, Cyrus Omar, Dervis Vural, and those who helped review this work for valuable comments.

Bibliography

- [1] Achler T. (2009). Using Non-Oscillatory Dynamics to Disambiguate Simultaneous Patterns. *IEEE IJCNN 2009*: 3570.
- [2] Achler T., in press 2012, Towards Bridging the Gap Between Pattern Recognition and Symbolic Representation Within Neural Networks, Workshop on Neural-Symbolic Learning and Reasoning AAAI.
- [3] Achler T. Amir E. (2008). Input Feedback Networks: Classification and Inference Based on Network Structure. *Artificial General Intelligence 1*: 15–26.
- [4] Achler T., Amir E., (2008). Hybrid Classification and Symbolic-Like Manipulation Using Self-Regulatory Feedback Networks, *Proc 4'th Int Neural-Symbolic Learning & Reasoning Workshop*.
- [5] Achler T., Bettencourt L., (2011). Evaluating the Contribution of Top-Down Feedback and Post-Learning Reconstruction, *Biologically Inspired Cognitive Architectures AAAI Proceedings*.
- [6] Achler T., Omar C., Amir E. (2008). Shedding Weights: More With Less. *IJCNN 2008*: 3020–3027.
- [7] Achler T. Vural D., Amir E. (2009). Counting Objects with Biologically Inspired Regulatory-Feedback Networks. *IEEE IJCNN 2009*: 36–40.
- [8] Boden M. (2006). A guide to recurrent neural networks and backpropagation. <http://www.itee.uq.edu.au/~mikaël/papers/rn_dallas.pdf>.
- [9] Feigenson L., Dehaene S., & Spelke E., (2004). Core systems of number. *Trends Cogn. Sci.* 8 (7): 307–14.
- [10] Gray C.M., Konig P., Engel A.K., Singer W. (1989). “Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties.” *Nature* 338 (6213): 334–7.
- [11] Hinton G.E., & Salakhutdinov R.R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313 (5786), 504–507
- [12] Hopfield J.J. (1982) Neural networks and physical systems with emergent collective computational abilities, *PNAS*, vol. 79 no. 8 pp. 2554–2558.
- [13] Hyvärinen A., Hurri J., Hoyer P.O. (2009). *Natural Image Statistics*, Springer-Verlag.
- [14] Legg S. and Hutter M. *Universal intelligence: A definition of machine intelligence*. *Minds & Machines*, 17 (4):391–444, 2007.
- [15] Olshausen B.A., Field D.J., (1997) Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research* 37:3311–3325.
- [16] Rosenblatt F. (1962). *Principles of neurodynamics; perceptrons and the theory of brain mechanisms*. Washington, Spartan Books.
- [17] Rumelhart D.E., & McClelland J.L. (1986). *Parallel distributed processing: explorations in the microstructure of cognition*. Cambridge, Mass.: MIT Press.

-
- [18] Schmidhuber J. (1992). Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4 (2):234–242.
 - [19] Vapnik V.N. (1995). *The nature of statistical learning theory*. New York: Springer.
 - [20] von der Malsburg C. (1999). The what and why of binding: the modeler's perspective. *Neuron*, 24 (1), 95-104, 111–125
 - [21] Williams R.J., Zipser D. (1994). Gradient-based learning algorithms for recurrent networks and their computational complexity. In *Back-propagation: Theory, Architectures & Applications*. Hillsdale, NJ: Erlbaum.
 - [22] Witten I.H., & Frank E. (2005). *Data mining: practical machine learning tools and techniques* (2nd ed.). Amsterdam; Boston, MA: Morgan Kaufman.
 - [23] Zeiler M.D., Kirshnan D., Taylor G.W., Fergus R. (2010). *Deconvolutional Networks*, *Computer Vision & Pattern Recognition CVPR*.

Chapter 12

Theory Blending as a Framework for Creativity in Systems for General Intelligence

Maricarmen Martinez, Tarek R. Besold, Ahmed Abdel-Fattah, Helmar Gust, Martin Schmidt, Ulf Krumnack, and Kai-Uwe Kühnberger

*Institute of Cognitive Science, University of Osnabrück,
Albrechtstr. 28, 49076 Osnabrück, Germany*

{*mmartine | tbesold | ahabdelfatta | hgust | martisch | krumnack | kkuehnbe*}@uos.de

Being creative is a central property of humans in solving problems, adapting to new states of affairs, applying successful strategies in previously unseen situations, or coming up with new conceptualizations. General intelligent systems should have the potential to realize such forms of creativity to a certain extent. We think that creativity and productivity issues can be best addressed by taking cognitive mechanisms into account, such as analogy-making, concept blending, computing generalizations and the like. In this chapter, we argue for the usage of such mechanisms for modeling creativity. We exemplify in detail the potential of such a mechanism like theory blending using a historical example from mathematics. Furthermore, we argue for the claim that modeling creativity by such mechanisms has a huge potential in a variety of domains.

12.1 Introduction

Artificial intelligence (AI) has shown remarkable success in many different application domains. Modern information technology, as most prominently exemplified by internet applications, control systems for machines, assistant systems for cars and planes, the generation of user profiles in business processes, automatic transactions in financial markets etc. would not be possible without the massive usage of AI technologies. In this sense, AI is a success story that triggered a significant impact to economic developments and changes in social relations and social networks. Nevertheless, there is a gap between the original goals of the founders of AI as a scientific discipline and current systems implementing state-of-the-art AI technologies. Whereas the original dream was to develop general-purpose

systems (like the famous general problem solver [19]), present AI systems are highly specialized, designed for a very particular domain, mostly without any generalization capabilities. Possible solutions for developing systems that approximate intelligence on a human scale are currently far from being achievable. This situation is unsatisfactory, at least if one does not want to give up the original motivation for the birth of AI as a discipline.

Artificial general intelligence (AGI) addresses precisely this gap between current AI systems and the obvious lack in providing solutions to the hard problem of modeling general intelligence, i.e. intelligence on a human scale. Higher cognitive abilities of humans fan out to the whole breadth of aspects that are usually examined in cognitive science. Humans are able to communicate in natural language, to understand and utter natural language sentences that they never heard or produced themselves, to solve previously unseen problems, to follow effectively strategies in order to achieve a certain goal, to learn very abstract theories (as in mathematics), to find solutions to open problems in such abstract disciplines etc. Although research has been examining frameworks for modeling such abilities, there are currently no good theories for computing creativity and productivity aspects of human cognition.

This chapter addresses the problem of how creativity and productivity issues can be modeled in artificial systems. Specifically, we propose to consider certain cognitive mechanisms as a good starting point for the development of intelligent systems. Such mechanisms are, for example, analogy-making, concept blending, and the computation of generalizations. In this text, we want to focus primarily on concept blending as an important cognitive mechanism for productivity.

The text has the following structure: Section 12.2 relates creativity abilities of natural agents to cognitive mechanisms that can be considered as a basis for explaining these abilities. Section 12.3 discusses cross-domain reasoning mechanisms that are important for the creative transfer of information from one domain to another domain. By such mechanisms it is possible to associate two domains that are originally independent and unconnected from each other. In section 12.4, some basic formal aspects of blending are introduced and section 12.5 models in detail the historically important creative invention of the complex plane in mathematics by blending processes. Section 12.6 gives an outlook for next generation general intelligent systems and section 12.7 concludes the chapter.

12.2 Productivity and Cognitive Mechanisms

Creativity occurs in many different contexts of human activity. It is not only a topic for artists or employees in the advertising industry, but also a topic in basic recognition processes, in the usage of natural language, in scientific disciplines, in solving problem tasks, in teaching situations, in developing new engineering solutions etc. The following list explains some examples in more detail.

- Besides the fact that metaphors in natural language are a strong tool to express propositions, feelings, warnings, questions etc. that are sometimes hard to express directly, they are moreover a possibility to establish connections between different, seemingly unconnected domains in order to facilitate learning. For example, if a high school teacher utters a sentence like *Gills are the lungs of fish* she expresses that lungs have the function in (let's say) mammals like gills have in fish. This is a remarkable interpretation for a sentence that is literally just semantic nonsense. In order to generate such an interpretation, the interpreter needs to be creative. An analogy-based framework for modeling metaphors can be found in [12].
- One of the classical domains for creativity of humans is the problem solving domain. Examples for problem solving are puzzles (like the tower of Hanoi problem), intelligence tests, scientific problem solving (like solving a problem in mathematics), or inventive problem solving in the business domain. To find solutions for a problem, humans show a remarkable degree of creativity. A classical reference for analogy-based problem solving is [13].
- Inventions in many domains are another prototypical example where creativity plays an important role. Besides the huge potential for business applications, also scientific disciplines are on a very basic level connected to inventions. For example, the human mind needs to be rather creative in order to come up with new concepts in rather abstract domains like mathematics. How is it possible that humans invent concepts like real numbers, complex numbers, Banach spaces, or operator semi-groups in Hilbert spaces? We will discuss the example of inventing the complex plane in mathematics in more detail below.
- The abstract concept of nesting an object in another object (quite often a similar one), is first of all a slightly abstract but simple idea. Usually it is referred to in design communities as the “nested doll principle” originating from the Russian Matryoshka doll, where dolls are contained in other dolls. The potential in building products out

of this simple idea is remarkable: planetary gearing, nesting tables and bowls, trojans (computer virus), self-similar fractals etc. are just some examples where this concept is used for designing products. In order to generate a conceptualization of such a general idea you need to transfer an abstract principle to completely unrelated domains, especially to such domains where it is not already obvious how this principle can be instantiated.

We claim that several forms of creativity in humans can be modeled by specific cognitive mechanisms. Examples of such mechanisms are analogy-making (transferring a conceptualization from one domain into another domain), concept blending (merging parts of conceptualizations of two domains into a new domain), computation of generalizations (abstractions), just to mention some of them. These mechanisms are obviously connected to other more standard cognitive mechanisms, like learning from sparse data or performing classical forms of inferences like deduction, induction, and abduction. We think that computational frameworks for the mentioned cognitive mechanisms are a way to implement creativity in machines. This may be considered as a necessary, although not a sufficient, step in order to achieve intelligence on a human scale, i.e. the ultimate goal of artificial general intelligence.

12.3 Cross-Domain Reasoning

In this text, we focus primarily on theory blending as a cognitive mechanism for modeling creativity. Nevertheless, from a more abstract perspective it is important to mention that blending, as well as analogy-making and the computation of generalizations are specific types of cross-domain reasoning. Establishing connections between seemingly unconnected domains is a highly important part of creativity as is obvious from the examples given in section 12.2 (compare for example the “nested doll principle” or metaphors in natural language). In this section, we introduce various cross-domain mechanisms, previously already specified in [17]. In each case, we illustrate the mechanism with an example taken from mathematics, without really going into the technical details (we provide the relevant references instead). We hope that by the end of this section the reader will have an appreciation of why we think mathematics is a good domain to exemplify and study creativity from a cognitive point of view: mathematical thinking spans a broad spectrum of domains, from the very concrete to the very abstract and requires the creative use of cognitive abilities that are not specific to mathematics. Section 12.5 will revisit some of the mechanisms

introduced here, in the context of a worked-out historical example of mathematical creativity.

All approaches to blending take knowledge to be organized in some form of domains, i.e. the underlying knowledge base provides concepts and facts in groups that can serve as input to the blending process. The different domains may in principle be incoherent and even mutually contradictory but they are nonetheless interconnected in a network organized by relations like generalization, analogy, similarity, projection, and instantiation. There are many examples in the literature on blending about conceptual integration in mathematics [1, 6, 15]. For example, [15] is a careful account of how complex mathematical notions are grounded on a rich network of more basic ones via metaphorical mappings and conceptual blends. The most basic linkages in this network consist of grounding metaphors through which mathematical notions acquire meaning in terms of everyday domains. For instance, basic arithmetic can be understood metaphorically in terms of four everyday domains: object collections, motion along a path, measuring with a unit rod, and Lego-like object constructions. Besides basic grounding metaphors there are linking metaphors and blends. These in turn can map domains that are not basic but either the target domains of metaphors or conceptual blends. The idea of the complex plane discussed below involves a blend of already non-basic algebraic, numerical, and geometric notions.

We use Heuristic-Driven Theory Projection (HDTP) as the underlying framework. Originally developed for metaphor and analogy-making (see [12] and [21] for details), HDTP has been applied to many different domains and has been extended in various directions, among other things, to cover also theory blending. In HDTP, domains are represented via finite first-order axiomatizations (defining domain theories) and intra-domain reasoning can be performed with classical logical calculi. Moreover, cross-domain reasoning can also be allowed, in many different ways.

- (1) *Analogy-making*: quite often identified as a central mechanism of cognition [7], the establishment of an analogical relation between two domains is based on identifying structural commonalities between them. Given an analogical relation, knowledge can be transferred between domains via analogical transfer. HDTP applies the syntactic mechanism of anti-unification [20] to find generalizations of formulas and to propose an analogical relation (cf. Figure 12.1). Several of the mechanisms listed next are actually supported by analogy-making and transfer.

Analogical transfer results in structure enrichment of the target side. In the HDTP framework such enrichment usually corresponds to the addition of new axioms to the

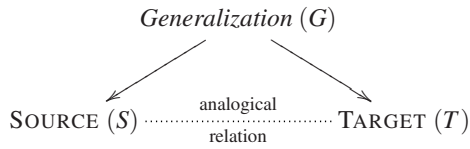


Fig. 12.1 Analogy via generalization in HDTP

target theory, but may also involve the addition of new first-order symbols. When using a sortal logic, even new sorts can be added to the language of the target theory.

There are cases in which analogical transfer is desired in order to create a new enriched domain, while keeping the original target domain unchanged. In such cases the generalization, source, target, and enriched domains are interconnected by a blend (see the next section for a depiction of how a blend interconnects four domains). An example of this kind of blending, and of structure enrichment involving new sorts, is the blend of (partial formalizations of) the domains MEASURING STICK (MS) (a metaphor that grounds the idea of discreteness) and MOTION ALONG A PATH (MAP) (grounding the idea of continuity) discussed in [10]. The blended domain is an expansion of MAP in which a notion of unit (taken from MS) is added. It can be thought of as a partial axiomatization of the positive real number line with markings for the positive integers. Without going into the formal details, in this example the blended domain (theory) comes from enriching the first-order theory by which MAP is represented with a subsort of “whole units”, and then importing the axioms of MS into it, in a form that relativizes them to the subsort.

- (2) *Cross-domain generalization*: As an example, an abstract approximation of naive arithmetic may be obtained by iterative pairwise generalizations of the basic grounding domains of Lakoff and Núñez mentioned above. These basic domains are analogous in various aspects that can then be generalized¹. The fact that HDTP always computes a generalization when it establishes an analogy between two domains was used in [11] to implement this example. Figure 12.2 shows that calculating an analogy between the first generalization and yet a third basic domain produces an even more generalized proto-arithmetic domain.
- (3) *Cross-domain specialization*: in the HDTP framework, after a generalized theory G has been obtained, the process of translating terms or properties from G to one of the

¹For example, joining object collections and putting together linear constructed Lego-objects are both commutative operations.

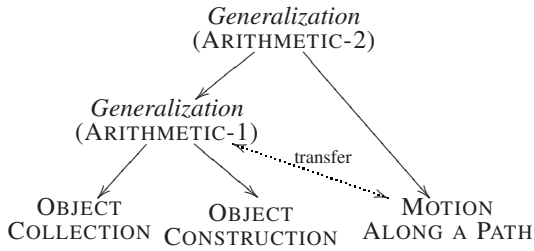


Fig. 12.2 Computing generalizations from Lakoff and Núñez' grounding metaphors for arithmetics

domains that it generalizes can be thought of as a process of cross-domain specialization. A key concern is to find conditions under which, say, the translation of any first-order proof done inside G (say Arithmetic-1 in Figure 12.2) is also a proof inside each of the domains G generalizes (say Object Collection in Figure 12.2). A discussion on such conditions, formulated in the theory of institutions, can be found in [14].

- (4) *Detection of congruence relations*: the detection of weak notions of equality (structural congruences) is pervasive in mathematics and beyond mathematics, as we will discuss. The issue is central even for the basic question of how humans can arrive at the notion of fractional numbers based only on basic grounding everyday domains like the four proposed in [15]. In [10], this challenge is approached using the fact that HDTP can in principle detect when some relation R in one of the input domains D_1 and D_2 is analogous to the equality relation in the other domain. This can then be used as a trigger for further processing, namely trying to establish if R is a congruence. If successful, the process should result in a diagram like Figure 12.3.

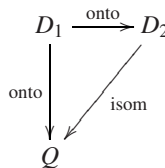


Fig. 12.3 Diagram of a quotient Q over a domain D_1 . Intuitively, at the level of models the elements of D_1 are mapped to their equivalence classes in the quotient Q .

While processes of quotient construction in the above sense may seem particular to mathematics, the facts show otherwise. Forming a quotient involves identifying a notion of

weak equality on a domain D_1 , deriving from it a useful way to categorize the entities of D_1 , and obtaining a theory of the space of categories. Each one of these aspects is important from the perspective of creativity in general cognitive systems. For example, it is a fact that through history humans have created many novel representation systems, of diverse kinds. One case in mathematics is the positive real numbers as a representation system for segments. The positive number line can be thought of as a quotient domain (Q) of the domain of physical segments (D_1) obtained thanks to the identification of segments of the same length (same image in the domain D_2 of positive reals) with the unique segment in the number line that has that length. By virtue of the isomorphism between the positive number line and the positive reals, one can also see the quotient Q as a geometric, more concrete model of the theory of the reals.

We conclude this section by noticing that the arrows in Figure 12.3 are labeled. The labels indicate *semantic* constraints on how the (unspecified but presupposed) universe of one domain can be mapped into another domain. The fact that we need the labels in Figure 12.3 to better approximate our intuition of what a quotient is, and the many examples we will see in our case study later, suggest that attaching semantic labels to the HDTP syntactic theory morphisms may turn out to be very useful. The semantic constraints of a theory morphism relating domain theories T_1 and T_2 may say things such as “the universe T_1 is about is embedded in the universe T_2 is about”. The complex plane example below will better illustrate what sorts of dynamics may be evoked by the kind of semantic labeling of morphisms we are proposing.

12.4 Basic Foundations of Theory Blending

To provide a formal basis for relating different domains to create new possible conceptualizations in a blending process we start with Goguen’s version of concept blending as given in [9], in particular in the form presented by his figure 3 (p. 6) here represented in Fig. 12.4.

The idea is that given two domain theories I_1 and I_2 representing two conceptualizations we look for a generalization G and then construct the blend space B in such a way as to preserve the correlations between I_1 and I_2 established by the generalization. The morphisms mapping the axioms of one theory to another are induced by signature morphisms mapping the symbols of the representation languages. Symbols in I_1 and I_2 coming from the same symbol in G correspond to each other in an analogical manner. Due to possible

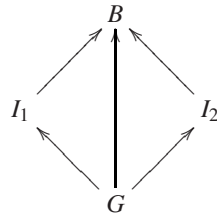


Fig. 12.4 The diagram of a blend.

incompatibilities between I_1 and I_2 the morphisms to B may be *partial*, in that not all the axioms from I_1 and I_2 are mapped to B . Goguen uses a generalized push-out construction² to specify the blend space B in such a way that the blend respects the relationship between I_1 and I_2 implicitly established by the generalization. That means that B is the smallest theory comprising as much as possible from I_1 and I_2 while reflecting the commonalities of I_1 and I_2 encoded in the generalization G .

A standard example discussed in [9] is that of the possible conceptual blends of the concepts HOUSE and BOAT into both BOATHOUSE and HOUSEBOAT (as well as other less obvious blends). In a very simplistic representation we can define HOUSE as

$$I_1 = \{ \text{HOUSE} \sqsubseteq \forall \text{LIVES-IN} . \text{RESIDENT} \}$$

and BOAT as

$$I_2 = \{ \text{BOAT} \sqsubseteq \forall \text{RIDES-ON} . \text{PASSENGER} \}.$$

Parts of the conceptual spaces of HOUSE and BOAT can be aligned, e.g. RESIDENT \leftrightarrow PASSENGER, LIVES-IN \leftrightarrow RIDES-ON, HOUSE \leftrightarrow BOAT (resulting in a conceptualization of HOUSEBOAT, while RESIDENT \leftrightarrow BOAT results in a conceptualization of BOATHOUSE). Conceptual blends are created by combining features from the two spaces, while respecting the constructed alignments between them. The blend spaces coexist with the original spaces: HOUSE and BOAT are kept untouched although there might be new relations between BOATHOUSE and HOUSE or BOAT. Hence, the blend space is not the result of extending one of the given spaces with new features or properties, but constructing a new one. Conceptual blending gives us a way to understand how theories that would simply be inconsistent if crudely combined, can nevertheless be blended by taking appropriately chosen parts of the theories, while respecting common features.

²For a formal definition of the category theoretic construction of a push-out compare for example [16].

12.5 The Complex Plane: A Challenging Historical Example

For a long time, the mathematical community considered the complex numbers to be only algebraic scaffolding, tools that carefully used *as if* they were numbers could facilitate calculations and discoveries about real numbers. One problem was that, while real numbers could be naturally conceived as distances from a reference point in a line, it was not clear at all whether a number of the form $a + bi$ (where a and b are real numbers and $i^2 = -1$) could correspond to any intuitive notion of magnitude. The discovery of the complex plane played a very important role in the acceptance of complex numbers as proper numbers. It turned out that the complex number $a + bi$ corresponds to an arrow (vector) starting from the origin of a cartesian system of coordinates and ending at the coordinate (a, b) , and the basic arithmetical operations on complex numbers such as sum and product have geometric meaning. The complex plane representation of complex numbers is a prime historical example of creativity in mathematics, one that involves devising a novel domain (the complex plane) through which a known one (the domain of complex numbers as pure algebraic entities) can be understood in geometrical terms. In this section, we reconstruct in our terms J. R. Argand's discovery of the complex plane in 1806, according to his own report [2]. Our aim is to illustrate a real case of discovery that involved the construction of a network of interrelated domains, obtained by either: (1) retrieval from memory, (2) analogical construction of a new domain, or (3) blending. The network of domains is shown in Figure 12.5. We will describe the way in which this graph is formed, presenting only minimal partial formalizations of the domains, and instead focusing on the motivations and constraints that guide the creative process in this case study. This section is partially based on ideas first presented in [17].

According to [22], Argand's discovery of the complex plane arose from thoughts not directly related to the complex numbers. His starting point was the observation that negative numbers may not seem "real" if you restrict yourself to magnitudes such as the size of object collections, but they do make sense when measuring weight in a two-plate scale. In such context, there is a natural reference point (an origin or "zero") and a displacement of a plate with respect to that point in one of two possible directions: upwards or downwards. In general, creative production is in numerous (maybe most) occasions a serendipitous process, and one would like human-level artificial intelligent systems to have the ability to be creative "on the spot", as the opportunity arises, not only in settings where a determinate goal has been set. However, it is fair to say that current artificial creative systems work by trying to achieve rather unconstrained but still pre-given goals (e.g. compose a narrative

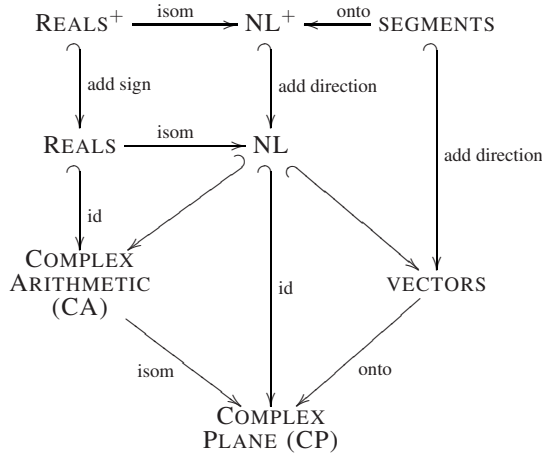


Fig. 12.5 Domains involved in Argand's reasoning. Arrows indicate the direction of (partial) theory morphisms. Arrow labels indicate constraints at the level of models. Curved tails are like an "injective" label on the arrow.

out a given set of old narratives, or compose a new musical piece in the style of certain composer). It is still a challenge to make them able, as Argand was, of posing interesting creative goals, desires, or questions out of casual observations.

From a network-of-domains viewpoint, Argand's observation starts forming the network in Figure 12.5 by recruiting the four domains in the upper left area. The number line representation (NL) is an already familiar domain, retrieved from memory, a geometric representation of the real numbers. Similarly, Argand remarks that the "right side" of the number line (NL⁺) is a geometric representation of the positive real numbers.

To clarify the nature of the arrows (morphisms) in the diagram, take the example of the arrow from NL⁺ to NL. There are two aspects to this arrow. Its direction indicates that there is a *partial* morphism³ from the first-order theory NL⁺ to the first-order theory NL. This accounts for the purely syntactic aspect of the arrow. There are also two semantic markers that together indicate how the intended *universes* of the domains relate to each other. Namely, the curved tail says that there is a total embedding from the *universe* of NL⁺ to the universe of NL that preserves the truth of the formulas mapped by the syntactic morphism, and the "add direction" marker is a construction marker (a program, maybe) that says how the embedding is actually constructed.

³This morphism is a partial function mapping a subtheory of the domain into the codomain.

As we will see, with the exception of the complex plane (CP) and the vector domain (VECTORS), which were Argand's creation, all other nodes in Figure 12.5 and the arrows connecting them were previously known to him. Hence, the corresponding subgraph is part of a long-term memory of interrelated domains. The diamond formed by the novel domains CP and VECTORS plus the complex arithmetic domain (CA) and the number line (NL) domain is a blend that crowns the whole process we will now describe.

Argand's positive number line NL^+ included notions of proportions (a:b::c:d) and geometric mean. The geometric mean of two segments a and b in NL^+ is the unique x such that a:x::x:b. It can be obtained algebraically as the solution of $x \cdot x = a \cdot b$, or via a geometric construction in SEGMENTS, the result of which (a segment) is mapped back to NL^+ . Argand uses the initial four-domains network to pose his first interesting question.

Argand's Challenge 1: Is there a notion of geometric mean in NL that extends that of NL^+ and has also natural algebraic and geometric corresponding notions?

To address the problem, Argand uses two consecutive analogical steps:

- (1) He detects the analogy between the "add sign" and "add direction" semantic markers: every real number is either positive or $-r$ for some positive real r , and each segment in the number line lies in the positive side of it or its (geometric) opposite. This highlights the domain of real numbers as a candidate for providing an algebraic interpretation of the extended notion of geometric mean.
- (2) Ensuring an analogy with the geometric mean in NL^+ , he proposes *as definition* that the magnitude of the geometric mean of two entities in NL is the geometric mean of their magnitudes and its direction is the geometric mean of their directions (the two possible directions are represented by the segments $+1$ and -1).

The heart of the initial challenge can now be stated in terms of the newly created notion of *geometric mean of directions*.

Argand's Challenge 1a: Find a segment x such that $1:x::x:-1$. The solution must have both algebraic and geometric readings, as does everything in NL.

Algebraically, moving to the isomorphic domain of reals, the problem is finding a real number x such that $x \cdot x = -1 \cdot 1 = -1$. No such x exists in the reals, but the domain of complex numbers is recruited as an extension of the reals where there is a solution⁴.

On the geometric side, by analogy with the "add direction" construction, Argand creates from the domain of SEGMENTS a new domain of directed segments, or VECTORS, with

⁴It is here that by serendipity, Argand ends up thinking about the complex numbers in a geometric context.

types
real, vector

constants
0, 1: *vector*
0, 1, 90, 180, 360: *real*

functions
| |, *angle: vector* → *real*
-: *vector* → *vector*
rotate, scale: vector × *real* → *vector*
+_v, *project: vector* × *vector* → *vector*
+, ·, +₃₆₀: *real* × *real* → *real*

predicates
proportion: vector × *vector* × *vector* × *vector*
geomean: vector × *vector* × *vector*
≡: *vector* × *vector*

laws
 $\forall \alpha \forall \beta \forall v \text{ (rotate(rotate}(v, 90), 90) = -v)$
 $\text{rotate(rotate}(\mathbf{1}, 90), 90) = -\mathbf{1}$
 $\forall \alpha \forall \beta \forall v \text{ (rotate(rotate}(v, \alpha), \beta) = \text{rotate}(v, \alpha +_{360} \beta))$
 $90 +_{360} 90 = 180$
 $\forall v \exists a \exists b \text{ (} v = \text{scale(rotate}(\mathbf{1}, a), b))$
 $\forall v \text{ (} v = \text{scale}(v, 1))$
 $\forall v_1 \forall v_2 \forall v_3 \forall v_4 \text{ (proportion}(v_1, v_2, v_3, v_4) \leftrightarrow \exists a \exists b \text{ (} v_2 = \text{rotate}(\text{scale}(v_1, a), b) \wedge v_4 = \text{rotate}(\text{scale}(v_3, a), b)))$
 $\forall v_1 \forall v_2 \forall v_3 \text{ (geomean}(v_1, v_2, v_3) \leftrightarrow \text{proportion}(v_1, v_3, v_3, v_2))$
(abelian group axioms for (+_v, -) and +₃₆₀)
(add also the missing vector space axioms for *scale* and +_v)
(congruence axioms for ≡)
(imported axioms from the reals for +, ·)

Fig. 12.6 Partial formalization of the VECTORS domain

a designated unit vector. Vectors can be stretched by a real factor, added, rotated, and projected. Figure 12.6 has a very partial and redundant axiomatization of this domain. The +₃₆₀ stands for real addition modulo 360 degrees, for adding angles. Argand's realization that the geometric mean of **1**, **-1** should be a unitary vector perpendicular to **1** corresponds to the fact that from our VECTORS axioms one can prove *geomean*(**1**, **-1**, *rotate*(**1**, 90)).

With this notion of geometric mean that makes sense algebraically and geometrically, Argand finally comes up with the key question about complex numbers.

Argand's Challenge 2: Can a geometric representation for the complex numbers be obtained from the vector plane?

In our terms, the problem is to find a quotient blend space CP with inputs CA and VECTORS and the additional constraint that NL must be embedded in it. To see how this could get started by using the HDTP framework, Figure 12.7 gives a partial axiomatization of the domain CA of complex numbers as objects of the form $a + bi$.

When comparing the axiomatizations of CA and VECTORS, the HDTP setup can detect that the first formulas in the domains are structurally the same. A generalization G will be created that, as shown in Figure 12.8, will include an axiom $\forall X(F(F(X, Y), Y) = -X)$ plus substitutions that lead back from G to the CA and VECTORS. These substitutions will map the term $F(X, Y)$ to the terms *rotate*($v, 90$) and $i * z$ respectively. Given the intended

types
real, complex (with *real* being a subsort of *complex*)

constants
i: *complex*
 0, 1: *real*

functions
re, im: *complex* → *real*
*+_c, **: *complex* × *complex* → *complex*
+, *:*: *real* × *real* → *real*
-: *complex* → *complex*

laws
 $\forall z (i * (i * z) = -z)$
 $i * i = -1$
 $\forall z \forall a \forall b (z = a + b \cdot i \leftrightarrow a = \text{re}(z) \wedge b = \text{im}(z))$
 $\forall z \forall z' (\text{re}(z +_c z') = \text{re}(z) + \text{re}(z'))$
 $\forall z \forall z' (\text{im}(z +_c z') = \text{im}(z) + \text{im}(z'))$
 $\forall z \exists a \exists b (z = a + b \cdot i)$
 $\forall z \forall z' (\text{re}(z * z') = \text{re}(z) \cdot \text{re}(z') - \text{im}(z) \cdot \text{im}(z'))$
 $\forall z \forall z' (\text{im}(z * z') = \text{re}(z) \cdot \text{im}(z') + \text{im}(z) \cdot \text{re}(z'))$
 (abelian group axioms for *+_c*)
 (add also the missing field axioms for *+_c* and ***)
 (imported axioms from the reals for *+*, *·*)

Fig. 12.7 Partial formalization of the CA domain

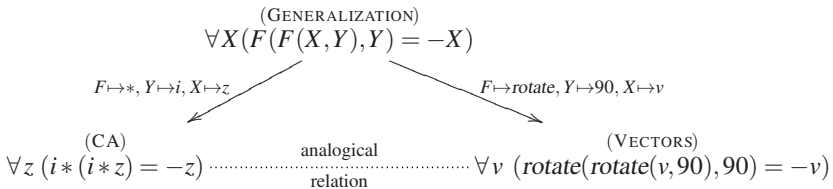


Fig. 12.8 Mapping from the generalization to CA and VECTORS.

meaning of our axiomatizations, this corresponds to detecting that rotating vectors by 90 degrees is analogous to multiplying by *i* in the complex arithmetic domain. This is the key observation on which Argand initiated his construction of the complex plane. The generalization *G* will include other things such as an operation *G* with axioms of Abelian group that generalizes *+_v* and *+_c*. We skip the full details here.

The example in this section shows the potential of the logic approach in modeling the role of very general cognitive mechanisms such as analogy-making, generalization, and blending in creativity. It illustrates ways in which our approach can suggest principled computational strategies for performing creative exploration and on the spot proposal of interesting creative tasks. Also, the strategies and issues shown here are general, relevant to many other subjects beyond mathematics. One example of this is the issue of *construction* of a context of relevant domains that may later facilitate the formation of blending domains. We saw that this can happen in at least two ways. First, such a context can provide the

motivation for finding a particular blend between two spaces. For instance, the motivation for the final blend in this section was a meta-analogy: can we get a geometric representation just as there is one in the network for the positive reals? Second, a context of relevant domains can suggest adequacy constraints to be imposed on a the blend to be created. In our blend example, two constraints went beyond the general diagram of a blend: the relevant part of NL must be embedded into the new blended domain CP, and CP must be the bottom domain of a quotient (i.e. play the role of Q in Figure 12.3).

12.6 Outlook for Next Generation General Intelligent Systems

The use of cognitively inspired techniques like conceptual blending or reasoning based on analogy and similarity in AI systems is (with few exceptions) to a large extent still in its very early infancy. In this section, continuing the overall argument from earlier sections, we will argue why accounting for an integration of these capacities when designing a general intelligent system might be a good and quite rewarding idea. Therefore, we will give an overview of different domains and scenarios, in which the availability of concept blending abilities can possibly make a significant difference to state-of-the-art systems.

Over the last decades, AI has made significant progress in the field of problem solving. Still, in most cases, the power of the current approaches is limited, as the danger of the often feared “combinatorial explosion” and the abyss of underspecification of problems are ubiquitous. For certain problems, conceptual blending might offer an alternate way to a solution, avoiding the classical mantraps AI researchers are struggling with. One of the most often cited examples is the “Riddle of the Buddhist Monk” [5]: A Buddhist monk is said to begin with the climb of a mountain at dawn, reaches the top at sunset, spends the night there, and again at dawn begins with the descent, reaching the foot of the mountain at sunset. Now, without making any further assumptions, it shall be decided whether there exists a place on the path which the monk occupies at the same hour of the day on both trips. Instead of computing for a sufficiently fine-grained discretization of the monk’s path the place–time correspondences of both days and checking whether there is such a place (which would also only be possible under violation of the prohibition of further assumptions, as some estimation concerning the monk’s moving speed etc. would have to be made), conceptual blending offers an elegant solution to the problem. Instead of treating both days as separate time spans, they are blended into one, resulting in a scenario in which there seem to be two monks, one starting from the peak, one starting from the foot

of the mountain, moving towards each other, and meeting exactly in the place the riddle asks for (thus not only giving an answer in terms of existence of the place, but also in a constructive way providing implicit information on its location). Although to the best of our knowledge there is not yet a clear characterization of the class of problems in which conceptual blending is first choice, we are positive that a blending mechanism can find application in a great number of practical problems.

A second domain in which concept blending (and possibly also analogy, cf. [3]) may be of use is the area of rationality and rational behavior. For many years, researchers from different areas (e.g. game theory, psychology, logic, and probability theory) have tried to find feasible theories and models for human reasoning and human-style rationality. Sadly, the results of these efforts, though being highly elaborate and theoretically well-founded, mostly fall short when actually predicting human behavior. One of the classical examples in which most of the probability-based models reach their limits is Tversky and Kahneman's "Linda Problem" [23]: Linda is described as (among others) single, outspoken and very bright, with a degree in philosophy, concerned with issues of social justice, and with a past as anti-nuclear activist. Now, several additional characteristics are offered, and subjects are asked to rank these additional properties by estimated likelihood of occurrence. There, humans show to be prone to the conjunction fallacy, as a significant number of subjects ranks the option "*Linda is a bank teller and is active in the feminist movement.*" higher than the single "*Linda is a bank teller.*", contradicting basic laws of probability. Again, conceptual blending offers a way out: Considering a blend of both concepts *bank teller* and *feminist*, maintaining some properties of a cliché account of *bank teller*, and adding key properties from a cliché *feminist* concept, which better fit the initially given properties of Linda, subjects' behavior can be explained and predicted (for a more detailed treatment cf. [17]). Thus, conceptual blending gives an intuitive and comparatively simple explanation to some of the classical challenges to existing theories of rationality, making it also an interesting candidate for inclusion in future general intelligent systems, which undoubtedly will have to deal with issues related to rationality and human behavior.

Also in language understanding and production within an AI system, concept blending can provide additional functionality. Humans in many cases exhibit impressive capabilities in making sense of neologisms and previously unknown word combinations, whilst actual natural language interface systems in many cases are stretched to their limits, although the individual meanings of the combined terms might be known to the system. Concept blending here offers a guideline for combining these concepts into a blend, thus making

accessible also the resulting blended concept, as e.g. in Goguen's by now classical example concerning the combined words HOUSEBOAT and BOATHOUSE mentioned in section 12.4 [8]. Therefore, an integration of concept blending faculties into a general intelligent system from a natural language interface point of view can be expected to be advantageous in both aspects, language production and language understanding: Whilst the output of the system might seem more human-like due to the occasional use of combined words based on blends, also the language input understanding part might profit from these capabilities, making use of blending techniques when trying to find the meaning of unknown, possibly combined words. (For some further considerations concerning concept blending and noun–noun combinations cf. [17].)

A further main domain in which we expect great benefit from an integration and development of concept blending capabilities into general intelligent systems is the branch of (artificial) creativity. Although to the best of our knowledge until today research in artificial general creativity and innovation has mostly been treated as an orphan within the AI community, a proper account of concept blending integrated into a general intelligent system might boost this branch of research in the future. Concept blending has already been identified as a key element in the concept generation stage in creative design processes [18]. It is undeniable that some form of concept blending is always present in most occurrences of human (productive) creativity over time, ranging from mythology and storytelling to product design and lifestyle. On the one hand, the former two provide numerous well-known examples, which by now sometimes even have become a fixed part of human culture: In mythology, Pegasus is a blend between a horse and a bird and centaurs as mythological horsemen have to be considered straightforward blended concepts. More recent storytelling forms of concept blending, for instance, prominently feature in the famous 1865 novel, "Alice's Adventures In Wonderland", in which Lewis Carroll⁵ narrates various scenes that naturally call for blending by readers: In addition to the talking, dancing, wisely-speaking animals and the humanly-reified playing cards, living flamingos appear as mallets and hedgehogs as balls, showing features and properties of both, their original concept and the (by the particular use) newly introduced concept. Ideas behind many modern science fiction novels can also be seen as creatively generated blended concepts. For example, "The Hunger Games" [4] is a young adult science fiction novel by Suzanne Collins, who claims that the idea of the novel occurred to her whilst channel surfing on television. On one channel Collins observed people competing on a reality show and on another she

⁵Lewis Carroll is the pseudonym of the English author Charles Lutwidge Dodgson.

saw footage of the Iraq War. The lines between the show competition and the war coverage began to blur, the two blended together in this very unsettling way, and the idea for the novel was formed. On the other hand, in recent product design, modern tablet computers can be seen as blends between journals and computers, keeping the size and format of a magazine and expanding it with the functionality of a laptop.

So far, many of the examples mentioned in this chapter are related to higher-order cognitive abilities, e.g. problem solving, the invention of new mathematical concepts, or aspects of human rationality. Furthermore, we assumed that the input conceptualizations for the cognitive mechanisms are already given and need not to be generated by the approach. A natural question concerns the generation of such inputs for the system, in particular, if lower cognitive abilities are considered or highly structured background theories are not available, like in the case of infant learning. A nice example of such a set-up was indirectly already mentioned in Section 12.3, namely the grounding of the acquisition of a rudimentary number concept by children in real world domains including real world actions by the learning agent: in this context, we mentioned Lakoff and Núñez's MOTION ALONG A PATH (MAP) and MEASURING STICK (MS) metaphor. Lakoff and Núñez take an embodied and action-oriented stance, in the sense that the acquisition and first approximations of a number concept by children is grounded in real world actions. The natural idea to model this is to take internal action sequences, like walking steps in the (MAP) metaphor, as initial representations for the system. Obviously, certain further action-oriented properties need to be taken into account, for example, the origin of the walking sequence or the possibility to walk back to the origin etc. Currently, we cannot give a formal approach of these issues, but we think that this type of grounding and the generation of on-the-fly representations in such domains are in principle possible.

Finally, we speculate about the potential breadth of applicability of concept blending as described in the HDTP framework and its covered types of reasoning. In Section 12.3 several cognitive mechanisms in the context of cross-domain reasoning mechanisms were already mentioned. These are summarized together with some additional mechanisms, like re-representation, frequency effects, and abduction, in Table 12.1. Although many interesting aspects of higher cognitive abilities are covered by these mechanisms sketched in Table 12.1, we do not claim that this list is in any sense complete. Besides the necessity to enable an AGI system with the mentioned mechanisms it is obvious that a system that is intended to model general intelligence in a holistic and complete sense, needs additionally several further low-level abilities. Obviously, the presented approach does not

Table 12.1 Many desirable functions of intelligent systems can be explained by cross-domain reasoning mechanisms. The left column lists mechanisms most of them introduced in this chapter and associates them with functions that can be based on them. Some mechanisms concern future extensions of the presented framework. The list is not intended to be complete.

Cross-Domain Mechanism	Function in Intelligent Systems
Analogical Mapping	Understanding of new domains; creation and comprehension of metaphorical expressions and analogies in general
Analogical Transfer	Problem solving; introduction of new concepts into a domain; creative invention of new concepts
Generalization	Learning of abstract concepts; compression of knowledge
Specialization	Applying abstract knowledge; problem solving by realizing a general strategy
Congruence Relation	Restructuring of a domain; identification of entities by a new concept
Blending	Creation of new concepts and theories; problem solving; human style rationality; understanding of compound nouns
Re-representation	Adaptation of the input domains in order to compute appropriate analogies and blend spaces
Frequency Effects	Probabilistic Extension for Modeling uncertainty
Abduction	Finding explanations of certain facts

cover aspects like reactive behaviors, sensorimotor loops, or certain attention mechanisms. These challenges should be addressed by other mechanisms that are not directly related to analogy-making and concept blending, because the conceptual level is in such low-level cognitive abilities most often missing. The integration of higher and lower cognitive mechanisms could be achieved by a hybrid architecture, presently a rather standard approach for integrating heterogeneous methodologies.

Taking all this evidence together, we consider concept blending as one of the cornerstones (maybe even a *conditio sine qua non*) of an architecture for general intelligent systems that also shall exhibit creative behavior and significant capabilities in the field of creativity, which in turn might possibly also open up new application areas and industry options to AI, as e.g. computational support systems for innovation and creativity in (product) design, architecture and product-creating arts.

12.7 Conclusions

Artificial general intelligence attempts to find frameworks and to implement systems that can operate not only in a highly specialized domain, namely the domain they were built for, but that show also generalizability properties, i.e. they can operate even in domains they never experienced before. Besides advanced learning capabilities and many classical technologies from AI, such systems should have, as a necessary prerequisite, a creativity potential for finding solution strategies in unknown problem solving set-ups. We claim that the best way to tackle the creativity problem in systems for general intelligence is to consider the underlying cognitive mechanisms that allow humans to be creative to a remarkable degree. This chapter addresses this issue by proposing cognitive mechanisms like analogy-making and theory /concept blending as a source for creativity. Both mechanisms fall under a more general human ability, namely cross-domain reasoning, summarizing several non-classical forms of reasoning that enable the human being to associate domains that are usually considered to be disconnected. We exemplified in some detail the cognitive mechanism theory blending in a concrete example of the invention of the complex plane in mathematics. Although it may seem to be the case that the domain of mathematics is a relatively special case for a general mechanism, we think that the applicability of blending to other domains, problems, set-ups etc. is rather straightforward. Some ideas for this claim are provided in Section 12.6.

We do not claim that the considerations presented in this chapter finally solve the task to make intelligent systems creative. They are rather a first (but important) step towards a theory of creativity. Many formal and methodological challenges still need to be resolved for a uniform framework of creativity. Furthermore, the integration of several non-standard reasoning techniques in one architecture remains a challenge for the future. Although the mentioned HDTP framework integrates several of the mechanisms occurring in this chapter, further work is necessary towards an integration of the methodological variety.

Bibliography

- [1] James Alexander. Blending in mathematics. *Semiotica*, 2011(187):1–48, 2011.
- [2] Jean-Robert Argand. Philosophie mathématique. Essai sur une manière de représenter les quantités imaginaires, dans les constructions géométriques. *Annales de Mathématiques pures et appliquées*, 4:133–146, 1813.
- [3] T. R. Besold, H. Gust, U. Krumnack, A. Abdel-Fattah, M. Schmidt, and K.-U. Kühnberger. An argument for an analogical perspective on rationality & decision-making. In Jan van Eijck and Rineke Verbrugge, editors, *Proceedings of the Workshop on Reasoning About Other Minds*:

- Logical and Cognitive Perspectives (RAOM-2011)*, Groningen, The Netherlands, volume 751 of *CEUR Workshop Proceedings*, pages 20–31. CEUR-WS.org, July 2011.
- [4] Suzanne Collins. *The Hunger Games*. Scholastic, 2010.
 - [5] Gilles Fauconnier and Mark Turner. Conceptual integration networks. *Cognitive Science*, 22(2):133–187, 1998.
 - [6] Gilles Fauconnier and Mark Turner. *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*. Basic Books, New York, 2002.
 - [7] D. Gentner, K. Holyoak, and B. Kokinov, editors. *The Analogical Mind: Perspectives from Cognitive Science*. MIT Press, 2001.
 - [8] Joseph Goguen. An introduction to algebraic semiotics, with application to user interface design. In *Computation for Metaphors, Analogy, and Agents*, volume 1562 of *Lecture Notes in Computer Science*, pages 242–291. Springer, 1999.
 - [9] Joseph Goguen. Mathematical models of cognitive space and time. In D. Andler, Y. Ogawa, M. Okada, and S. Watanabe, editors, *Reasoning and Cognition: Proc. of the Interdisciplinary Conference on Reasoning and Cognition*, pages 125–128. Keio University Press, 2006.
 - [10] Markus Guhe, Alison Pease, Alan Smaill, Maricarmen Martinez, Martin Schmidt, Helmar Gust, Kai-Uwe Kühnberger, and Ulf Krumnack. A computational account of conceptual blending in basic mathematics. *Journal of Cognitive Systems Research*, 12(3):249–265, 2011.
 - [11] Markus Guhe, Alison Pease, Alan Smaill, Martin Schmidt, Helmar Gust, Kai-Uwe Kühnberger, and Ulf Krumnack. Mathematical reasoning with higher-order anti-unification. In *Proceedings of the 32st Annual Conference of the Cognitive Science Society*, pages 1992–1997, 2010.
 - [12] Helmar Gust, Kai-Uwe Kühnberger, and Ute Schmid. Metaphors and Heuristic-Driven Theory Projection (HDTP). *Theoretical Computer Science*, 354:98–117, 2006.
 - [13] Douglas Hofstadter and the Fluid Analogies Research Group. *Fluid Concepts and Creative Analogies. Computer Models of the Fundamental Mechanisms of Thought*. Basic Books, New York, 1995.
 - [14] U. Krumnack, H. Gust, A. Schwering, and K.-U. Kühnberger. Remarks on the meaning of analogical relations. In M. Hutter, E. Baum, and E. Kitzelmann, editors, *Artificial General Intelligence, 3rd International Conference AGI*, pages 67–72. Atlantis Press, 2010.
 - [15] George Lakoff and Rafael Núñez. *Where Mathematics Comes From: How the Embodied Mind Brings Mathematics into Being*. Basic Books, New York, 2000.
 - [16] Saunders Mac Lane. *Categories for the Working Mathematician*. Spinger, Berlin, 2nd edition, 1998.
 - [17] M. Martinez, T. R. Besold, A. Abdel-Fattah, K.-U. Kühnberger, H. Gust, M. Schmidt, and U. Krumnack. Towards a domain-independent computational framework for theory blending. In *AAAI Technical Report of the AAAI Fall 2011 Symposium on Advances in Cognitive Systems*, pages 210–217, 2011.
 - [18] Yukari Nagai. Concept blending and dissimilarity: factors for creative concept generation process. *Design Studies*, 30:648–675, 2009.
 - [19] Allen Newell and Herbert Simon. GPS, a program that simulates human thought. In E. Feigenbaum and J. Feldmann, editors, *Computers and Thought*, pages 279–293. McGraw-Hill, 1963/1995.
 - [20] Gordon D. Plotkin. A note on inductive generalization. *Machine Intelligence*, 5:153–163, 1970.
 - [21] Angela Schwering, Ulf Krumnack, Kai-Uwe Kühnberger, and Helmar Gust. Syntactic principles of Heuristic-Driven Theory Projection. *Journal of Cognitive Systems Research*, 10(3):251–269, 2009.
 - [22] Jaqueline Stedall. *Mathematics emerging: a Sourcebook 1540–1900*. Oxford University Press, Oxford, 2008.
 - [23] A. Tversky and D. Kahneman. Extensional versus intuitive reasoning: The conjunction fallacy in probability judgement. *Psychological Review*, 90(4):293–315, 1983.

Chapter 13

Modeling Motivation and the Emergence of Affect in a Cognitive Agent

Joscha Bach

*Berlin School of Mind and Brain, Humboldt University of Berlin
Unter den Linden 6, 10199 Berlin, Germany*

joscha.bach@hu-berlin.de

Emotion and motivation play a crucial role in directing and structuring intelligent action and perception. Here, we will look at the relationship between motives, affects and higher-level emotions, and give directions for their realization in computational models, with a focus on enabling autonomous goal-setting and learning.

13.1 Introduction

Any functional model that strives to explain the full breadth of mental function will have to tackle the question of understanding emotion, affective states and motivational dynamics (Sloman, 1981; Lisetti and Gmytrasiewicz, 2002). Consequently, the field of affective computing has made considerable progress during the last decades. Today, several families of models are available for incorporation in agent implementations.

This chapter is not concerned with giving a comprehensive review of these approaches (for an overview on current emotion modeling, see Gratch, Marsella, and Petta, 2011; and for a look at its history Hudlicka and Fellous, 1996; Gratch and Marsella, 2005), but mainly with a particular approach: how to build a system that treats emotion and affect as emergent phenomena. I will also refrain from giving a particular implementation (if you are interested in that, see the author's work: Bach, 2009), but instead, I want to supply a general framework for emergent emotions and motivation, that could be adapted to various different agent architectures.

Most emotion models start out from a set of pre-defined low-level or high-level emotions, which are characterized and implemented as stimulus related functional parameters (*appraisals*, see Roseman, 1991; Lazarus, 1991; Ellsworth and Scherer, 2003). These parameters give rise to complex behavioral tendencies, which can then be functionally classified (Ortony, Clore, and Collins, 1988; Plutchik, 1994). The term appraisal describes the relationship between stimulus and emotion; an appraisal is a valenced reaction to a situation, as the agent perceives it. In this view, emotions are triggered by a causal interpretation of the environment (Gratch and Marsella, 2004) with respect to the current goals, beliefs, intentions and relations of the agent. By evaluating these, a frame of the appraisal and a corresponding affective state of the agent are set, which in turn enable it to cope with the situation. Here, coping subsumes the external and the cognitive behaviors with relation to the appraisal: actions and speech acts, as well as the modification of beliefs, intentions, goals and plans. This way, the agent influences the external environment (the world accessible by action and communication) and the internal environment (its model of the world, along with its plans and goals) to address the issues according to their valence and context. Appraisal frame and affective state are the link between external and internal situational stimuli, and the internal and external response (see Figure 13.1).

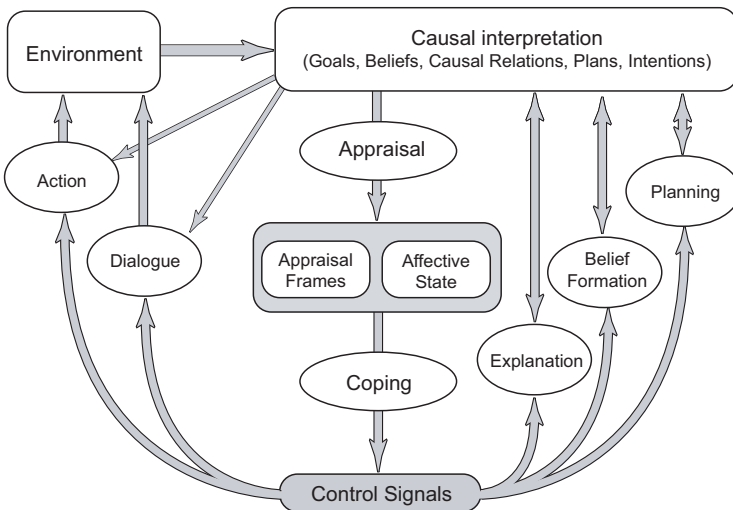


Fig. 13.1 The role of appraisals in the cognitive system (Gratch and Marsella, 2004, p. 11)

Appraisal models have a distinct engineering advantage: if you want to design an agent that expresses an emotion or behavioral tendency in a given situation, associating stimulus and behavior with a functional relationship is a straightforward approach. On the other hand, classical appraisal models are less suited to explain the internal cognitive dynamics and specific interactions of which emotional effects are the result, i.e. they are more concerned with which emotion is expressed when, then what it is like to be in an emotional state, and thereby lose out not only in explanatory power, but also in functionality.

In my view, neither emotion nor motivation are strict requirements for artificial intelligent systems: as we will see, affective modulation and emotions may increase the efficiency of the use of cognitive resources, or aid in communication, but that does hardly imply that cognition and communication would be impossible without these. (Likewise, while motivation may be a requirement for autonomous goal-directed behavior, non-autonomous AIs are entirely conceivable.) So, what exactly are the advantages of emotional systems?

Emotions play a number of important roles in human cognition: They structure and filter perceptual content, according to current needs and processing requirements. They prime the retrieval of memory content, both by providing an associative context, and by informing the resolution and urgency of cognitive operations. They provide valenced feedback (e.g., positive and negative reinforcement signals) for learning, deliberation and decision-making. Furthermore, emotions structure and inform social interaction between individuals, and self-reflection of the individual in social contexts. The effect of emotional states on cognition amounts not just to an adaptation to external events, but in a modulation that enhances the efficiency of the cognitive processing itself: emotional states may affect the mode of access to mental content, and dynamically prune representational structures and decision trees according to a motivational context and an environmental situation. Thus, any agent implementation that wants to model and perhaps utilize these benefits will have to look beyond externally observable behavior and externalist emotion classification, but into the functionality of the affective and motivational system at the level of the agent architecture.

In the following, I will explain how to anchor emotion and motivation within a cognitive architecture, based on the Psi theory, which originates in theoretical psychology (Dörner, 1999; Dörner *et al.*, 2002) and its formalization in MicroPsi (Bach, 2003, 2007, 2009, 2011).

MicroPsi does not treat affective states as distinct events or parameters, but as configurations of the cognitive system. Emotions are modeled as emergent phenomena. They can

be classified and attributed by observing the agent (potentially even classified and observed internally, by the agent itself), but they are not pre-defined entities. Colloquially put, I will describe how to build a system that does not just simulate emotions, but that actually *has* emotions—within the framework of a certain definition of having emotions, of course.

Our model distinguishes between demands of the system, *motives*, cognitive *modulators*, *affects* and *directed emotions*. Demands are pre-defined needs that give rise to goals, but whereas goals depend on the given environment, needs only depend on the agent itself and are part of its definition. When needs are signaled to the cognitive system, they may become motives, and give rise to intentions. Motives and intentions give relevance to objects of the agent's perception, deliberation and action, and therefore define higher-level emotions (such as pride, jealousy or anger). Higher-level emotions are not just characterized by relevance and object, but also by a valence (a representation of desirability) and an affective state. Affects are modulations of cognition, such as arousal, surprise, anxiety, and elation. While we will not characterize each affect or emotion individually and with all the detail that they deserve, this chapter will give an introduction on the relationship between motivation, affect and emotion, and should give you a fair idea how to implement them in an agent model.

On the following pages, we start out by characterizing emotions and affects, before we look at the relevant architectural components for realizing them. These components include a motivational system, decision making functionality, access to mental and perceptual content, cognitive modulation and facilities for reinforcement learning based on valence signals.

13.2 Emotion and affect

Emotions are states and processes that influence the allocation of physical and mental resources, perception, recognition, deliberation, learning and action. This is not a proper definition, of course: the same is true for a lot of other states that we would not intuitively call emotions. A raising or lowering of the blood pressure, for instance, does affect organisms in a similar way, but would not be called an emotion in itself. Emotions are phenomena that manifest on the cognitive and behavioral level, they are not identical to physiological mechanisms. Furthermore, emotion needs to be distinguished from motivational phenomena: while hunger is cognitively represented (as an urge) and implies a valence and an effect on the allocation of mental and physical resources of an organism, it

is not an emotion, but a *motivator*. Generally speaking, motivation determines *what* has to be done, while emotion determines *how* it has to be done.

Defining emotions is a notoriously difficult and contested issue within psychology, so instead of trying my luck with a general definition, we might start with pointing out the following *components* (Diener, 1999) that a theory of emotion should account for:

- Subjective experience (how it *feels* to be in an emotional state)
- Physiological processes (neural, neurochemical and physiological mechanisms that facilitate emotion)
- Expressive behavior (facial and bodily expression, movement patterns, modulation of voice and breath etc.)
- Cognitive evaluations.

These components help us to characterize a range of affective phenomena, like emotional reflexes (especially *startling*), undirected *moods* (euphoria, depression etc.), valenced affective states (joy, distress, anxiety), affects directed upon motivationally relevant states (jealousy, pity, pride) and affects directed upon motivationally relevant *processes* (disappointment, relief).

Emotions are adaptive (i.e. the emotional states an organism is subjected to in a certain situation depend partly on its history of past experiences in similar situations), and the range of emotional states varies between individuals (Russel, 1995). On the other hand, it seems that emotions themselves are not the product of learning and largely invariant in dimensionality and expression (Ekman and Friesen, 1971; Izard, 1994). For instance, individuals may learn in what situations fear is appropriate or inappropriate, as well what kind of fear and what intensity. But the ability to perceive fear itself is not acquired, rather, it stems from the way an organism is equipped to react to certain external or internal stimuli. Thus, it makes sense to develop general taxonomies of emotional states.

For our purpose, we will make the following basic distinctions:

- (i) On the lowest level, we identify *modulators* of cognition, like arousal, and valence. The arousal is a cognitive parameter that controls the general activation and action readiness of an agent, and it is controlled by the urgency of its perceived demands. The valence is a reinforcement signal that emanates from changes in the perceived demands: a rapid lowering of a demand corresponds to a positive valence (“pleasure”), while a rapid increase in a demand corresponds to a negative valence (“distress”).

- (ii) Taken together, the individual modulators define a *configuration*, a mode of cognitive processing. I will call this configuration an *affective state*. Most affective states are valenced (i.e., perceived as desirable or undesirable), with the exception of *surprise*, which can have a neutral valence.
- (iii) Affects that have a mental representation (like a goal situation or another agent) as their object are called *higher-level emotions*. This includes emotions like *disappointment*, which is a negatively valenced affect directed upon a *change* in expectations, i.e. upon a process instead of a state. The relevance of the objects of affects is given by the motivational system of a cognitive agent. Without a corresponding motivator, a mental representation cannot elicit a valenced response, and is therefore emotionally irrelevant.

Note that there are alternatives to this approach of capturing emotions, for instance:

- Modeling emotions as explicit states. Thus, the emotional agent has a number of states it can adopt, possibly with varying intensity, and a set of state transition functions. These states parameterize the modules of behavior, perception, deliberation and so on.
- Modeling emotions by connecting them directly to stimuli, assessments or urges (like hunger or social needs) of the agent. (A similar approach has been suggested by Frijda, 1986.)
- Disassembling emotions into compounds (sub-emotions, basic emotions), and modeling their co-occurrence. Some suggestions for suitable sets of primary emotions and/or emotion determinants have been made by some emotion psychologists (for instance Plutchik, 1994).

Conversely, we are going to capture emotions implicitly by identifying the parameters that modify the agent's cognitive behavior and are thus the correlates of the emotions. The manipulation of these parameters will modify the emotional setting of the agent, and lead to the emergence of affective states.

13.3 Affective states emerging from cognitive modulation

If affects are emergent over continuous basic parameters, they amount to areas in a continuous multidimensional space (with each dimension given by one of the modulators).

One of the first attempts to treat emotion as a continuous space was made by Wilhelm Wundt (1910). According to Wundt, every emotional state is characterized by three com-

ponents that can be organized into orthogonal dimensions. The first dimension ranges from pleasure to displeasure, the second from arousal to calmness, and the last one from tension to relaxation (Figure 13.2), that is, every emotional state can be evaluated with respect to its positive or negative content, its stressfulness, and the strength it exhibits. Thus, an emotion may be pleasurable, intense and calm at the same time, but not pleasurable and displeasurable at once. Wundt's model has been re-invented by Charles Osgood in 1957, with an *evaluation* dimension (for pleasure/displeasure), *arousal*, and *potency* (for the strength of the emotion) (Osgood *et al.*, 1957), and re-discovered by Ertel (1965) as *valence*, *arousal*, and *potency*. Wundt's model does not capture the social aspects of emotion, so it has been sometimes amended to include extraversion/introversion, apprehension/disgust and so on, for instance by Traxel and Heide, who added *submission/dominance* as the third dimension to a *valence/arousal* model (Traxel and Heide 1961).

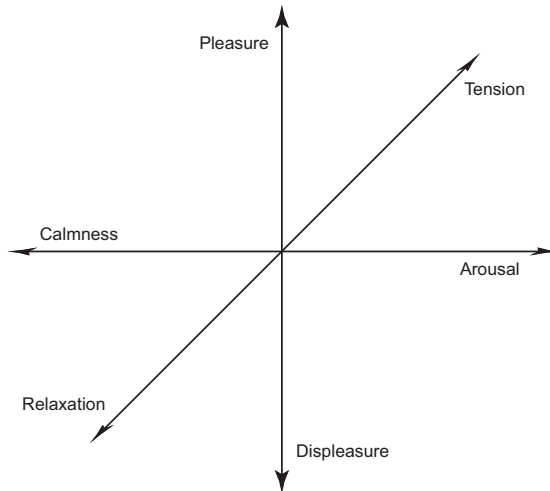


Fig. 13.2 Dimensions of Wundt's emotional space (see Wundt, 1910).

Note that *arousal*, *valence* and *stress* are themselves not emotions, but mental configuration parameters that are much closer to the physiological level than actual emotions—they are modulators. Affects, i.e. undirected emotions, are regions within the space spanned by the modulator dimensions.

The model implemented in MicroPsi suggests a very specific set of modulators, which are determined by additional architectural assumptions within MicroPsi. If your model is

built on a different cognitive architecture, the possible set of suitable modulators will be different.

MicroPsi's modulators include:

- *Valence*: events and stimuli that influence demands of the agent are accompanied with a valence signal (pleasure or displeasure).
- *Arousal*: this is related to the urgency of the currently active demands. High demands result in a high arousal. The arousal has behavioral implications, it roughly corresponds to the physiological *unspecific sympathicus syndrome* in humans, and increases goal directedness. In biological systems, arousal also controls the allocation of physiological resources, for instance, it diverts oxygen to increase muscular responses at the cost of digestive processes.
- *Resolution level*: speed and accuracy of perception, memory access and planning are inversely related. The resolution level controls this trade-off, in MicroPsi by adjusting the width and depth of activation spreading in the agent's representations. A high resolution level results in slow processing, but a deep and detailed perceptual/memory exploration. Conversely, low resolution will speed up processing at the cost of accuracy. The resolution level is inversely proportional to the arousal.
- *Selection threshold*: this parameter controls the stability of the dominant motive. It amounts to an adaptive "stubbornness" of the agent and avoids goal oscillation when demands change. A high urgency and importance of the leading motive will increase the selection threshold, making it harder for competing motives to seize control.
- *Goal directedness*, as mentioned before, depends on the level of arousal, and influences the selection of action vs. deliberation/exploration strategies.
- *Securing rate*: controls the rate of background checks of the agent, and thus the allocation of attention between perception and other cognitive processing.

These dimensions account for behavioral tendencies and physiological correlates of affective states. Anger, for instance, is characterized by high arousal, low resolution, strong motive dominance, few background checks and strong goal-directedness; sadness by low arousal, high resolution, strong dominance, few background-checks and low goal-directedness. The modulators also account for much of the hedonic aspect of emotions, i.e., the specific way emotion is reflected via proprioception. For example, the high arousal of anger will lead to heightened muscular tension and a down-regulation of digestion, which adds to the specific way anger feels to an individual.

For a more detailed look at the relations between the modulator dimensions, see Figure 13.3.

The states of the modulators (the proto-emotional parameters) are a function of the urgency and importance of motives, and of the ability to cope with the environment and the tasks that have to be fulfilled to satisfy the motives. As illustrated in Figure 13.3, a high *urgency* of the leading motive will decrease resolution and increase goal orientation and selection threshold (motive dominance), while less time is spent checking for changes in the background; whereas a high *importance* of the leading motive will increase the resolution level. Uncertainty and lack of experience (*task specific competence*) increase the rate of securing behavior. A high level of confidence (*general competence*) increases the selection threshold, and the arousal is proportional to the general demand situation (urgency and importance of all motives). Uncertainty is measured by comparing expectations with events as they happen, competence depends on the rate of success while attempting to execute goal-directed behavior. Furthermore, the modulator dynamics are controlled by the strength of the leading motive (importance), the time left to satisfy it (urgency), the current selection threshold and the expected chance to satisfy it (*task specific competence*). As mentioned before, motive importancies and urgencies are supplied by the motivational system.

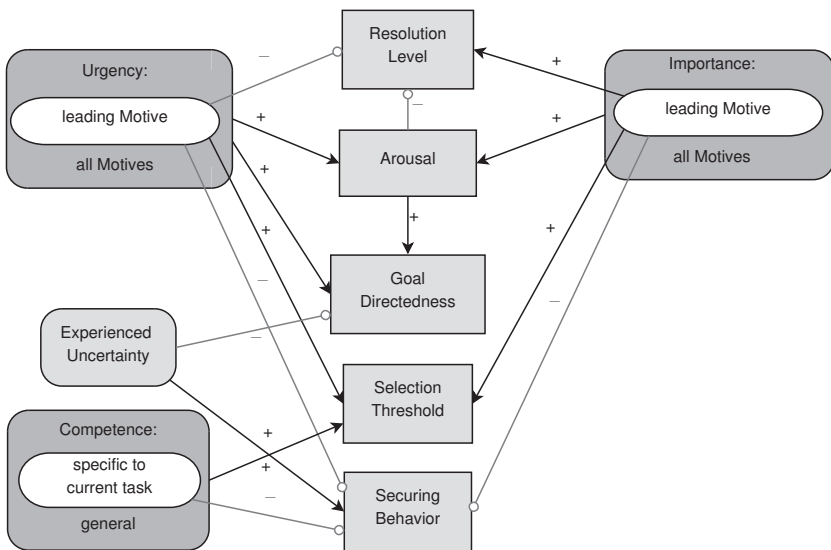


Fig. 13.3 Dimensions of emotion according to the Psi theory (adopted from Hille, 1998).

The six-dimensional sketch is not exhaustive; especially when looking at social emotions, at least the demands for *affiliation* (external legitimacy signals) and “honor” (internal legitimacy, ethical conformance), which are *motivational dimensions* like *competence* and *uncertainty reduction*, would need to be added.

The Psi theory is not a complete theory of human motivation yet. Extensions and modifications to the modulator dimensions will have an effect on the scope of the emergent affective states. While this approach characterizes the nature of affects well (by treating them as the result of cognitive modulation), the resulting emotional categories are likely to exhibit differences to the set of human emotions. This argument could be extended to animal cognition: while most vertebrates and all mammals certainly have emotions, in the sense that their cognition is modulated by valence, arousal, resolution level and so on, their emotions might be phenomenologically and categorically dissimilar to human emotions, because they have a different motivational system, different cognitive capabilities and organization, and perhaps even different modulators.

Nonetheless, this simple model is already sufficient to address the affective level, both conceptual and with detailed implementation, and its explanatory power is sufficient to account for subtle differences, like the one between enthusiastic joy and bliss (both are characterized by strong positive valence, but bliss is accompanied by a low arousal and high resolution level).

13.4 Higher-level emotions emerging from directing valenced affects

When we talk about emotions, we do not just refer to affective configurations like joy and bliss, but mostly to reactions to complex situations, such as social success, impending danger, a goal reached due to the beneficial influence of another agent. These higher-level emotions are characterized by an affective state that is directed upon (i.e., functionally associated to) a motivationally relevant object. Perhaps the most successful and prominent model of this category is based on work by Ortony, Clore and Collins (1988). The OCC model is easily adapted for the simulation of believable behavior, and even though it does not account for more specific emotions, like jealousy or envy, its application and extension is rather straightforward.

Ortony, Clore and Collins distinguish three main classes of emotions with respect to their *object*, which is either the consequence of some event, an aspect of some thing, or the action of some agent. From this perspective, the difference between social emotions (the

appraisal of the actions of oneself or other agents) and event-based emotions (hope, relief) becomes visible (Figure 13.4).

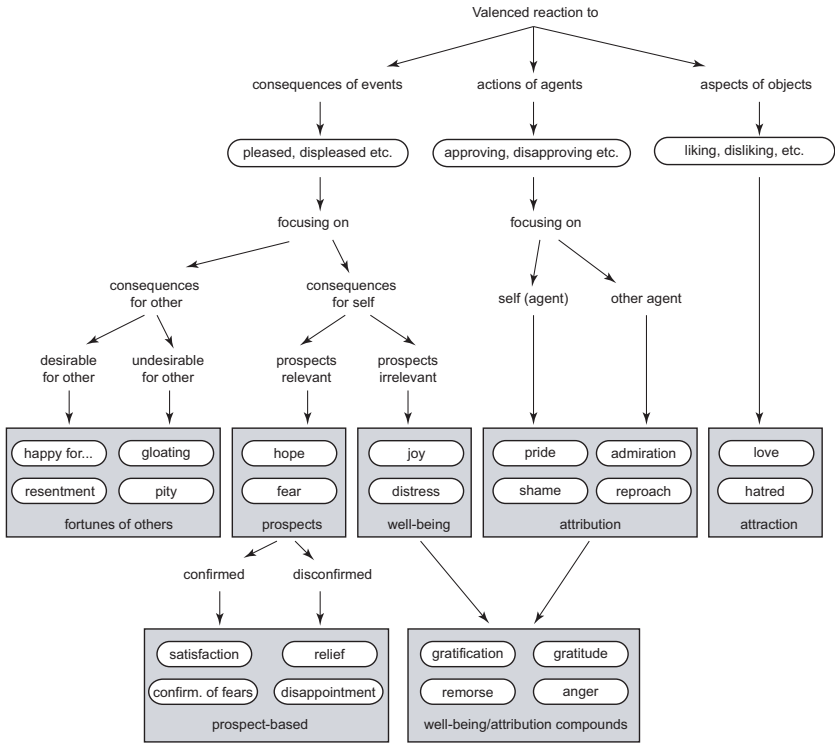


Fig. 13.4 Taxonomy of higher-level emotions (after Ortony, Clore and Collins, 1988, p. 19).

At the first stage, the OCC model distinguishes for each group of emotions whether they are positively or negatively valenced; for events this is their degree of pleasurable-ness, resentment etc., for agents it is approval or disapproval, and for objects it is their desirability, and thus the degree of attraction or repulsion.

Event-related emotions are further separated depending on whether the consequences apply to others or oneself, and whether the event is already present or not. Present events may lead to joy or distress, anticipated events to hope and fear. If the anticipated events materialize, the reactions are either satisfaction or the confirmation of fear, if they do not occur, then the consequence is either disappointment or relief.

Emotions with respect to events happening to others depend on the stance taken towards these others—if they are seen positively, reactions may be happiness for them, or pity (if the

event has negative consequences). If the others are resented, then a positive outcome may lead to envy and resentment, a negative to gloating.

Agent-directed emotions (attributions) depend on whether the agent in question is someone else (who may be admired or reproached), or one self (in which case the emotion could be pride or shame).

Of course, appraisals may also relate to the consequences of events that are caused by the actions of agents. The OCC taxonomy calls the resulting emotions “attribution/well-being compounds”: Here, if oneself is responsible, the reaction may be gratification or remorse, and if the culprit is someone else, it could be gratitude or anger.

Every emotion can be specified in a formal language, by using threshold parameters to specify intervals of real-valued variables in a weight-matrix to describe

- *for events*: their desirability for the agent itself, their desirability for others, their deservingness, their liking, the likelihood of their occurrence, the related effort, and whether they are realized,
- *for agents*: their praiseworthiness, their cognitive relevance, the deviation of expectations,
- *for objects*: their appeal and their familiarity.

By setting the thresholds accordingly, the emotion model can be tuned to different applications, or to individual variances. The OCC taxonomy is an engineering approach that constructs the emotion categories based on systematizing our common-sense understanding of emotion. It does not say much about how the cognitive appraisals are realized – this is left to the designer of an architecture for a believable agent, nor does it explain the reason or the behavioral result of the individual higher-level emotions. However, if used in conjunction with a motivational system, it characterizes the emotions that emerge from dynamically attaching relevance to situations, agents and expectations.

13.5 Generating relevance: the motivational system

In my view, emotion is best understood as part of a larger cognitive architecture, including a motivational system that determines relevance. Desires and fears, affective reflexes and mood changes correspond to *needs*, such as environmental exploration, identification and avoidance of danger, and the attainment of food, shelter, cooperation, procreation, and intellectual growth. Since the best way to satisfy the individual needs varies with the en-

vironment, the motivational system is not aligned with particular *goal situations*, but with the needs themselves, through a set of *drives*.

Let us call events that satisfy a need of the system a *goal*, or an *appetitive event*, and one that frustrates a need an *aversive event* (for instance, a failure or an accident). Since goals and aversive events are given by an open environment, they can not be part of the definition of the architecture, and the architecture must specify a set of drives according to the needs of the system. Drives are indicated to the system as *urges*, as signals that make a need apparent. An example of a need would be nutrition, which relates to a drive for seeking out food. On the cognitive level of the system, the activity of the drive is indicated as *hunger*.

The connection between urges and events is established by *reinforcement learning*. In our example, that connection will have to establish a representational link between the indicator for food and a *consumptive action* (i.e., the act of ingesting food), which in turn must refer to an environmental situation that made the food available. Whenever the urge for food becomes active in the future, the system may use the link to retrieve the environmental situation from memory and establish it as a goal.

Needs

All urges of the agent stem from a fixed and finite number of “hard-wired” needs’, implemented as parameters that tend to deviate from a target value. Because the agent strives to maintain the target value by pursuing suitable behaviors, its activity can be described as an attempt to maintain a *dynamic homeostasis*.

The current agent model of the Psi theory suggests several “physiological” needs (fuel, water, intactness), two “cognitive” needs (certainty, competence) and a social need (affiliation). It is straightforward to extend this set, for instance by adding a need for warmth to the set of physiological demands.

All behavior of Psi agents is directed towards a goal situation that is characterized by a *consumptive action* satisfying one of the needs. In addition to what the physical (or virtual) embodiment of the agent dictates, there are cognitive needs that direct the agents towards exploration and the avoidance of needless repetition.

The needs of the agent are weighted against each other, so differences in importance can be represented.

Physiological needs *Fuel and water*: In the MicroPsi simulations, water and fuel are used whenever an agent executes an action, especially locomotion. Certain areas of the environment caused the agent to lose water quickly, which associated them with additional negative reinforcement signals.

Intactness: Environmental hazards may damage the body of the agent, creating an increased intactness need and thus leading to negative reinforcement signals (akin to *pain*). If damaged, the agent may look for opportunities for repair, which in turn increase intactness.

These simple needs can be extended at will, for instance by needs for shelter, for rest, for exercise, for certain types of nutrients etc.

Cognitive needs *Certainty*: To direct agents towards the exploration of unknown objects and affairs, they possess an urge specifically for the reduction of uncertainty in their assessment of situations, knowledge about objects and processes and in their expectations. Because the need for certainty is implemented similar to the physiological urges, the agent reacts to uncertainty just as it would to pain signals and will display a tendency to remove this condition. This is done by triggering explorative behavior.

Events leading to an urge for uncertainty reduction include:

- the agent meets unknown objects or events,
- for the recognized elements, there is no known connection to behavior—the agent has no knowledge what to do with them,
- there are problems to perceive the current situation at all,
- there has been a breach of expectations; some event has turned out differently as anticipated,
- over-complexity: the situation changes faster than the perceptual process can handle,
- the anticipated chain of events is either too short or branches too much. Both conditions make predictions difficult.

In each case, the uncertainty signal is weighted according to the relation to the appetitive or aversive relevance of the object of uncertainty.

The urge for certainty may be satisfied by “certainty events”—the opposite of uncertainty events:

- the complete identification of objects and scenes,
- complete embedding of recognized elements into agent behaviors,
- fulfilled expectations (even negative ones),
- a long and non-branching chain of expected events.

Like all urge-satisfying events, certainty events create a positive reinforcement signal and reduce the respective need. Because the agent may anticipate the reward signals from successful uncertainty reduction, it can actively look for new uncertainties to explore (“diversive exploration”).

Competence: When choosing an action, Psi agents weight the strength of the corresponding urge against the chance of success. The measure for the chance of success to satisfy a given urge using a known behavior program is called “specific competence”. If the agent has no knowledge on how to satisfy an urge, it has to resort to “general competence” as an estimate. Thus, general competence amounts to something like self-confidence of the agent, and it is an urge on its own. (Specific competencies are not urges.)

The general competence of the agent reflects its ability to overcome obstacles, which can be recognized as being sources of negative reinforcement signals, and to do that efficiently, which is represented by positive reinforcement signals. Thus, the general competence of an agent is estimated as a floating average over the reinforcement signals and the inverted displeasure signals. The general competence is a heuristics on how well the agent expects to perform in unknown situations.

As in the case of uncertainty, the agent learns to anticipate the positive reinforcement signals resulting from satisfying the competence urge. A main source of competence is the reduction of uncertainty. As a result, the agent actively aims for problems that allow to gain competence, but avoids overly demanding situations to escape the frustration of its competence urge. Ideally, this leads the agent into an environment of medium difficulty (measured by its current abilities to overcome obstacles).

Aesthetics: Environmental situations and relationships can be represented in infinitely many ways. Here “aesthetics” corresponds to a need for improving representations, mainly by increasing their sparseness, while maintaining or increasing their descriptive qualities.

Social needs *Affiliation:* Because the explorative and physiological desires of our agents are not sufficient to make them interested in each other, they have a need for positive social signals, so-called “*legitimacy signals*”. With a legitimacy signal (or *l-signal* for short), agents may signal each other “okayness” with regard to the social group. Legitimacy signals are an expression of the sender’s belief in the social acceptability of the receiver. The need for l-signals needs frequent replenishment and thus amounts to an urge to affiliate with other agents. Agents can send l-signals to reward each other for successful cooperation.

Anti-l-signals are the counterpart of l-signals. An anti-l-signal (which basically amounts to a frown) “punishes” an agent by depleting its legitimacy reservoir.

Agents may also be extended by *internal l-signals*, which measure the conformance to internalized social norms.

Supplicative signals are “pleas for help”, i.e. promises to reward a cooperative action with l-signals or likewise cooperation in the future. Supplicative signals work like a specific kind of anti-l-signals, because they increase the legitimacy urge of the addressee when not answered. At the same time, they lead to (external and internal) l-signals when help is given. They can thus be used to trigger *altruistic behavior*.

The need for l-signals should adapt to the particular environment of the agent, and may also vary strongly between agents, thus creating a wide range of types of social behavior. By making the receivable amount of l-signals dependent of the priming towards particular other agents, Psi agents might be induced to display “*jealous*” behavior.

Social needs can be extended by romantic and sexual needs. However, there is no explicit need for social power, because the model already captures social power as a specific need for competence—the competence to satisfy social needs.

Even though the affiliation model is still fragmentary, we found that it provides a good handle on the agents during experiments. The experimenter can attempt to induce the agents to actions simply by the prospect of a smile or frown, which is sometimes a good alternative to a more solid reward or punishment.

Appetence and aversion

In order for an urge to have an effect on the behavior on the agent, it does not matter whether it *really* has an effect on its (physical or simulated) body, but that it is represented in the proper way within the cognitive system. Whenever the agent performs an action or is subjected to an event that reduces one of its urges, a reinforcement signal with a strength that is proportional to this reduction is created by the agent’s “pleasure center”. The naming of the “pleasure” and “displeasure centers” does not necessarily imply that the agent experiences something like pleasure or displeasure. The name refers to the fact that—like in humans—their purpose lies in signaling the reflexive evaluation of positive or harmful effects according to physiological, cognitive or social needs. (*Experiencing* these signals would require an observation of these signals at certain levels of the perceptual system of the agent.)

Motives

A motive consists of an urge (that is, the value of an indicator for a need) and a goal that has been associated to this indicator. The goal is a situation schema characterized by an action or event that has successfully reduced the urge in the past, and the goal situation tends to be the end element of a behavior program. The situations leading to the goal situation—that is, earlier stages in the connected occurrence schema or behavior program—might become intermediate goals. To turn this sequence into an instance that may initiate a behavior, orient it towards a goal and keep it active, we need to add a connection to the pleasure/displeasure system. The result is a *motivator* and consists of:

- a *need sensor*, connected to the pleasure/displeasure system in such a way, that an increase in the deviation of the need from the target value creates a displeasure signal, and a decrease results in a pleasure signal. This reinforcement signal should be proportional to the strength of the increment or decrement.
- optionally, a *feedback loop* that attempts to normalize the need automatically
- an urge indicator that becomes active if there is no way of automatically adjusting the need to its target value. The urge should be proportional to the need.
- an *associator* (part of the pleasure/displeasure system) that creates a connection between the urge indicator and an episodic schema/behavior program, specifically to the aversive or appetitive goal situation. The strength of the connection should be proportional to the pleasure/displeasure signal. Note that usually, an urge gets connected with more than one goal over time, since there are often many ways to satisfy or increase a particular urge.

13.6 Motive selection

The motivational system defines the action control structures of a cognitive agent. Above the level of motives and demands, motives must be chosen, i.e. established as *leading motives*, or *intentions*. Intentions amount to selected motives, combined with a way to achieve the desired outcome. Within MicroPsi, an *intention* refers to the set of representations that initiates, controls and structures the execution of an action. (It is not required that an intention be conscious, that it is directed onto an object etc.—here, intentions are simply those things that make actions happen.)

Intentions may form *intention hierarchies*, i.e. to reach a goal it might be necessary to establish sub-goals and pursue these. An intention can be seen as a set of a goal state,

an execution state, an intention history (the protocol of operations that took place in its context), a plan, the urge associated with the goal state (which delivers the relevance), the estimated specific competency to fulfill the intention (which is related to the probability of reaching the goal) and the time horizon during which the intention must be realized (Figure 13.5).

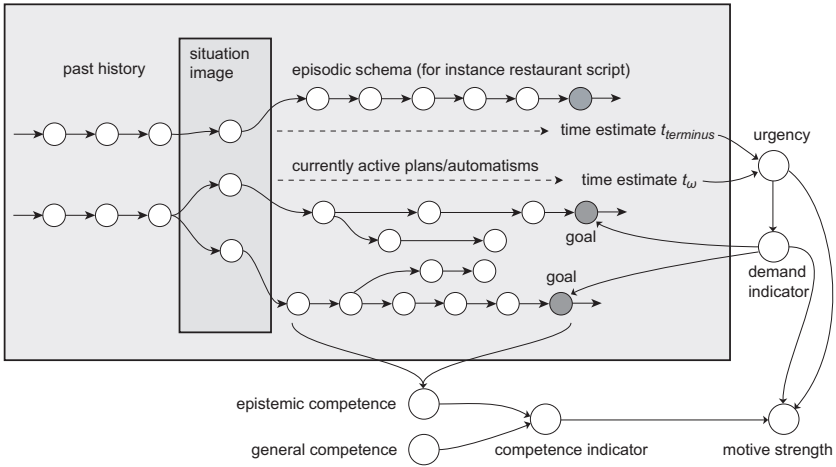


Fig. 13.5 The structure of a motive.

If a motive becomes active, it is not always selected immediately; sometimes it will not be selected at all, because it conflicts with a stronger motive or the chances of success when pursuing the motive are too low. In the terminology of *Belief-Desire-Intention agents* (Bratman, 1987), motives amount to *desires*, selected motives give rise to goals and thus are *intentions*. Active motives can be selected at any time, for instance, an agent seeking fuel could satisfy a weaker urge for water on the way, just because the water is readily available, and thus, the active motives, together with their related goals, behavior programs and so on, are called *intention memory*. The selection of a motive takes place according to a *value* by *success probability* principle, where the value of a motive is given by its importance (indicated by the respective urge), and the success probability depends on the competence of the agent to reach the particular goal.

In some cases, the agent may not know a way to reach a goal (i.e., it has no epistemic competence related to that goal). If the agent performs well in general, that is, it has a high *general* competence, it should still consider selecting the related motive. The chance

to reach a particular goal might be estimated using the sum of the general competence and the epistemic competence for that goal. Thus, the *motive strength* to satisfy a need d is calculated as $urged_d \cdot (generalCompetence + competenced_d)$, i.e. the product of the strength of the urge and the combined competence.

If the window of opportunity is limited, the motive strength should be enhanced with a third factor: *urgency*. The rationale behind urgency lies in the aversive goal created by the anticipated failure of meeting the deadline. The urgency of a motive related to a time limit could be estimated by dividing the time needed through the time left, and the motive strength for a motive with a deadline can be calculated using $(urged_d + urgencyd_d) \cdot (generalCompetence + competenced_d)$, i.e. as the combined urgency multiplied with the combined competence. The time the agent has left to reach the goal can be inferred from episodic schemas stored in the agent's current expectation horizon, while the necessary time to finish the goal oriented behavior can be determined from the behavior program. (Obviously, these estimates require a detailed anticipation of things to come, which may be difficult to obtain.)

At each time, only one motive is selected for the execution of its related behavior program. There is a continuous competition between motives, to reflect changes in the environment and the internal states of the agent. To avoid oscillations between motives, the switching between motives may be taxed with an additional cost: the *selection threshold*. As explained in Section 13.3, this amounts to a bonus that is added to the strength of the currently selected motive. The value of the selection threshold can be varied according to circumstances, rendering the agent "opportunistic" or "stubborn".

13.7 Putting it all together

The functionality described in the previous sections amounts to structural components of a more general affective agent architecture. While this functionality constrains the design space considerably, I would like to add some more necessary detail to the sketch of the framework for cognitive agents, according to the Psi theory.

If motive selection and affective modulation are indispensable, then we will need operations that can be objects of motives, representations that can capture them, and processes that can be modulated. Let me briefly address these requirements (Figure 13.6).

Our agent will need to have the following facilities:

- A *set of urges* that signal demands of the agent.

- A *selection mechanism* that rises the satisfaction of one of the urges to an intention (an *active motive*).
- An *action selection/planning* mechanism that chooses actions to reach the goal associated with satisfying the urge.
- An *associative memory*, containing a *world model* (situation image), a protocol memory (i.e., representations of the *past*), and a *mental stage* (a representation of counterfactual events, such as plans and expectations). The representations in this memory can be *primed* (pre-activated or biased for) by active motives.
- *Perceptual mechanisms* that give rise to the world model.
- *Action execution mechanisms* that actually perform the chosen actions.
- A set of *modulators* that modify the access to memory content, and the way perception, action selection and action execution work.
- A *reinforcement learning mechanism* that creates associations between urges, goal situations and aversive events, based on the effect that encountered situations have on the demands.

These simple requirements leave room for a large variety of possible architectures, and yet they demonstrate how motivation, affect and higher-level emotions can emerge from low more basic functionality. Obviously, emotions are not *part* of this architecture, they are not built into it explicitly, but they can be consistently *attributed* to it. I believe that this reflects an important aspect of the nature of emotions that incidentally causes so much difficulty in their definition and treatment. While emotions are an important part of the descriptions that we use to interpret others, and not least reflect ourselves, they are probably not best understood as natural kinds. Rather, the experience of an emotion amounts to a *perceptual gestalt* (Castelfranchi and Miceli, 2009), an associated, internally perceived co-occurrence of aspects of modulation, proprioceptive correlates of this modulation, of valence, action tendencies and motivationally relevant mental content. Similarities in these perceptual gestalts allow us to group emotions into (by and large) distinct, (mostly) recognizable categories. Attributing the same set of categories to other agents helps us to interpret and predict their behavior, expression, past, and future reactions. And yet, emotions are probably best characterized by decomposing them into their modulatory and motivational components.

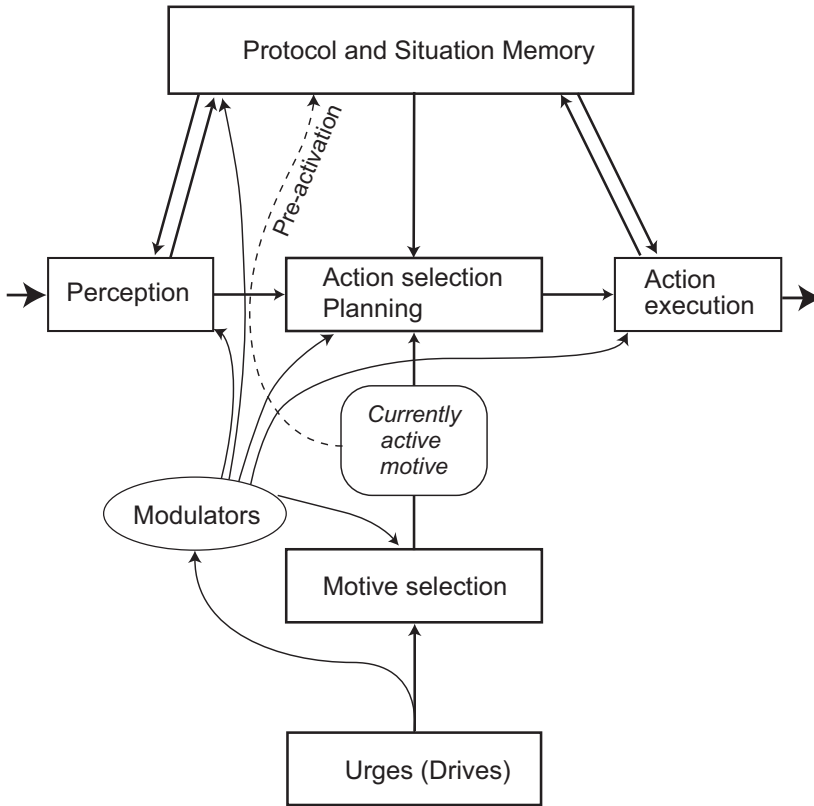


Fig. 13.6 Minimal architectural requirements for emergent emotions.

Acknowledgments

This work was supported by a postdoctoral fellowship grant of the Berlin School of Mind and Brain, Humboldt University of Berlin. I would like to thank Pei Wang, the editor of this volume, and many of its contributors for inspiring and helpful discussions, and the anonymous reviewers for constructive feedback.

Bibliography

- [1] Bach, J. (2003). The MicroPsi Agent Architecture Proceedings of ICCM-5, International Conference on Cognitive Modeling, Bamberg, Germany, 15–20.
- [2] Bach, J. (2007). Motivated, Emotional Agents in the MicroPsi Framework, in Proceedings of 8th European Conference on Cognitive Science, Delphi, Greece.

- [3] Bach, J. (2009). *Principles of Synthetic Intelligence. Psi, an architecture of motivated cognition.* Oxford University Press.
- [4] Bach, J. (2011). A Motivational System for Cognitive AI. In Schmidhuber, J., Thorisson, K.R., Looks, M. (eds.): *Proceedings of Fourth Conference on Artificial General Intelligence*, Mountain View, CA. 232–242.
- [5] Bratman, M. (1987). *Intentions, Plans and Practical Reason.* Harvard University Press.
- [6] Castelfranchi, C., Miceli, M. (2009). The cognitive-motivational compound of emotional experience. *Emotion Review*, 1, 223–231.
- [7] Diener, E. (1999). Special section: The structure of emotion. *Journal of Personality and Social Psychology*, 76, 803–867.
- [8] Dörner, D. (1999). *Bauplan für eine Seele.* Reinbeck: Rowohlt.
- [9] Dörner, D., Bartl, C., Detje, F., Gerdes, J., Halcour, (2002). *Die Mechanik des Seelenwagens. Handlungsregulation.* Verlag Hans Huber, Bern.
- [10] Ekman, P., Friesen, W. (1971). Constants across cultures in the face and emotion. In: *Journal of Personality and Social Psychology* 17(2): 124–29.
- [11] Ellsworth, P.C., Scherer, K.R. (2003). Appraisal processes in emotion. in Davidson, R.J., Goldsmith, H.H., Scherer, K.R. (Eds.) *Handbook of the affective sciences.* New York, Oxford University Press.
- [12] Ertel, S. (1965). EED - Ertel-Eindrucksdifferential (PSYINDEX Tests Review). *Zeitschrift für experimentelle und Angewandte Psychologie*, 12, 22–58.
- [13] Frijda, N.H. (1986). *The emotions.* Cambridge, U.K., Cambridge University Press.
- [14] Frijda, N. (1987). Emotion, cognitive structure, and action tendency. *Cognition and Emotion*, 1, 115–143.
- [15] Gratch, J., Marsella, S. (2004). A framework for modeling emotion. *Journal of Cognitive Systems Research*, Volume 5, Issue 4, 2004, p. 269–306.
- [16] Hudlicka, E., Fellous, J.-M. (1996). Review of computational models of emotion (Technical Report No. 9612). *Psychometrix.* Arlington, MA.
- [17] Izard, C.E. (1994). Innate and universal facial expressions: Evidence from developmental and cross-cultural research. *Psychological Bulletin*, 115, 288–299.
- [18] Lazarus, R. (1991). *Emotion and Adaptation*, NY, Oxford University Press.
- [19] Lisetti, C., Gmytrasiewicz, P. (2002). Can a rational agent afford to be affectless? A formal approach. *Applied Artificial Intelligence*, 16, 577–609.
- [20] Marsella, S., Gratch, J., Petta, P. (in press): Computational Models of Emotion. In Scherer, K.R., Bänziger, T., Roesch, E. (eds.). *A blueprint for an affectively competent agent: Cross-fertilization between Emotion Psychology, Affective Neuroscience, and Affective Computing.* Oxford University Press.
- [21] Sloman, A. (1981). Why robots will have emotions. *Proceedings IJCAI.*
- [22] Ortony, A., Clore, G.L., Collins, A. (1988). *The cognitive structure of emotions.* Cambridge, U.K., Cambridge University Press.
- [23] Osgood, C.E., Suci, G.J. Tannenbaum, P.H. (1957). *The measurement of meaning.* Urbana: University of Illinois Press.
- [24] Plutchik, R. (1994). *The Psychology and Biology of Emotion.* New York: Harper Collins.
- [25] Roseman, I.J. (1991). Appraisal determinants of discrete emotions. In: *Cognition and Emotion*, 3, 161–200.
- [26] Russel, J.A. (1995). Facial expressions of emotion. What lies beyond minimal universality. *Psychological Bulletin*, 118, 379–391.
- [27] Wundt, W. (1910). Gefühlselemente des Seelenlebens. In: *Grundzüge der physiologischen Psychologie II.* Leipzig: Engelmann D.R. Bates, *Phys. Rev.*, 492 (1950).

Chapter 14

AGI and Machine Consciousness

Antonio Chella¹ and Riccardo Manzotti²

¹ *DICGIM - University of Palermo, Viale delle Scienze, Building 6, 90128 Palermo, Italy*

² *Institute of Consumption, Communication and Behavior, IULM University, Via Carlo Bo, 8, 16033 Milano, Italy*

antonio.chella@unipa.it, riccardo.manzotti@iulm.it

*Could consciousness be a theoretical
time bomb, ticking away in the belly of
AI? Who can say?*

John Haugeland [24] (p. 247)

This review discusses some of main issues to be addressed to design a conscious AGI agent: the agent's sense of the body, the interaction with the environment, the agent's sense of time, the free will of the agent, the capability for the agent to have some form of experience, and finally the relationship between consciousness and creativity.

14.1 Introduction

Artificial General Intelligence aims at finding computational models for the deep basic mechanisms of intelligence [50].

The classical view of intelligence from the standpoint of psychology is usually related with the *g* factor and the *psychometric* approach (see Neisser *et al.* [45] for a review). Alternative views have been proposed by Gardner [22] and by Sternberg [61]. Legg and Hutter [36] provided several different definitions related to the different aspects of intelligence, from psychology to philosophy to cognitive sciences.

However, whatever intelligence is, one may wonder if an agent can actually be intelligent without facing the problem of consciousness (for an introductory textbook on consciousness see Blackmore [6]).

The relationship between consciousness and intelligence appears to be subtle. On the one side the need of awareness for an intelligent agent is obviously given for granted. On the other side, it is suspected that many cognitive processes that are necessary for acting intelligently may be unconscious and happen in the absence of consciousness. However, it is undeniable that consciousness is closely related with the broader unpredictable and less automatic forms of intelligence.

Briefly, we may distinguish between two main aspects of intelligence: an aspect that we may call *syntactic* and another one that we may call *semantic*. The first one is related with the capability to suitably manipulate by combining and recombining a fixed set of symbols by *brute force* and according to some heuristics. Many successful AI systems belong to this category.

The second aspect is related to the smarter capabilities required to generate meaning and to ground symbols. While the syntactic manipulation of symbols could occur without consciousness, it is claimed that the meaningful intelligent act does not seem to be possible without consciousness. If we want to build a real smart AGI based agent, we have to face the problem of consciousness.

In the following, we review some of the main issues in the current research field of machine consciousness that are most strictly related with the problem of building an AGI system. In particular, Section 14.2 briefly outlines the main philosophical positions about consciousness, Section 14.3 introduces the framework of machine consciousness, while Section 14.4 and Section 14.5 discuss the sense of the body and the interactions with the environment for a conscious AGI agent. Section 14.6 analyzes the problem of agent's time and Section 14.7 the free will for an AGI system. Section 14.8 discusses the capability for an AGI agent to have some form of experience, and finally, Section 14.9 analyzes some of the relationships between consciousness and creativity in the framework of an AGI system.

14.2 Consciousness

The prestigious journal *Science*, in a special collection of articles published beginning July 2005, celebrated the journal's 125th anniversary with a look forward at the most compelling puzzles and questions facing scientists today. The first one was the question about

what the universe is made of, and the second one concerned the biological basis of consciousness [43]. Therefore, the scientific interest towards the understanding of consciousness is far higher than other scientific questions.

But what is consciousness? Consciousness is an elusive concept which “resists to definitions”, as the Oxford Companion to Philosophy admits [27]. However, consciousness is not the only one concept in science which is difficult to define: the same is true for other concepts as e.g., *life* or *intelligence*, not to mention the fundamental quantities of physics as *space*, *time*, *matter*, and nevertheless science has been able to make important steps in dealing with these elusive concepts.

John Searle [57] proposed the following vivid definition of consciousness (p. 559):

Consciousness consists of inner, qualitative, subjective states and processes of sentience or awareness. Consciousness, so defined, begins when we wake in the morning from a dreamless sleep and continues until we fall asleep again, die, go into a coma, or otherwise become “unconscious.”

Searle evidences that conscious states are inner, qualitative and subjective, and many definitions proposed in the cognitive science literature concur in considering consciousness as tightly related with our subjective experience of the world.

The philosopher Thomas Nagel, in a famous essay on “what it is like to be a bat” [44], discussed the difficulty of studying conscious experience from a scientific point of view. In fact, he argues that we know everything about sonars and how they work, but we can never experience the sense of distance as a bat does by using sonar principles. The sonar experience is purely subjective and the physical analysis of the bat’s brain is compatible with absence of experience. Therefore, Nagel concludes that experience cannot be reduced to the physical domain.

Ned Block [7] contrasts *P-consciousness* and *A-consciousness*. *P-consciousness* is the phenomenal aspect of consciousness and refers to subjective experience, while *A-consciousness* is related with the function of consciousness. Therefore, *A-consciousness* deals with different cognitive capabilities such as reasoning, action, control of speech and so on, and may be the subject of scientific studies. *P-consciousness*, according to Block, is subject to an “explanatory gap”, i.e., how to step from physical and biological descriptions of processes in the brain to qualitative experience.

David Chalmers [10] claimed a similar view. He distinguished between the *easy* and the *hard* problem of consciousness. The easy problems are related with how the brain is able to categorize and react to stimuli, how it is able to integrates information, how it may report internal states, how it may pay attention, and so on. Instead, the hard problem

of consciousness is about the explanation of the subjective aspect of experience. This is, according to Chalmers, related to the information-processing capabilities of the brain. In fact, he suggests a *double-aspect* of information that may have a physical and a phenomenal aspect. Information processing is physical, while experience is suggested to arise from the phenomenal aspect of information.

The aforementioned philosophical views are essentially forms of *dualism*, i.e., they consider two different levels: the physical domain where the brain resides, and the mental domain where experience resides. These two levels are irreducible, i.e., it is not possible to explain experience, which belongs to mental domain, in terms of entities in the physical domain.

A related position is *epiphenomenalism*, that states that although experience is generated by processes in the brain, it is essentially useless, i.e. it does not add anything to one's behavior. Frank Jackson [29] describes this position by means of a famous mental experiment that allows us to better clarify the distinction between knowledge of a thing and subjective experience of the same thing, e.g., a color. He describes the fictional case of Mary: she is a brilliant scientist and she knows everything about colors; but, for some reason, Mary has never seen a color. In fact, she sees the world by means of a black and white monitor and she lives in a black and white room. What happens when eventually Mary leaves the room? She experiences true colors, but, by definition, nothing new is added to her knowledge of colors and of the world.

A further view is put forward by Francis Crick who, in his book *The Astonishing Hypothesis* [18], outlines a *reductionist* view about consciousness (p. 3):

The Astonishing Hypothesis is that "You," your joys and your sorrows, your memories and your ambitions, your sense of personal identity and free will, are in fact no more than the behavior of a vast assembly of nerve cells and their associated molecules. As Lewis Carroll's Alice might have phrased it: "You're nothing but a pack of neurons."

According to this view there are no levels other than the physical level. The mental level is only a way to explain the physical processes happening in the brain. As Christof Koch [31] discusses in details, a serious scientific study of consciousness should focus on the neural *correlates* of consciousness, i.e., the processes in the brain that are correlated with subjective experience. Following this line of thought, Koch proposes a general framework for studying consciousness from the point of view of neuroscience.

Another interesting approach to consciousness from the point of view of neuroscience is due to Giulio Tononi [64]. According to Tononi, consciousness is strictly related with information integration. Experience, e.g., information integration, is suggested to be a fun-

damental quantity as mass, charge, energy. Interestingly, according to Tononi, any physical system may have subjective experience to the extent that it is capable to integrate information. Thus it may be possible, at least in principle, to build a really conscious artifact.

Finally, Daniel Dennett [20] carried out a view that is often considered to be on the verge of *eliminativism*. Dennett claims that consciousness does not exist as a real process but it is a sort of *umbrella* concept that incorporates several different and sometimes unrelated physical processes. Therefore, a study of consciousness should focus on the different processes under this overarching concept. He introduced the term *heterophenomenology* as to indicate the study of consciousness from a rigorous third person point of view. According to this view, the alleged subjective experience may be studied in terms of explaining objective reports as verbal reports, PET, MRI, and so on.

14.3 Machine Consciousness

The discipline of machine consciousness aims at studying the problems of consciousness, briefly introduced in the previous section, by means of the design and implementation of conscious machines.

In recent years there has been a growing interest in machine consciousness [12–14]. The popular journal *Scientific American* in the June 2010 issue reviewed the twelve events that “will change everything”, and one of these events, considered *likely* before 2050, is machine self-awareness [23]. This interest in machine consciousness is mainly motivated by the belief that the approach based on the construction of conscious artifacts can shed new light on the many critical aspects that affect mainstream studies of cognitive sciences, philosophy of mind and also AI and neuroscience.

As a working definition of *conscious AGI agent* we may consider the following one: a conscious AGI agent is an artifact that has some form of subjective experience, and it employs it in many cognitive tasks. Therefore, a conscious AGI agent has both P-consciousness and A-consciousness.

The study of conscious AGI agents is an intriguing field of enquiry for at least two reasons. First, it takes consciousness as a real physical phenomenon to be explained at the physical level and with practical effects on behavior, a debated standpoint as stated in the previous section. Secondly, it hypothesizes the possibility to reproduce, by means of artifacts, the most intimate of mental aspects, namely, the subjective experience.

Although many argued against the possibility of a conscious agent, so far no one has conclusively argued against such a possibility. In fact, arguments which deny the possibility of machine consciousness often would deny the possibility of human consciousness as well.

Contrary to AI and functionalism, some scholars regard the functional view of the mind as insufficient to endorse the design of a conscious agent. Here, the commonly raised arguments against strong AI, such as the Searle's Chinese Room argument [56], may lose some of their strength: although most available computational systems are Turing machines that instantiate the von Neumann's blue print, other non-conventional architectures are becoming available and more are going to be designed and implemented in the near future. The recent progresses in the fields of neuromorphic computing [28] and DNA computing [1], just to mention two main cases, may open new ways towards the implementation of a conscious system. Therefore a true conscious AGI agent could be implemented in some non-conventional hardware.

Furthermore, it is an open issue whether AGI agents are reducible to a pure Turing machine view, when considered as a complex system made up by the interaction between the agent itself and its environment.

Roughly speaking, machine consciousness lies in the middle ground between the extremes of the strong biological position (i.e., only biological brains are conscious) and liberal functionalism position (i.e., any behaviorally equivalent functional systems is conscious). Machine consciousness proponents maintain that biological position is too narrow and yet they concede that some kind of physical constraints will be unavoidable (hence no multiple feasibility) in order to build a conscious agent.

A common objection to machine consciousness emphasizes the fact that biological entities may have unique characteristics that cannot be reproduced in artifacts. If this objection is true, machine consciousness may not be feasible. However, this contrast between biological and artificial entities has often been over exaggerated, especially in relation to the problems of consciousness. So far, nobody was able to satisfactorily prove that the biological entities may have characteristics that can not be reproduced in artificial entities with respect to consciousness. Instead, at a meeting on machine consciousness in 2001 organized by the Swartz Foundation at Cold Spring Harbor Laboratories, the conclusion of Christof Koch¹ was that:

we know of no fundamental law or principle operating in this universe that forbids the existence of subjective feelings in artifacts designed or evolved by humans.

¹http://www.theswartzfoundation.org/abstracts/2001_summary.asp

Consider the well-known distinction introduced by Searle [56] related to *weak AI* and *strong AI*. According to weak AI position, a computer may be an useful tool for the study of the mind but it is not itself a mind, while for the strong AI position, also known as *computationalism*, an appropriately programmed computer is a real mind itself and it has real effective cognitive capabilities. Recently, some scholars working in the field of machine consciousness emphasized the behavioral role of consciousness to avoid the problem of phenomenal experience (see e.g. Seth [58]). They, paraphrasing Searle, suggested that it is possible to distinguish between weak machine consciousness and strong machine consciousness. The former approach deals with AGI agents which behaved as if they were conscious, at least in some respects. Such a view avoids any commitment to the hard problem of consciousness. The latter approach explicitly deals with the possibility of designing and implementing AGI agents capable of real conscious experience. This distinction is also mirrored in the previously discussed distinction between P-consciousness and A-consciousness, where, roughly speaking, a weak AGI conscious agent corresponds to an agent that has A-consciousness only, while a strong AGI conscious agent corresponds to an agent that has P-consciousness and A-consciousness.

Although the distinction between weak and strong AGI agents may set a useful working hypothesis, it may suggest a misleading view. Setting aside experience, i.e., P-consciousness, something relevant could be missed for the understanding of cognition. Skipping the hard problem of consciousness could be not a viable option in the business of making real conscious AGI agents.

Further, the distinction between weak and strong AGI systems may be misleading because it mirrors a dichotomy between true conscious agents and *as if* conscious agents. Yet, human beings are conscious and there is evidence that most animals exhibiting behavioral signs of consciousness are phenomenally conscious. It is a fact that intelligent human beings have phenomenal consciousness. They experience pains, pleasures, colors, shapes, sounds, and many more other phenomena. They feel emotions, feelings of various sort, bodily and visceral sensations. Arguably, they also have experiences of thoughts and of some cognitive processes.

In summary, it would be very strange whether natural selection had gone at such great length to provide us with consciousness if there was a way to get all the advantages of a conscious being like intelligence without actually producing it. Thus we cannot help but wonder whether it is possible to design an AGI agent without dealing with the hard problem of consciousness.

14.4 Agent's Body

A crucial aspect for a conscious AGI agent is the relationships with its own body, as pointed out by Metzinger [42]. Although it cannot be underestimated the importance of the interface between a robot and its environment, as well as the importance of an efficient body, it is far from clear whether this aspect is intrinsically necessary to the occurrence of consciousness. Having a body does not seem to be a sufficient condition for consciousness. Arguably, it could be a necessary condition.

Apart from intuitive cases, when is an agent truly embodied? On one hand, there is no such a thing as a *non-embodied agent*, since even the most classic AI system has to be implemented as a set of instructions running inside a physical device as a computer. On the other hand, a complex and sophisticated robot such as ASIMO by Honda² is far from being conscious, as it is actually controlled by carefully hard-wired behavioral rules.

There are many biological agents that would apparently score very well on embodiment but yet do not seem good candidate for consciousness. Take insects for instance. They show impressive morphological structures that allow them to perform outstandingly well without very sophisticated cognitive capabilities.

It should be noticed that having a body could also influence higher cognitive processes more strictly related with intelligence. In their seminal and highly debated book, Lakoff and Núñez [35] discuss in detail and with many examples how mathematical concepts and reasonings could be deeply rooted in the human body and in its interactions with environment.

In this field, an effective robot able to build an internal model of its own body and environment has been proposed by Holland and Goodman [25]. The system is based on a neural network that controls a Khepera minirobot and it is able to simulate perceptual activities in a simplified environment. Following the same principles, Holland *et al.* [26] discuss the robot CRONOS, a very complex *anthropomimetic* robot whose operations are controlled by SIMNOS, a 3D simulator of the robot and of its environment. ECCEROBOT³ is the new complex incarnation of this anthropomimetic robot.

Chella and Macaluso [11] present a model of robot perception based on a comparison loop between the actual and the expected robot sensory data generated by a 3D model of the robot body. The perception loop process is operating in *CiceRobot*, a robot that offered guided tours at the Archaeological Museum of Agrigento, Italy.

²<http://asimo.honda.com>

³<http://eccerobot.org>

Shanahan [59, 60] discusses a cognitive robot architecture based on the Global Workspace Theory [2] in which the planning operations are performed by simulating the interactions of the robot with the external environment. He also discusses about implementations of the architecture based on different kinds of neural networks.

While the previously discussed robots start with their own basic model of the body and they had to learn how to control it, Bongard *et al.* [9] discuss a starfish robot which is able to build a model of its own body from scratch. The body model is then employed to make the robot walk. The body model changes consequently if some damages occur to the robot's real body.

In summary, the notion of embodiment is far more complex than the simple idea of controlling a robot's body. It refers to the kind of development and causal processes engaged between a robot, its body, and its environment. This step appears to be unavoidable for a truly conscious AGI agent.

14.5 Interactions with the Environment

Besides having a body, a conscious AGI agent needs to be situated in an environment. Yet the necessity of *situatedness* is not totally uncontroversial. For instance, authors like Metzinger [41] argued that consciousness needs a purely virtual inner world created inside a system which, to all respects, lacks any direct contact with the environment.

If consciousness requires a body in the environment, real or simulated one, we should be able to point out what is to be situated. What kind of architecture and individual history is sufficient for being situated? A classic example of interaction with the environment is the case of passive dynamic walker, extensively discussed by Pfeifer and Bongard [51].

A fruitful approach towards situatedness is represented by those implementations that outsource part of the cognitive processes to the environment and explicitly consider the agent as a part of the environment. An interesting idea in this field is the *morphological computation* introduced by Paul [49]. She built different robots shaped in such a way so that the robot is able to perform simple logic operations as a XOR by means of their interactions with environment.

About the interaction between body and environment, there are two powerful conceptual attractors in the discussion on consciousness. They are going to exert their strength in the machine consciousness arena, too. Where is the mind and its content located? Inside or

outside the body of the agent? So far, neither options proved entirely satisfactory and the debates keeps running.

It would be very simple if we could locate consciousness inside of the body of the agent and thus inside our conscious AGI robots. So, the mind should somehow depend on what takes place exclusively inside of the body of the robot. Therefore, the mental world must be inside of the agent from the beginning or it must be concocted inside. This position can broadly be labeled as *internalism*.

However, such a view is not fully convincing since the robot's mental states (broadly speaking) are about something that often appears as being external to the body. In fact, mental states typically address external states of affairs (whether they are concepts, thoughts, percepts, objects, events).

In fact, consciousness refers to the external world. Then, we could reframe our model of the agent's mind such as to include also the external world in the agent's mind. Such an *externalist* change in our perspective would endorse those views that consider the sense of the body and the interaction with external environment as main conditions for a conscious AGI agent [15, 53–55].

Initial suggestions have been presented on how to implement an externalist agent. The most relevant ones are due to O'Regan and Noë [46–48]. They discuss the *enactive* process of visual awareness as based on sensorimotor contingencies. Following this approach, an AGI agent should be equipped by a pool of sensorimotor contingencies so that entities in the environment activate the related contingencies that define the interaction schemas between the robot and the entity itself.

Some contingencies may be pre-programmed in the agent by design (phylogenetic contingencies), but during the working life, the agent may acquire novel contingencies and therefore novel way of interacting with the environment. Moreover, the agent should acquire new ways of mastery, i.e., new ways to use and combine contingencies, in order to generate its own goal tasks and motivations (ontogenetic contingencies). A mathematical analysis of the enactive theory applied to a simple robot in a simulated environment is presented in Philipona *et al.* [52].

Manzotti and Tagliasco [40] discuss the theory of *enlarged mind* as an externalist theory covering the phenomenal and the functional aspects of consciousness with respect to a robot vision system. Following this line, Manzotti [39] analyzed the human and robotic conscious perception as a process that unifies the activity in the brain and the perceived events in the external world.

In summary, consciousness involves some kind of developmental integration with the environment such that what the robot is and does is a result of the deep interactions with the environment. A conscious AGI agent should be an agent that changes in some non-trivial way as a result of its tight coupling with the environment.

14.6 Time

Conscious experience is located in time. We experience the flow of time in a characteristic way which is both continuous and discrete. On one hand, there is the flow of time in which we float seamlessly. On the other hand, our cognitive processes require time to produce conscious experience and they are located in time. Surprisingly, there is evidence from the famous Libet's studies [37] showing that we are visually aware of something only half a second after our eyes have received the relevant information.

The classic notion of time from physics fits very loosely with our experience of time. Only the instantaneous present is real. Everything has to fit in such Euclidean temporal point. For instance, speed is nothing more than the value of a derivative and can be defined at every instant. We are expected to occupy only an ever-shifting temporal point with no width. Such an instantaneous present cannot accommodate the long lasting and content-rich conscious experience of present.

Neuroscience faces similar problems. According to the neural view of the mind, every cognitive and conscious process is instantiated by patterns of neural activity. This apparently innocuous hypothesis hides a problem. If a neural activity is distributed in time (as it has to be since neural activity consists in temporally distributed series of spikes), there must be some strong sense in which something taking place in different instants of time belong to the same cognitive or conscious process.

But what glues together the first and the last spike of a neural activity leading a subject to perceive a face? Simply suggesting that they occur inside the same window of neural activity is like explaining a mystery with another mystery. What is a temporal window? And how does it fit with our physical picture of time? Indeed, it seems to be at odds with the instantaneous present of physics.

In the case of AGI agents, this issue is extremely counterintuitive. For instance, let us suppose that a certain computation is identical with a given conscious experience. What would happen if we purposefully slow down the taking place of the same computation, as imagined in the famous science fiction book *Permutation City* [21]? Certainly, we can

envisage an artificial environment where the same computation is performed at an altered time (for instance we could simply slow down the internal clock of such a machine). Would the AGI agent have identical conscious experience but spread in a longer span of time?

Some initial interesting experiments have been described by Madl *et al.* [38] in the framework of the LIDA system [3]. They report experiments that simulate the timing of the cognitive cycle of LIDA; interestingly, the performances of the system are similar to humans just when the simulated timing of the cognitive cycle is comparable with the human timing.

A related issue is the problem of the present. As in the case of brains, what defines a temporal window? Why are certain states part of the present? Does it depend on certain causal connections with behavior or is it the effect of some intrinsic property of computations? It is even possible that we would need to change our basic notion of time.

14.7 Free Will

Another issue that does not fit with the picture of a classic AI system is the fact that a conscious AGI agent should be capable of a unified will assumed as free (see, among others, Wegner [67] and Walter [66]).

A classic argument against free will in human and hence in an artifact is the following [30]. If a subject is nothing but the micro-particles constituting it (and their state), all causal powers are drained by the smallest constituents. If the argument holds, there will be no space left for any high level causal will. All reality ought to reduce causally to what is done by the micro-particles who would be in total charge of what happens. No causation would be possible and no space would remain for the will of a subject to interfere on the course of events.

Yet, we have a strong intuition that we are capable of willing something and that our conscious will is going to make a difference in the course of events. Many philosophers strongly argued in favor of the efficacy of conscious will.

Another threat to free will comes from previously cited Libet's studies [37], according to which we are conscious of our free choices only after 300 ms our brain has made them. Although Libet left open the possibility that we can *veto* the deliberations of our brains, there is an open debate about the interpretation of these experimental results.

In short, an open problem is whether a complex agent as a whole can have any kind of causal power over its constituents. Since consciousness seems to be strongly related

with the agent as a whole, we need some theory capable of addressing the relation between wholes and parts.

For an AGI agent, this issue is difficult as ever. The classic approach and several design strategies (from the traditional *divide et impera* to sophisticated object-oriented programming languages) suggest to conceive AGI agents as made of separate and autonomous modules. Then, AGI agents would be classic examples of physical systems where the parts completely drain the causal power of the system as a whole. From this point of view, these systems would be completely unsuited to endorse a conscious will. However, there are some possible approaches that can provide a new route.

A possible approach [64] is based on recent proposals that stressed the connectivity between elementary computational units of an agent. According to such proposals it is possible to implement networks of computational units whose behavior is not reducible to any part of the network, but rather it stems out of the integrated information of the system as a whole.

Another approach [40] stresses the roles of suitable feedback loops that could do more than classic control feedbacks. Here, the idea is to implement machines capable of taking into account their whole history, also in a summarized way, in order to decide what to do. Thus, the behavior of an AGI agent would be the result of its past history as a whole. There would not be separate modules dictating what the system has to do, but rather the past history as a whole would have effect in every choice.

14.8 Experience

The more complex problem for consciousness is: how can a physical system like an AGI agent produce something similar to our subjective experience? During a jam session, the sound waves generated by the musical instruments strike our ears and we experience a sax solo accompanied by bass, drums and piano. At sunset, our retinas are struck by rays of light and we have the experience of a symphony of colors. Swallow molecules of various kinds and, therefore, feel the taste of a delicious wine.

It is well known that Galileo Galilei suggested that smells, tastes, colors and sounds do not exist outside the body of a conscious subject (*the living animal*). Thus experience would be created by the subject in some unknown way.

A possible hypothesis concerns the separation between the domain of experience, namely, the subjective content, and the domain of objective physical events. This hypoth-

esis is at the basis of science itself. The claim is that physical reality can be adequately described only by the quantitative point of view in a *third person* perspective while ignoring any qualitative aspects. After all, in a physics textbook there are many mathematical equations that describe a purely quantitative reality. There is no room for quality content, feelings or emotions.

Yet many scholars (see Strawson [63]) have questioned the validity of such a distinction as well as the degree of real understanding of the nature of the physical world.

Whether the mental world is a special construct generated by some feature of the nervous systems of mammals is still an open question, as briefly summarized in Section 14.2. It is fair to stress that there is neither empirical evidence nor theoretical arguments supporting such a view. In the lack of a better theory, we could also take into consideration the idea inspired by the previously mentioned *externalism* view that the physical world comprehends also those features that we usually attribute to the mental domain. A physicalist must be held that if something is real, and we assume consciousness is real, it has to be physical. Hence, in principle, a device can envisage it.

In the case of AGI agents, how is it possible to overcome the distinction between function and experience? As previously outlined, a typical AGI agent is made up by a set of interconnected modules, each operating in a certain way. How the operation of some or all of the interconnected modules should generate conscious experience? However, the same question could be transferred to the activity of neurons. Each neuron, taken alone, does not work differently from a software module or a chip. But it could remain a possibility: it is not the problem of the physical world, but of our theories of the physical world. AGI systems are part of the same physical world that produce consciousness in human subjects, so they may exploit the same properties and characteristics that are relevant for conscious experience.

In this regard, Giulio Tononi [65] proposed the information integration theory, briefly introduced in Section 14.2. According to Tononi, the degree of conscious experience is related with the amount of *integrated information*. The primary task of the brain is to integrate information and, noteworthy, this process is the same whether it takes place in humans or in artifacts like AGI agents. According to this theory, conscious experience has two main characteristics. On the one side, conscious experience is differentiated because the potential set of different conscious states is huge. On the other side, conscious experience is integrated; in fact a conscious state is experienced as a single entity. Therefore, the substrate of conscious experience must be an integrated entity able to differentiate among a

big set of different states and whose informational state is greater than the sum of the informational states of the component sub-entities. Tononi provides a formal methodology for the static case [64] and for the dynamic case [4, 5] of the theory. According to this theory, Koch and Tononi [32] discuss a potential new Turing test based on the integration of information: artificial systems should be able to mimic the human being not in language skills (as in the classic version of Turing test), but rather in the ability to integrate information from different sources, for example in the generation of the explanation of a picture.

We must emphasize the fact that the implementation of a true AGI agent able to perform information integration is a real technological challenge. In fact, as previously stated, the typical software engineering techniques for the construction of AGI agents are essentially based on the design of a system through the decomposition of the system into easier subsystems. Each subsystem then will communicate with the others subsystems through well-defined interfaces so that the interaction between the subsystems happen in a controlled way. Tononi's theory rather requires maximum interaction between the subsystems to allow an effective integration. Therefore new engineering techniques are required to design conscious AGI agents.

Information integration theory could represent a first step towards a theoretically well-founded approach to machine consciousness. The idea of being able to find the *consciousness equations* that, like Maxwell's equations in physics, explains consciousness in living beings and in artifacts is a kind of ultimate goal for scholars of consciousness.

14.9 Creativity

Can an AGI system be so creative to the point that its creations could be indistinguishable from those of a human being? According to Sternberg [62], creativity is the ability to produce something that is new and appropriate. The result of a creative process is not reducible to some sort of deterministic reasoning. No creative activity seems to identify a specific chain of activity, but an emerging *holistic* result [33].

Therefore, a creative AGI should be able to generate novel artifacts not by following preprogrammed instructions, as in typical industrial robots, but on the contrary by means of a real creative act.

The problem of creativity in artifacts has been widely debated for example in the field of automatic music composition. The software system EMI by David Cope [16] produces impressive results: even for an experienced listener it is difficult to distinguish musical

compositions created by these programs from those ones created by a human composer. There is no doubt that these systems may capture some main aspects of the creative process, at least in music.

However, one may wonder if an AGI system can actually be creative without being conscious. In this regard, Damasio [19] suggests a close connection between consciousness and creativity. Also Cope [17] discusses the relationship between consciousness and creativity. Although he does not take a clear position on this matter, he seems to favor the view according to which consciousness is not necessary for creative process. In fact, Cope asks if a creative agent should need to be aware of being creating something and if it needs to experience its own creations.

According to Boden [8], the argument of consciousness is typically adopted to support the thesis according to which an artificial agent can never be conscious and therefore it can never be really creative. In this respect, a conscious AGI system may be a breakthrough towards a real creative agent.

The relationship between consciousness and creativity is difficult and complex. On the one side some scholars claim the need of awareness of the creative act. On the other side, it is suspected that, similarly to intelligence, many processes that are necessary for the creative act may happen in the absence of consciousness.

However it is undeniable that consciousness is closely linked with the broader unpredictable and less “automatic” forms of creativity. In addition, we could distinguish between the mere production of new combinations and the aware creation of new content. If the wind would create (like the monkeys on a keyboard) a form which is indistinguishable from the *Pieta* by Michelangelo, it would be a creative act? Many authors would debate this argument.

Let us consider as an example the design of an AGI system able to improvise jazz. The aspect of corporeality seems to be fundamental to the jazz performance. Auditory feedback is not sufficient to explain the characteristics of a performance: making music is essentially a full body activity [34]. The movement of the hands on the instrument, the touch and the strength needed for the instrument to play, the vibrations of the instrument propagated through the fingers of the player, the vibration of the air perceived by the player’s body, are all examples of feedbacks guiding the musician during the performance. The player receives different types of bodily feedbacks, e.g., through the receptors of the skin and through the receptors of the tendons and muscles.

In addition to having a body, an artist, during a jam session, is typically situated in a group where she has a continuous exchange of information. The artist receives and provides continuous feedbacks with the other players of the group, and sometimes even with the audience, in the case of live performances.

The classical view, often theorized in textbooks of jazz improvisation, suggests that during a solo, the player follows his own musical path largely made up by a suitable musical sequence of previously learned patterns. This is a partial view of an effective jazz improvisation. Undoubtedly, the musician has a repertoire of musical patterns, but she is also able to freely deviate from its path depending on her past experience and sensitivity, and according to the feedback she receives from other musicians or the audience, for example from suggestions from the rhythm section or due to signals of appreciation from the listeners.

Finally, an AGI system aware of its jazz improvisation should be able to integrate during time the information generated by the instrument, the instruments of its group as well as information from its own body.

Therefore, many of the challenges previously reviewed for a conscious AGI agent, as the agent's body, the interaction with environment, the sense of time, the capability to take free decisions and to have experiences, are all challenges for a truly creative AGI system.

14.10 Conclusions

The list of problems related with machine consciousness that have not been properly treated here is long: the problem of meaning, the generation of mental images, the problem of representation, the problem of higher order consciousness, and so on. These are issues of great importance for the creation of a conscious AGI agent, although some of them may overlap in part with the arguments discussed above.

The sense of the body of an agent, its interactions with the environment, the problem of agent's time, the free will of the agent and the capability for the agent to have some form of experience and creativity are all issues relevant to the problem of building a conscious AGI agent.

Machine consciousness is, at the same time, a theoretical and technological challenge that forces us to deal with old problems by means of new and innovative approaches. It is possible that research in machine consciousness will push to re-examine in a novel way

many hanging threads from classic AI, as Haugeland summarizes in a provocative way in the quotation at the beginning of this review.

Bibliography

- [1] M. Amos, *Theoretical and Experimental DNA Computation*. vol. XIII, *Theoretical Computer Science*, (Springer, Heidelberg, 2005).
- [2] B. Baars, *A Cognitive Theory of Consciousness*. (Cambridge University Press, Cambridge, MA, 1988).
- [3] B. Baars and S. Franklin, Consciousness is computational: The LIDA model of global workspace theory, *International Journal of Machine Consciousness*. **1**(1), 23–32, (2009).
- [4] D. Balduzzi and G. Tononi, Integrated information in discrete dynamical systems: Motivation and theoretical framework, *PLoS Computational Biology*. **4**(6), e1000091, (2008).
- [5] D. Balduzzi and G. Tononi, Qualia: The geometry of integrated information, *PLoS Computational Biology*. **5**(8), e1000462, (2009).
- [6] S. Blackmore, *Consciousness An Introduction (second edition)*. (Hodder Education, London, 2010).
- [7] N. Block, On a confusion about a function of consciousness, *Behavioral and Brain Sciences*. **18**(02), 227–247, (1995).
- [8] M. Boden, *The Creative Mind: Myths and Mechanisms - Second edition*. (Routledge, London, 2004).
- [9] J. Bongard, V. Zykov, and H. Lipson, Resilient machines through continuous self-modeling, *Science*. **314**, 1118–1121, (2006).
- [10] D. Chalmers, *The Conscious Mind*. (Oxford University Press, New York, NY, 1996).
- [11] A. Chella and I. Macaluso, The perception loop in *CiceRobot*, a museum guide robot, *Neuro-computing*. **72**, 760–766, (2009).
- [12] A. Chella and R. Manzotti, Eds., *Artificial Consciousness*. (Imprint Academic, Exeter, UK, 2007).
- [13] A. Chella and R. Manzotti, Machine consciousness: A manifesto for robotics, *International Journal of Machine Consciousness*. **1**(1), 33–51, (2009).
- [14] A. Chella and R. Manzotti. Artificial consciousness. In eds. V. Cutsuridis, A. Hussain, and J. Taylor, *Perception-Action Cycle: Models, Architectures, and Hardware*, vol. 1, *Springer Series in Cognitive and Neural Systems*, pp. 637–671. Springer, New York, NY, (2011).
- [15] A. Clark, *Supersizing The Mind*. (Oxford University Press, Oxford, UK, 2008).
- [16] D. Cope, Computer modeling of musical intelligence in EMI, *Computer Music Journal*. **16**(2), 69–83, (1992).
- [17] D. Cope, *Computer Models of Musical Creativity*. (MIT Press, Cambridge, MA, 2005).
- [18] F. Crick, *The Astonishing Hypothesis*. (Simon and Schuster Ltd, 1994).
- [19] A. Damasio, *The Feeling of What Happens: Body and Emotion in the Making of Consciousness*. (Harcourt Brace, New York, 1999).
- [20] D. Dennett, *Consciousness Explained*. (Little, Brown, New York, 1991).
- [21] G. Egan, *Permutation City*. (Weidenfeld Military, 1994).
- [22] H. Gardner, *Frames of Mind: The Theory of Multiple Intelligences*. (Basic Books, New York, NY, 1983).
- [23] L. Greenemeier, Machine self-awareness, *Scientific American*. **302**(6), 28–29 (June, 2010).
- [24] J. Haugeland, *Artificial Intelligence: The Very Idea*. (MIT Press, Bradford Book, Cambridge, MA, 1985).

- [25] O. Holland and R. Goodman, Robots with internal models - a route to machine consciousness?, *Journal of Consciousness Studies*. **10**(4-5), 77–109, (2003).
- [26] O. Holland, R. Knight, and R. Newcombe. A robot-based approach to machine consciousness. In eds. A. Chella and R. Manzotti, *Artificial Consciousness*, pp. 156–173. Imprint Academic, (2007).
- [27] T. Honderich, Ed., *The Oxford Companion To Philosophy - New edition*. (Oxford University Press, Oxford, UK, 2005).
- [28] G. Indiveri and T. Horiuchi, Frontiers in neuromorphic engineering, *Frontiers in Neuroscience*. **5**, fnins.2011.00118, (2011).
- [29] F. Jackson, Epiphenomenal qualia, *The Philosophical Quarterly*. **32**(127), 127–136, (1982).
- [30] J. Kim, *Mind in a Physical World*. (MIT Press, Bradford Books, Cambridge, MA, 2000).
- [31] C. Koch, *The Quest for Consciousness*. (Roberts and Co., Engewood, CO, 2004).
- [32] C. Koch and G. Tononi, Can machines be conscious?, *IEEE Spectrum*. pp. 47–51 (June, 2008).
- [33] A. Koestler, *The Act of Creation*. (Penguin Books, New York, 1964).
- [34] J. Krueger, Enacting musical experience, *Journal of Consciousness Studies*. **16**, 98–123, (2009).
- [35] G. Lakoff and Nunez, *Where Mathematics Comes From: How the Embodied Mind Brings Mathematics into Being*. (Basic Books, New York, 2000).
- [36] S. Legg and M. Hutter. A collection of definitions of intelligence. In eds. B. Goertzel and P. Wang, *Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms*, pp. 17–24, Amsterdam, The Netherlands, (2006). IOS Press.
- [37] B. Libet, *Mind Time*. (Harvard University Press, Cambridge, MA, 2004).
- [38] T. Madl, B. Baars, and S. Franklin, The timing of the cognitive cycle, *PLoS ONE*. **6**(4), e14803, (2011).
- [39] R. Manzotti, A process oriented view of conscious perception, *Journal of Consciousness Studies*. **13**(6), 7–41, (2006).
- [40] R. Manzotti and V. Tagliasco, From behaviour-based robots to motivation-based robots, *Robotics and Autonomous Systems*. **51**, 175–190, (2005).
- [41] T. Metzinger, *Being No One*. (MIT Press, Bradford Books, Cambridge, MA, 2003).
- [42] T. Metzinger, *The Ego Tunnel*. (Basic Books, New York, 2009).
- [43] G. Miller, What is the biological basis of consciousness, *Science*. **309**, 79 (July, 2005).
- [44] T. Nagel, What is it like to be a bat?, *The Philosophical Review*. **83**(4), 435–450, (1974).
- [45] U. Neisser, G. Boodoo, T. Bouchard, A. Wade Boykin, S. Ceci, D. Halpern, J. Loehlin, R. Perloff, R. Sternberg, and S. Urbina, Intelligence: Knowns and unknowns, *American Psychologist*. **51**(2), 77–101, (1996).
- [46] A. Noë, *Action in Perception*. (MIT Press, Bradford Books, Cambridge, MA, 2004).
- [47] J. O'Regan, *Why Red Doesn't Sound Like a Bell*. (Oxford University Press, Oxford, UK, 2011).
- [48] J. O'Regan and A. Noë, A sensorimotor account of vision and visual consciousness, *Behavioral and Brain Sciences*. **24**, 939–1031, (2001).
- [49] C. Paul, Morphological computation a basis for the analysis of morphology and control requirements, *Robotics and Autonomous Systems*. **54**, 619–630, (2006).
- [50] C. Pennachin and B. Goertzel. Contemporary approaches to artificial general intelligence. In eds. B. Goertzel and C. Pennachin, *Artificial General Intelligence*, pp. 1–30. Springer, Berlin, (2007).
- [51] R. Pfeifer and J. Bongard, *How the Body Shapes the Way We Think*. (MIT Press, Bradford Books, Cambridge, MA, 2007).
- [52] D. Philipona, J. O'Regan, and A. Noë, Is there something out there? Inferring space from sensorimotor dependencies, *Neural Computation*. **15**, 2029–2049, (2003).
- [53] T. Rockwell, *Neither Brain nor Ghost*. (MIT Press, Cambridge, MA, 2005).
- [54] M. Rowlands, *Externalism – Putting Mind and World Back Together Again*. (McGill-Queen's University Press, Montreal and Kingston, 2003).

-
- [55] M. Rowlands, *The New Science of the Mind*. (MIT Press, Bradford Books, Cambridge, MA, 2010).
- [56] J. Searle, Minds, brains, and programs, *Behavioral and Brain Sciences*. **3**(3), 417–457, (1980).
- [57] J. Searle, Consciousness, *Annual Review of Neuroscience*. **23**, 557–578, (2000).
- [58] A. Seth, The strength of weak artificial consciousness, *International Journal of Machine Consciousness*. **1**(1), 71–82, (2009).
- [59] M. Shanahan, A cognitive architecture that combines internal simulation with a global workspace, *Consciousness and Cognition*. **15**, 433–449, (2006).
- [60] M. Shanahan, *Embodiment and the Inner Life*. (Oxford University Press, Oxford, UK, 2010).
- [61] R. Sternberg, *Beyond IQ: A Triarchic Theory of Human Intelligence*. (Cambridge University Press, 1985).
- [62] R. Sternberg, Ed., *Handbook of Creativity*. (Cambridge University Press, Cambridge, UK, 1998).
- [63] G. Strawson, Does physicalism entail panpsychism?, *Journal of Consciousness Studies*. **13**, 3–31, (2006).
- [64] G. Tononi, An information integration theory of consciousness, *BMC Neuroscience*. **5**(42), (2004).
- [65] G. Tononi, Consciousness as integrated information: a provisional manifesto, *Biological Bulletin*. **215**, 216–242, (2008).
- [66] H. Walter, *Neurophilosophy of Free Will*. (MIT Press, Bradford Books, Cambridge, MA, 2001).
- [67] D. Wegner, *The Illusion of Conscious Will*. (MIT Press, Bradford Books, Cambridge, MA, 2002).

Chapter 15

Human and Machine Consciousness as a Boundary Effect in the Concept Analysis Mechanism

Richard Loosemore

Mathematical and Physical Sciences, Wells College, Aurora, NY 13026, U.S.A.

rloosemore@wells.edu

To solve the hard problem of consciousness we observe that any cognitive system of sufficient power must get into difficulty when it tries to analyze consciousness concepts, because the mechanism that does the analysis will “bottom out” in such a way as to make the system declare these concepts to be both real and ineffable. Rather than use this observation to dismiss consciousness as an artifact, we propose a unifying interpretation that allows consciousness to be explicable at a meta level, while at the same time being mysterious and inexplicable on its own terms. This implies that science must concede that there are some aspects of the world that deserve to be called “real”, but which are beyond explanation. We conclude that some future thinking machines will, inevitably, have the same subjective consciousness that we do. Some testable predictions are derived from this theory.

15.1 Introduction

The scope of this chapter is defined by the following questions:

- When we use the term “consciousness” what exactly are we trying to talk about?
- How does consciousness relate to the functioning of the human brain?
- If an artificial general intelligence (*AGI*) behaved as if it had consciousness, would we be justified in saying that it was conscious?
- Are any of the above questions answerable in a scientifically objective manner?

The ultimate goal is to answer the third question, about machine consciousness, but in order to make meaningful statements about the consciousness of artificial thinking systems, we need first to settle the question of what consciousness is in a human being. And before

we can answer that question, we need to be clear about whatever it is we are trying to refer to when we use the term “consciousness”. Finally, behind all of these questions there is the problem of whether we can explain any of the features of consciousness in an objective way, without stepping outside the domain of consensus-based scientific enquiry and becoming lost in a wilderness of subjective opinion.

To anyone familiar with the enormous literature on the subject of consciousness, this might seem a tall order. But, with due deference to the many intellectual giants who have applied themselves to this issue without delivering a widely accepted solution, I would like to suggest that the problem of consciousness is actually much simpler than it appears on the surface. What makes it seem difficult is the fact that the true answer can only be found by asking a slightly different question than the one usually asked. Instead of asking directly for an explanation of the thing, we need to ask why we have such peculiar difficulty stating what exactly the thing is. Understanding the nature of the *difficulty* reveals so much about the problem that the path to a solution then becomes clear.

15.1.1 *The Hard Problem of Consciousness*

One of the most troublesome aspects of the literature on the problem of consciousness is the widespread confusion about what exactly the word “consciousness” denotes. In his influential book on the subject, Chalmers [2] resolved some of this confusion when he drew attention to the fact that the word is often used for concepts that do not contain any deep philosophical mystery. These straightforward senses include:

- The ability to introspect or report mental states. A fly and a human can both jump out of the way of a looming object, but a human can consciously think and talk about many aspects of the episode, whereas the fly simply does not have enough neural machinery to build internal models of its action. By itself, though, this ability to build internal models is not philosophically interesting.
- Someone who is asleep can be described as not being conscious, but in this case the word is only used for a temporary condition, not a structural incapacity.
- We occasionally say that a person *consciously* did something, when what we really mean is that the person did it *deliberately*.
- If a person *knows* a fact we sometimes say that they are *conscious* of the fact.

In contrast to these senses (and others in a similar vein), there is one meaning for the word “consciousness” that is so enigmatic that it is almost impossible to express. This

is the subjective quality of our experience of the world. For example, the core thing that makes our sensation of redness different from our sensation of greenness, but which we cannot talk about with other people in any kind of objective way. These so-called *qualia*—the quality of our tastes, pains, aches, visual and auditory imagery, feelings of pleasure and sense of self—are all experiences that we can talk about with other people who say they experience them, but which we cannot describe to a creature that does not claim to experience them. When a person who is red-green color blind asks what difference they would see between red and green if they had a full complement of color receptors, the only answer we can give is “It is like the difference between your color red/green and the color blue, only different.” To the extent that this answer leaves out something important, that omitted thing is part of the problem of consciousness.

The terms “phenomenology” or “phenomenal consciousness” are also used to describe these core facts about being a conscious creature. This is in contrast to the *psychology* of being a thinking creature: we can analyze the mechanisms of thought, memory, attention, problem solving, object recognition, and so on, but in doing so we still (apparently) say nothing about what it is like to be a thing that engages in cognitive activity.

One way to drive this point home is to notice that it is logically possible to conceive of a creature that is identical to one of us, right down to the last atom, but which does not actually experience this inner life of the mind. Such a creature—a philosophical zombie—would behave as if it did have its own phenomenology (indeed its behavior, *ex hypothesi*, would be absolutely identical to its normal twin) but it would not experience any of the subjective sensations that we experience when we use our minds. It can be argued that if such a thing is logically possible, then we have a duty to explain what it means to say that there is a thing that we possess, or a thing that is an attribute of what we are, that marks the difference between one of us and our zombie twin [1, 5]. If it is conceivable that a thing could be absent, then there must be a “thing” there that can be the subject of questions. That thing—absent in the zombie but present in ourselves—is consciousness.

In order to make a clear distinction between the puzzle of this kind of consciousness, versus the relatively mundane senses of the word listed earlier, Chalmers [2] labeled this the “hard problem” of consciousness. The other questions—for example, about the neural facts that distinguish waking from sleeping—may be interesting in their own right, but they do not involve deep philosophical issues, and should not be confused with the hard problem.

Many philosophers would say that these subjective aspects of consciousness are so far removed from normal science that if anyone proposed an objective, scientific explanation for the hard problem of consciousness they would be missing the point in a quite fundamental way. Such an explanation would have to start with a bridge between the ideas of *objective* and *subjective*, and since the entire scientific enterprise is, almost by definition, about explaining objectively verifiable phenomena, it seems almost incoherent to propose a scientific (i.e. non-subjective) explanation for consciousness (which exists only in virtue of its pure subjectivity).

The story so far is that there is confusion in the literature about the exact definition of consciousness because it is ambiguous between several senses, with only one of the senses presenting a deep philosophical challenge. This ambiguity is only part of the confusion, however, because there are many cases where a piece of research begins by declaring that it will address the hard problem (for example, there is explicit language that refers to the mystery of subjective experience), but then shifts into one of the other senses, without touching the central question at all. This is especially true of neuroscience studies that purport to be about the “neural correlate of consciousness”: more often than not the actual content of the study turns out to devolve on the question of which neural signals are present when the subject is awake, or engaging in intentional acts, and so on.

The eventual goal of the present chapter is to answer questions about whether machines can be said to be conscious, so it should be clear that the hard problem, and only the hard problem, is at issue here. Knowing that an artificial intelligence has certain circuits active when it is attending to the world, but inactive when it is not, is of no relevance. Similarly, if we know that wires from a red color-detection module are active, this tells us the cognitive level fact that the machine is detecting red, but it does not tell us if the machine is experiencing a sensation of redness, in anything like the way that we experience redness.

It is this subjective experience of redness—as well as all the other aspects of phenomenology—that we need to resolve. What does it mean to say that a human experiences a subjective phenomenal consciousness, and is it possible to be sure that an artificial intelligence of sufficient completeness would (or would not) have the same phenomenal experience?

15.1.2 A Problem within the Hard Problem

We now focus on the fact that even after we separate the hard problem of consciousness from all the non-hard, or easy problems, there is still some embarrassing vagueness in

the definition of the hard problem itself. The trouble is that when we try to say what we mean by the hard problem, we inevitably end up by saying that *something is missing* from other explanations. We do not say “Here is a thing to be explained,” we say “We have the feeling that there is something that is not being addressed, in any psychological or physical account of what happens when humans (or machines) are sentient.” It seems impossible to articulate what we actually want to see explained—we can only say that we consider all current accounts (as well as every conceivable future account) of the mechanisms of cognition to be not relevant to phenomenology.

The situation can perhaps be summarized in the form of a dialectic:

Skeptic: *If you give us an objective definition for terms such as “consciousness” and “phenomenology,” then and only then can we start to build an explanation of those things; but unless someone can say exactly what they mean by these terms, they are not really saying anything positive at all, only complaining about some indefinable thing that ought to be there.*

Phenomenologist: *We understand your need for an objective definition for the thing that we want explained, but unfortunately that thing seems to be intrinsically beyond the reach of objective definition, while at the same time being just as deserving of explanation as anything else in the universe. The difficulty we have in supplying an objective definition should not be taken as grounds for dismissing the problem—rather, this lack of objective definition IS the problem!*

If we step back for a moment and observe this conflict from a distance, we might be tempted to ask a kind of meta-question. Why should the problem of consciousness have this peculiar indefiniteness to it? This new question is not the same as the problem of consciousness itself, because someone could conceivably write down a solution to the problem of consciousness tomorrow, and have it accepted by popular acclamation as *the* solution, and yet we could still turn around and ask: “Yes, but now please explain why the problem was so hard to even *articulate!*” That question—regarding the fact that this problem is different from all other problems because we cannot seem to define it in positive terms—might still be askable, even after the problem itself had been solved.

15.1.3 *An Outline of the Solution*

In fact, this meta-question needs to be addressed first, because it is the key to the mystery. I would like to propose that we can trace this slipperiness back to a specific cause: all intelligent systems must contain certain mechanisms in order to be fully intelligent, and a

side effect of these mechanisms is that some questions (to wit, the exact class of questions that correspond to consciousness) can neither be defined nor properly answered.

When we pose questions to ourselves we engage certain cognitive mechanisms whose job is to analyze the cognitive structures corresponding to concepts. If we take a careful look at what those mechanisms do, we notice that there are some situations in which they drive the philosopher's brain into a paradoxical mixed state in which she declares a certain aspect of the world to be both *real* and *intrinsically inexplicable*. In effect, there are certain concepts that, when analyzed, throw a monkey wrench into the analysis mechanism.

That is a precis of the first phase of the argument. But then there is a second—and in many ways more important—phase of the argument, in which we look at the “reality” of the particular concepts that break the cognitive mechanism responsible for explaining the world. Although phase one of the argument seemed to explain consciousness as a malfunction or short-circuit in the cognitive mechanism that builds explanations, in this second part we make an unusual turn into a new compromise, neither dualist nor physicalist, that resolves the problem of consciousness in a somewhat unorthodox way.

15.2 The Nature of Explanation

All facets of consciousness have one thing in common: they involve some particular types of introspection, because we “look inside” at our subjective experience of the world (qualia, sense of self, and so on) and ask what these experiences amount to. In order to analyze the nature of these introspections we need to take one step back and ask what happens when we think about any concept, not just those that involve subjective experience.

15.2.1 *The Analysis Mechanism*

In any intelligent system—either a biological mind or a sufficiently complete artificial general intelligence (AGI) system—there has to be a powerful mechanism that enables the system to analyze its own concepts. The system has to be able to explicitly think about what it knows, and to deconstruct that knowledge in many ways. If the degree of intelligence is high enough, the scope of this *analysis mechanism* (as it will henceforth be called) must be extremely broad. It must be able to ask questions about basic-level concepts, and then ask further questions about the constituent concepts that define basic-level concepts, and then continue asking questions all the way down to the deepest levels of its knowledge.

AGI systems will surely have this analysis mechanism at some point in the future, because it is a crucial part of the “general” in “artificial general intelligence,” but since there is currently no consensus about how to do this, we need to come up with a language that allows us to talk about the kind of things that such a mechanism might get up to. For the purposes of this chapter I am going to use a language derived from my own approach to AGI—what I have called elsewhere a “molecular framework” for cognition [6, 7].

It is important to emphasize that there are no critical features of the argument that hinge on the exact details of this molecular framework. In fact, the framework is so general that any other AGI formalism could, in principle, be translated into the MF style. However, the molecular framework is arguably more explicit about what the analysis mechanism does, so by using the language of the framework we get the benefit of a more concrete picture of its workings.

Some AGI formalisms will undoubtedly take a different approach, so to avoid confusion about the role played by the MF in this chapter, I will make the following claim, which has the status of a postulate about the future development of theories of intelligence:

- Postulate (*Analysis Mechanism Equivalence*): Any intelligent system with the ability to ask questions about the meaning of concepts, with the same scope and degree of detail as the average human mind, will necessarily have an equivalent to the analysis mechanism described here.

Different forms of the analysis mechanism will be proposed by different people, but the intended force of the above postulate is that in spite of all those differences, all (or most) of those analysis mechanisms will have the crucial features on which this explanation of consciousness depends. So the use of the molecular framework in this chapter does nothing to compromise the core of the argument.

15.2.2 *The Molecular Framework*

The Molecular Framework (MF) is a generic model of the core processes inside any system that engages in intelligent thought. It is designed to be both a description of human cognition and a way to characterize a broad range of AGI architectures.

The basic units of knowledge, in this framework, are what cognitive psychologists and AGI programmers loosely refer to as “concepts,” and these can stand for *things* [chair], *processes* [sitting], *relationships* [on], *actions* [describe], and so on.

The computational entities that encode concepts are found in two places in the system: the *background* (long-term memory, where there is effectively one entity per concept) and the *foreground*, which is roughly equivalent to working memory, or the contents of consciousness, since it contains the particular subset of concepts that the system is using in its current thoughts and all aspects of its current model of the world.

The concept-entities in the foreground are referred to here as *atoms*, while those in the background are called *elements*. This choice of terminology is designed to make it clear that, in the simplest form of the molecular framework, each concept is represented by just one element in the background, whereas there can be many instances of that concept in the foreground. If the system happens to be thinking about several instances of the [chair] concept there would be several [chair] *atoms* in the foreground, but there would only be one [chair] *element* in the background.

For the purposes of this chapter we will almost exclusively be concerned with atoms, and (therefore) with events happening in the foreground.

The contents of the foreground could be visualized as a space in which atoms link together to form clusters that represent models of the state of the world. One cluster might represent what the system is seeing right now, while another might represent sounds that are currently being heard, and yet another might represent some abstract thoughts that the system is entertaining (which may not have any connection to what is happening in its environment at that moment). The function of the foreground, then, is to hold models of the world.

Theorists differ in their preference for atoms that are either active or passive. A passive approach would have all the important mechanisms on the outside, so that the atoms were mere tokens manipulated by those mechanisms. An active approach, on the other hand, would have few, if any, external mechanisms that manipulate atoms, but instead would have all the interesting machinery in and between the atoms. In the present case we will adopt the active, self-organized point of view: the atoms themselves do (virtually) all the work of interacting with, and operating on, one another. This choice makes no difference to the argument, but it gives a clearer picture of some claims about semantics that come later.

Two other ingredients that need to be mentioned in this cognitive framework are external sensory input and the system's model of itself. Sensory information originates at the sensory receptors (retina, proprioceptive detectors, ears, etc.), is then pre-processed in some way, and finally arrives at the "edge" of the foreground, where it causes atoms

representing primitive sensory features to become active. Because of this inward flow of information (from the sensory organs to the edge of the foreground and then on into the “interior” region of the foreground), those atoms that are near the edge of the foreground will tend to represent more concrete, low-level concepts, while atoms nearer the center will be concerned with more high-level, abstract ideas.

The *self-model* is a structure (a large cluster of atoms), somewhere near the center of the foreground, that represents the system itself. It could be argued that this self-model is present in the foreground almost all of the time because when the mind is representing some aspect of the world, it usually keeps a representation of its own ongoing existence as part of that world. There are fluctuations in the size of the self model, and there may be occasions when it is almost absent, but most of the time we seem to maintain a model of at least the minimal aspects of our self, such as our being located in a particular place. Although the self-model proper is a representation of the system, somewhere near to it there would also be a part of the system that has the authority to initiate and control actions taken by the system: this could be described as the *Make It So* place.

Finally, note that there are a variety of operators at work in the foreground, whose role is to make changes to clusters of atoms. The atoms themselves do some of this work, by trying to activate other atoms with which they are consistent. So, for example, a [cat] atom that is linked to a [crouching-posture] atom will tend to activate an atom representing [pounce]. But there will also be operators that do such things as *concept creation* (making a new atom to encode a new conjunction of known atoms), *elaboration* (where some existing atoms are encouraged to bring in others that can represent more detailed aspects of what they are already representing), various forms of *analogy building*, and so on.

This cognitive framework depicts the process of thought as a collective effect of the interaction of all these atoms and operators. The foreground is a molecular soup in which atoms assemble themselves (with the help of operators) into semi-stable, dynamically changing structures. Hence the use of the term “molecular framework” to describe this approach to the modeling of cognition.

15.2.3 *Explanation in General*

Atoms can play two distinct roles in the foreground, mirroring the distinction between *use* and *mention* of words. When the word “cat” appears in a sentence like “The cat is on the chair,” it is being used to refer to a cat, but when the same word appears in a sentence like “The word *cat* has three letters,” the word itself, not the concept, is being mentioned.

In much the same way, if the foreground has atoms representing a chair in the outside world a [chair] atom will be part of the representation of that outside situation, and in this case the [chair] atom is simply being used to stand for something. But if the system asks itself “What is a chair?”, there will be one [chair] atom that stands as the target of the cluster of atoms representing the question. There is a strong difference, for the system, between representing a particular chair, and trying to ask questions about the *concept* of a chair. In this case the [chair] atom is being “mentioned” or referenced in the cluster of atoms that encode the question. It helps to picture the target atom as being placed in a special zone, or bubble, attached to the cluster of atoms that represent the question—whatever is inside the bubble is playing the special role of being examined, or mentioned. This is in contrast to the ordinary role that most atoms play when they are in the foreground, which is merely to be used as part of a representation.

So, when an atom, [x], becomes the target of a “What is x?” question, the [x] atom will be placed inside the bubble, then it will be elaborated and unpacked in various ways. What exactly does it mean to elaborate or unpack the atom? In effect, the atom is provoked into activating the other atoms that it would normally expect to see around it, if it were part of an ordinary representation in the foreground. Thus, the [chair] atom will cause atoms like [legs], [back], [seat], [sitting], [furniture] to be activated. And note that all of these activated atoms will be within the bubble that holds the target of the question.

What the question-cluster is doing is building a model of the meaning of [chair], inside the bubble. The various features and connotations of the [chair] concept try to link with one another to form a coherent cluster, and this coherent cluster inside the bubble is a model of the meaning, or definition, of the target concept.

One important aspect of this [meaning-of-“chair”] cluster is that the unpacking process tends to encourage more basic atoms to be activated. So the concepts that make up the final answer to the question will tend to be those that are subordinate features of the target atom. This is clearly just a matter of looking in the opposite direction from the one that is normally followed when an atom is being recognized: usually the activation of a cluster of atoms like [legs], [back] and [seat] will tend to cause the activation of the [chair] atom (this being the essence of the recognition process), so in order to get the meaning of [chair], what needs to happen is for the [chair] atom to follow the links backwards and divulge which other atoms would normally cause it to be activated.

We can call this set of elaboration and unpacking operations the *analysis mechanism*. Although it is convenient to refer to it as a single thing, the analysis mechanism is not really

a single entity, it is an open-ended toolkit of flexible, context-dependent operators. More like a loosely-defined segment of an ecology than a single creature. However, at the core of all these operators there will still be one basic component that grabs the target atom and starts following links to extract the other atoms that constitute the evidence (the features) that normally allow this atom to be activated. All other aspects of the analysis mechanism come into play after this automatic unpacking event.

If this were about narrow AI, rather than AGI, we might stop here and say that the essence of “explanation” was contained in the above account of how a [chair] concept is broken down into more detailed components. In an AGI system, however, the analysis mechanisms will have extensive connections to a large constellation of other structures and operators, including representations of, among other things:

- The person who asked the question that is being considered;
- That person’s intentions, when they asked the question;
- Knowledge about what kinds of explanation are appropriate in what contexts;
- The protocols for constructing sentences that deliver an answer;
- The status and reliability of the knowledge in question.

In other words, there is a world of difference between a dictionary lookup mechanism that regurgitates the definition of “chair” (something that might be adequate in a narrow AI system), and the massive burst of representational activity that is triggered when a human or an AGI is asked “What is a chair?”. The mental representation of that one question can be vastly different between cases where (say) the questioner is a young infant, a non-native-speaker learning the English language, and a professor who sets an exam question for a class of carpentry or philosophy students.

15.2.4 *Explaining Subjective Concepts*

In the case of human cognition, what happens when we try to answer a question about our subjective experience of the color red? In this case the analysis mechanism gets into trouble, because any questions about the essence of the color red will eventually reach down to a [redness] concept that is directly attached to an incoming signal line, and which therefore has no precursors. When the analysis mechanism tries to follow downward links to more basic atoms, it finds that this particular atom does not have any! The [redness] concept cannot be unpacked like most other concepts, because it lies at the very edge of the foreground: this is the place at which atoms are no longer used to represent parts of the

world. Outside the foreground there are various peripheral processing mechanisms, such as the primitive visual analysis machinery, but these are not within the scope of the operators that can play with atoms in the foreground itself. As far as the foreground is concerned the [redness] atom is activated by outside signals, not by other atoms internal to the foreground.

Notice that because of the rich set of processes mentioned above, the situation here is much worse than simply not knowing the meaning of a particular word. If we are asked to define a word we have never heard of, we can still talk about the letters or phonemes in the word, or specify where in the dictionary we would be able to find it, and so on. In the case of color qualia, though, the amount of analysis that can be done is precisely zero, so the analysis mechanism returns nothing.

Or does it return nothing? What exactly would we expect the analysis mechanism to do in this situation? Bear in mind that the mechanism itself is not intelligent (the global result of all these operations might be intelligent, but the individual mechanisms are just automatic), so it cannot know that the [red] concept is a special case that needs to be handled differently. So we would expect the mechanism to go right ahead and *go through the motions* of producing an answer. Something will come out of the end of the process, even if that something is an empty container where a cluster of atoms (representing the answer to the question) should have been.

So if the analysis mechanism does as much as it can, we would expect it to return an atom representing the concept [*subjective-essence-of-the-color-red*], but this atom is extremely unusual because it contains nothing that would allow it to be analyzed. And any further attempt to apply the analysis mechanism to *this* atom will yield just another atom of the same element. The system can only solve its problem by creating a unique type of atom whose only feature is itself.

This bottoming-out of the analysis mechanism causes the cognitive system to eventually report that “There is definitely something that it is like to be experiencing the subjective essence of red, but that ‘something’ is ineffable and inexplicable.” What it is saying is that there is a perfectly valid concept inside the foreground—the one that encodes the raw fact of redness—but that the analysis of this concept leads beyond the edge of the foreground (out into the sensory apparatus that supplies the foreground with visual signals), where the analysis mechanism is not able to go. This is the only way it can summarize the peculiar circumstance of analyzing [red] and getting [red] back as an answer.

This same short-circuit in the analysis mechanism is common to all of the consciousness questions. For qualia, the mechanism hits a dead end when it tries to probe the sensory

atoms at the edge of the foreground. In the case of emotions there are patterns of activation coming from deeper centers in the brain, which are also (arguably) beyond the reach of the foreground. For the concept of self, there is a core representation of the self that cannot be analyzed further because its purpose is to represent, literally, itself. The analysis mechanism can only operate within the foreground, and it seems that all aspects of subjective phenomenology are associated with atoms that lie right on the boundary.

In every case where this happens it is not really a “failure” of the mechanism, in the sense that something is broken, it is just an unavoidable consequence of the fact that the cognitive system is powerful enough to recursively answer questions about its own knowledge. If this were really a failure due to a badly designed mechanism, then it might be possible to build a different type of intelligent system that did not have this problem. Perhaps it would be possible to design around this problem, but it seems just as likely that any attempt to build a system capable of analyzing its own knowledge without limitations will have a boundary that causes the same short-circuit. Attempts to get the system to cope gracefully with this problem may only move the boundary to some other place, because any fix that is powerful enough to make the system not sense a problem, for these special concepts, is likely to have the unwanted side effect of causing the system to be limited in the depth of its analytic thought.

If a system has the ability to powerfully analyze its own concepts, then, it will have to notice the fact that some concepts are different because they cannot be analyzed further. If we try to imagine a cognitive system that is, somehow, not capable of representing the difference between these two classes of concepts, we surely get into all kinds of trouble. The system can be asked the direct question “When you look at the color red, what is the difference between that and the color blue? Because my friend here, who has never been able to see the color blue, would like to know.” In the face of that direct question, the system is not only supposed to find no difference between its internal ability to handle the analysis of the [redness] concept and its handling of others, like the [chair] concept, it is also supposed to somehow not notice that its verbal reply contains the peculiarly empty phrase “Uh, I cannot think of any way to describe the difference.” At some level, it must surely be possible for us to draw the attention of this hypothetical cognitive system to the fact that it is drawing a blank for some kinds of concept and not for others—and as soon as we can draw its attention to that fact, it is on a slippery slope toward the admission that there is a drastic difference between subjective phenomenology and objective concepts. There is something approaching a logical incoherence in the idea that a cognitive system can have

a powerful (i.e. human-level) analysis mechanism but also be immune to the failure mode described above.

15.2.5 *The “That Misses The Point” Objection*

The principal philosophical objection to the above argument is that it misses the point. It explains only the *locutions* that philosophers produce when talking about consciousness, not the actual experiences they have. The proposed explanation looks like it has slipped from being about the phenomenology, at the beginning, to being about the psychology (the cognitive mechanisms that cause people to say the things they do about consciousness) at the end. That would make this entire proposal into a discussion of a non-hard problem, because the philosopher can listen to the above account and yet still say “Yes, but why would *that* short circuit in my psychological mechanism cause *this* particular feeling in my phenomenology?”

Here we come to the crux of the proposed explanation of consciousness. Everything said so far could, indeed, be taken as just another example of a non-hard sidetracking of the core question. What makes this a real attempt to address the hard problem of consciousness is the fact that there is a flaw in the above objection, because *it involves an implicit usage of the very mechanism that is supposed to be causing the trouble.*

So if someone says “There is something missing from this argument, because when I look at my subjective experience I see things (my qualia!) that are not referenced in any way by the argument”, what they are doing is asking for an explanation of (say) color qualia that is just as satisfactory as explanations of ordinary concepts, and they are noticing that the proposed explanation is inferior because it leaves something out. But this within-the-system comparison of consciousness with ordinary concepts is precisely the kind of thought process that will invoke the analysis mechanism! The analysis mechanism inside the mind of the philosopher who raises this objection will then come back with the verdict that the proposed explanation fails to describe the nature of conscious experience, just as other attempts to explain consciousness have failed.

The proposed explanation, then, can only be internally consistent with itself if the philosopher finds the explanation wanting.

There is something wickedly recursive about this situation. The proposed explanation does not address the question of why the phenomenology of the color red should be the way that it is—so in a certain respect the explanation could be said to have failed. But at the core of the explanation itself is the prediction that when the explanation is processed through

the head of a philosopher who tries to find objections to it, the explanation must necessarily cause the philosopher's own analysis mechanism to become short-circuited, resulting in a verdict that the explanation delivers no account of the phenomenology.

Do all of the philosophical objections to this argument fall into the same category (i.e. they depend for their force on a deployment of the analysis mechanism that is mentioned in the argument)? I claim that they do, for the following reason. The way that Chalmers [2] formulated it, there is a certain simplicity to the hard problem, because whenever an objection is lodged against any proposed resolution of the problem, the objection always works its way back to the same final point: the proposed explanation fails to make contact with the phenomenological mystery. In other words, the buck always stops with "Yes, but there is still something missing from this explanation." Now, the way that I interpret all of these different proposed explanations for consciousness—and the matching objections raised by philosophers who say that the explanation fails to account for the hard problem—is that these various proposals may differ in the way that they approach that final step, but that in the end it is only the final step that matters. In other words, I am not aware of any objection to the explanation proposed in this chapter that does not rely for its force on that final step, when the philosophical objection deploys the analysis mechanism, and thereby concludes that the proposal does not work *because* the analysis mechanism in the head of the philosopher returned a null result. And if (as I claim) all such objections eventually come back to that same place, they can all be dealt with in the same way.

But this still leaves something of an impasse. The argument does indeed say nothing about the nature of conscious experience, *qua* subjective experience, but it does say why it cannot supply an explanation. Is explaining why we cannot explain something the same as explaining it? This is the question to be considered next.

15.3 The Real Meaning of Meaning

This may seem a rather unsatisfactory solution to the problem of consciousness, because it appears to say that our most immediate, subjective experience of the world is an artifact of the operation of the brain. The proposed explanation of consciousness is that subjective phenomenology is a thing that intelligent systems *must* say they experience (because their analysis mechanism would not function correctly otherwise)—but this seems to put consciousness in the same category as visual artifacts, illusions, hallucinations and

the like. But something is surely wrong with this conclusion: it would be bizarre to treat something that dominates every aspect of our waking lives as if it were an artifact.

I believe that condemning consciousness as an artifact is the wrong conclusion to draw from the above explanation. I am now going to make a case that all of the various subjective phenomena associated with consciousness should be considered just as “real” as any other phenomena in the universe, but that science and philosophy must concede that consciousness has the special status of being unanalyzable. The appropriate conclusion is that consciousness can be predicted to occur under certain circumstances (namely, when an intelligent system has the kind of powerful analysis mechanism described earlier), but that there are strict limits to what we can say about its nature. We are obliged to say that these things are real, but even though they are real they are beyond the reach of science.

15.3.1 *Getting to the Bottom of Semantics*

The crucial question that we need to decide is what status we should give to the atoms in a cognitive system that have the peculiar property of making the analysis mechanism return a verdict of “this is real, but nothing can be said about it”.

To answer this question in a convincing way, we need to understand the criteria we use when we decide:

- The “realness” or validity of different concepts (their epistemology);
- The meaning of concepts, or the relationships between concepts and things in the world (their semantics and ontology);
- The validity of concepts that are used in scientific explanations.

We cannot simply wave our hands and pick a set of criteria to apply to these things, we need to have some convincing reasons to make one choice or another.

Who adjudicates the question of which concepts are “real” and which are “artifacts”? On what *basis* can we conclude that some concepts (e.g. the phenomenological essence of redness) can be dismissed as “not real” or “artifactual”?

There seem to be two options here. One would involve taking an already well-developed theory of semantics or ontology—an off-the-shelf theory, so to speak—and then applying it to the present case. The second would be to take a detailed look at the various semantic/ontological frameworks that are available and find out which one is grounded most firmly; which one is secure enough in its foundations to be *the* true theory of meaning.

Unfortunately, both of these options lead us into a trap. The trap works roughly as follows. Suppose that we put forward a Theory of Meaning (let's call it Theory X), in the hope that Theory X will be so ontologically complete that it gives us the "correct" or "valid" method for deciding which concepts are real and which are artifacts; which concepts are scientifically valid and which are illusory/insufficient/incoherent.

Having made that choice, we can be sure of one thing: given how difficult it is to construct a Theory of Meaning, there will be some fairly abstract concepts involved in this theory. And as a result the theory itself will come under scrutiny for its conceptual coherence. Lying at the root of this theory there will be some assumptions that support the rest of the theory. Are those assumptions justified? Are they valid, sufficient or coherent? Are they necessary truths? You can see where this is leading: any Theory of Meaning that purports to be the way to decide whether or not concepts have true meaning (refer to actual things in the world) is bound to be a potential subject of its own mechanism. But in that case the theory would end up justifying its own validity by referring to criteria that it already assumes to be correct.

The conclusion to draw from these considerations is that any Theory X that claims to supply *absolute* standards for evaluating the realness or validity of concepts cannot be consistent. There is no such thing as an objective theory of meaning.

This circularity or question-begging problem applies equally to issues like the meaning of "meaning" and explanations of the concept of "explanation," and it afflicts anyone who proposes that the universe can be discovered to contain some absolute, objective standards for the "meanings" of things, or for the fundamental nature of explanatory force.

15.3.2 *Extreme Cognitive Semantics*

There is only one attitude to ontology and semantics that seems capable of escaping from this trap, and that is an approach that could be labeled "Extreme Cognitive Semantics"—the idea that there is no absolute, objective standard for the mapping between symbols inside a cognitive system and things in the world, because this mapping is entirely determined by the purely contingent fact of the design of real cognitive systems [3, 8]. There is no such thing as the pure, objective meaning of the symbols that cognitive systems use, there is only the way that cognitive systems actually do, as a matter of fact, use them. Meanings are determined by the ugly, inelegant design of cognitive systems, and that is the end of it.

How does this impact our attempt to decide the status of those atoms that cause our analysis mechanisms to bottom out? The first conclusion should be that, since the meanings and status of all atoms are governed by the way that cognitive systems actually use them, we should give far less weight to an externally-imposed formalism—like the possible-worlds semantics popular in artificial intelligence [4]—which says that subjective concepts point to nothing in the real world (or in functions defined over possible worlds) and are therefore fictitious.

Second—and in much the same vein—we can note that the atoms in question are such an unusual and extreme case, that formalisms like traditional semantics should not even be expected to handle them. This puts the shoe on the other foot: it is not that these semantic formalisms are capable of dismissing the consciousness-concepts and *therefore* the latter are invalid, it is rather that the formalisms are too weak to be used for such extreme cases, and therefore they have no jurisdiction in the matter.

Finally, we can use the Extreme Cognitive Semantics viewpoint to ask if there is a way to make sense of the idea that various concepts possess different degrees of “realness.”

In order to do this, we need to look at how concepts are judged to be or not be “real” in ordinary usage. Ordinary usage of this concept seems to have two main aspects. The first involves the precise content of a concept and how it connects to other concepts. So, unicorns are not real because they connect to our other concepts in ways that clearly involve them residing only in stories. The second criterion that we use to judge the realness of a concept is the directness and immediacy of its phenomenology. Tangible, smellable, seeable things that lie close at hand are always more real. Abstract concepts are less real.

Interestingly, the consciousness atoms that we have been considering in this argument ([redness], [self] and so on) score very differently on these two measures of realness. They connect poorly to other concepts on their downward side because we cannot unpack them. But on the other hand they are the most immediate, closest, most tangible concepts of all, because they *define* what it means to be “immediate” and “tangible.” When we say that a concept is more real the more concrete and tangible it is, what we actually mean is that it gets more real the closer it gets to the most basic of all concepts. In a sense there is a hierarchy of realness among our concepts, with those concepts that are phenomenologically rich being the most immediate and real, and with a decrease in that richness and immediacy as we go toward more abstract concepts.

15.3.3 Implications

What can we conclude from this analysis? I believe that the second of these two criteria of “realness” is the one that should dominate. We normally consider the concepts that are closest to our phenomenology to be the ones that are the best-connected and most thoroughly consistent with the rest of our conceptual systems. But the concepts associated with consciousness are an exception to that rule: they have the most immediacy, but a complete lack of connections going to other concepts that “explain” what they are. If we are forced to choose which of the two criteria is more important, it seems most coherent to treat immediacy as the real arbiter of what counts as real. Perhaps the best way to summarize the reason why this should be so is to consider the fact that in ordinary usage “realness” of a concept is to some extent inherited: if a concept is defined in terms of others that are considered very real, then it will be all the more real. But then it would make little sense to say that all concepts obey the rule that they are more real, valid and tangible, the closer they are to the phenomenological concepts at the root of the tree ... but that the last layer of concepts down at the root are themselves *not* real.

Given these considerations, I maintain that the correct explanation for consciousness is that all of its various phenomenological facets deserve to be called as “real” as any other concept we have, because there are no meaningful *objective* standards that we can apply to judge them otherwise. But while they deserve to be called “real” they also have the unique status of being beyond the reach of scientific inquiry. We can talk about the circumstances under which they arise, but we can never analyze their intrinsic nature. Science should admit that these phenomena are, in a profound and specialized sense, mysteries that lie beyond our reach.

This seems to me a unique and unusual compromise between materialist and dualist conceptions of mind. Minds are a consequence of a certain kind of computation; but they also contain some mysteries that can never be explained in a conventional way. We cannot give scientific explanations for subjective phenomena, but we can say exactly *why* we cannot do so. In the end, we can both explain consciousness and not explain it.

15.4 Some Falsifiable Predictions

This theory of consciousness can be used to make some falsifiable predictions. We are not yet in a position to make empirical tests of these predictions, because the tests would seem to require the kind of nanotechnology that would let us rewire our brains on the fly,

but the tests can be lodged in the record, against the day that some experimentalist can take up the challenge of implementing them.

The uniqueness of these predictions lies in the fact that there is a boundary (the edge of the foreground) at which the analysis mechanism gets into trouble. In each case, the prediction is that these phenomena will occur *at exactly that boundary*, and nowhere else. Bear in mind, however, that we do not yet know where this boundary actually lies, in the implementation that is the human brain.

If we are able to construct AGI systems that function at the human level of intelligence, with a full complement of cognitive mechanisms that includes the analysis mechanism described earlier, then these predictions will be testable by asking the AGI what it experiences in each of the following cases.

15.4.1 Prediction 1: Blindsight

Some kinds of brain damage cause people to experience ‘blindsight’, a condition in which the person reports little or no conscious awareness of a certain visual stimulus, while at the same time they can sometimes act on the stimulus as if it were visible [9].

The prediction in this case is that some of the visual pathways in the human brain will be found to lie within the scope of the analysis mechanism, while others will be found to lie outside. The ones outside the scope of the analysis mechanism will be precisely those that, when spared after damage, allow visual awareness without consciousness.

15.4.2 Prediction 2: New Qualia

If we built three sets of new color receptors in the eyes, with sensitivity to three bands in the ultraviolet range, and if we built enough brain wiring to supply the foreground with new concept-atoms triggered by these receptors, this should give rise to three new color qualia. After acclimatizing to the new qualia, we could then swap connections on the old color receptors and the new UV pathways, at a point that lies *just outside the scope of the analysis mechanism*. The prediction here is that the two sets of color qualia will be swapped in such a way that the new qualia will be associated with the old visible-light colors, and that this will only occur if the swap happens beyond the analysis mechanism.

If we then removed all trace of the new UV pathways and retinal receptors, outside the foreground (beyond the reach of the analysis mechanism), then the old color qualia would disappear, leaving only the new qualia. The subject will have a ghost of a memory of the old color qualia, because the old concept atoms will still be there, but those atoms will only

be seen in imagination. And if we later reintroduce a set of three color receptors and do the whole procedure again, we can bring back the old color qualia if we are careful to ensure that the new visual receptors trigger the foreground concept-atoms previously used for the visible-light colors. The subject would suddenly see the old qualia again

15.4.3 Prediction 3: *Synaesthetic Qualia*

Take the system described above (after the first installation of new qualia) and arrange for a cello timbre to excite the old concept-atoms that would have caused red qualia. Cello sounds will now cause the system to have a disembodied feeling of redness.

15.4.4 Prediction 4: *Mind Melds*

Join two minds so that B has access to the visual sensorium of A, using new concept-atoms in B's head to encode the incoming information from A. B would say that she knew what A's qualia were like, because she would be experiencing new qualia. If B were getting sounds from A's brain, but these were triggering entirely new atoms designed especially to encode the signals, B would say that A did not experience sound the way she did, but in an entirely new way. If, on the other hand, the incoming signals from A triggered the same sound atoms that B uses (with no new atom types being created), then B will report that she is hearing all of A's sonic input mixed in with her own. In much the same way, B could be given an extra region of her foreground periphery exclusively devoted to the visual stream coming from A. She would then say that she had two heads, but that she could only attend to one of them at a time. With new atoms for the colors, again, she would report that B's qualia differed from her own.

Note that any absolute comparison between the way that different people experience the world is not possible. The reported qualia in these mind-meld cases would be entirely dependent on choices of how to cross-wire the systems.

15.5 Conclusion

The simplest explanation for consciousness is that the various phenomena involved have an irreducible dual aspect to them. On the one hand, they are explicable because we can understand that they are the result of a powerful cognitive system using its analysis mechanism to probe concepts that happen to be beyond its reach. But on the other hand, these concepts deserve to be treated as the most immediate and real objects in the universe,

because they define the very foundation of what it means for something to be real. These consciousness concepts—such as the subjective phenomenological experience of the color red—cannot be explained by any further scientific analysis. Rather than try to resolve this situation by allowing one interpretation to trump the other, it seems more parsimonious to conclude that both are true at the same time, and that the subjective aspects of experience belong to a new category of their own: they are real but inexplicable, and no further scientific analysis of them will be able to penetrate their essential nature.

According to this analysis, an Artificial General Intelligence designed in such a way that it had the same problems with its analysis mechanism that we humans do (and I have argued that this would mean any fully sentient computer capable of a near-human degree of intelligence, because the analysis mechanism plays such a critical role in all types of general intelligence) would experience consciousness for the same reasons that we do. We could never prove this statement the way that we prove statements about objective concepts, but that is part of what it means to say that consciousness concepts have a special status (they are real, but beyond analysis). The only way to be consistent about our interpretation of these phenomena is to say that, insofar as we can say anything at all about consciousness, we can be sure that the right kind of artificial general intelligence would experience a subjective phenomenology comparable in scope to human subjective consciousness.

Bibliography

- [1] Cambell, K. K. (1970). *Body and Mind* (Doubleday, New York).
- [2] Chalmers, D. J. (1996). *The Conscious Mind: In Search of a Fundamental Theory* (Oxford University Press, Oxford).
- [3] Croft, W. and Cruse, D. A. (2004). *Cognitive Linguistics* (Cambridge University Press, Cambridge).
- [4] Dowty, D. R., Wall, R. E., and Peters, S. (1981). *Introduction to Montague Semantics* (D. Reidel, Dordrecht).
- [5] Kirk, D. (1974). Zombies versus materialists, *Aristotelian Society* **48** (suppl.), pp. 135–52.
- [6] Loosemore, R. P. W. (2007). Complex Systems, Artificial Intelligence and Theoretical Psychology, in B. Goertzel and P. Wang (eds.), *Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms* (IOS Press, Amsterdam), pp. 159–173.
- [7] Loosemore, R. P. W. and Harley, T. A. (2010). Brains and Minds: On the Usefulness of Localization Data to Cognitive Psychology, in M. Bunzl and S. J. Hanson (eds.), *Foundational Issues in Human Brain Mapping* (MIT Press, Cambridge, MA), pp. 217–240.
- [8] Smith, L. B. and Smith, L. K. (1997). Perceiving and Remembering: Category Stability, Variability, and Development, in K. Lamberts and D. Shanks (eds.), *Knowledge, Concepts, and Categories* (Cambridge University Press, Cambridge).
- [9] Weiskrantz, L. (1986). *Blindsight: A Case Study and Implications* (Oxford University Press, Oxford).

Chapter 16

Theories of Artificial Intelligence — Meta-theoretical considerations

Pei Wang

Temple University, Philadelphia, USA

<http://www.cis.temple.edu/~pwang/>

This chapter addresses several central meta-theoretical issues of AI and AGI. After analyzing the nature of the field, three criteria for desired theories are proposed: *correctness*, *concreteness*, and *compactness*. The criteria are clarified in the AI context, and using them, the current situation in the field is evaluated.

16.1 The problem of AI theory

Though it is a common practice for a field of science or engineering to be guided and identified by the corresponding theories, the field of Artificial Intelligence (AI) seems to be an exception. After more than half of a century since its formation, AI still has no widely accepted theory, and in the related discussions the following opinions are often heard:

- “*The best model of intelligence is the human brain itself (and all theories are merely poor approximations ...)*”
- “*There is no need for any new theory, since AI can be built according to X (depending on who said it, the X can be mathematical logic, probability theory, theory of computation, ...)*”
- “*A theory of AI has to be established piece by piece, and we are starting from Y (depending on who said it, the Y can be search, reasoning, learning, perception, actions, ...)*”
- “*There cannot be any good theory of intelligence (since intelligence is so complicated, though our work is obviously central to it ...)*”

- *“Theoretical debating is a waste of time (and we should focus on practical applications. For example, an intelligent system should be able to ...)”*
- *“A good theory only comes at the end of the research (so don’t worry about it now, and it will come as long as we continue the current research on ...)”*

There is a historical reason for this situation. Though the idea of “thinking machine” can be traced further back in history, the field of AI was started from the realization that computers, though initially designed to do numerical calculations, can be made to carry out other mental activities, such as theorem proving and game playing, which are hard intellectual problems that are usually considered as demanding “intelligence” [12, 28]. This “problem-oriented” attitude toward AI focuses on the problem-solving capability of a computer system, while does not care much for the underlying theory. Consequently, the early works in AI often showed the “Look, ma, no hands” syndrome — “A paper reports that a computer has been programmed to do what no computer program has previously done, and that constitutes the report. How science has been advanced by this work or other people are aided in their work may be unapparent.” [25]. For such a work, “the question, Where’s the AI? is a tough one to answer” [40].

To many AI researchers, the lack of a common theory is not an issue at all. As said by Minsky [30], “Our minds contains processes that enable us to solve problems we consider difficult. ‘Intelligence’ is our name for whichever of those processes we don’t yet understand.” According to this opinion, a “theory of AI” is impossible *by definition*, since we cannot have a theory for “those processes we don’t yet understand” — as soon as we have a good theory for such a process, it is no longer considered as AI anymore [30, 40].

To get out of this annoying situation, in mainstream AI “intelligence” is treated as the collaboration of a group of loosely coupled functions, each of them can be separately specified in computational and algorithmic terms, implemented in computers, and used to solve certain practical problems [24, 39]. In an influential AI textbook by Russell and Norvig [39], it is written that in the late 1980s “AI adopts the scientific method”, since “It is now more common to build on existing theories than to propose brand new ones ...”. However, it is not mentioned that none of those “existing theories” were proposed with intelligence as the subject matter, nor has shown the potential of solving the problem of intelligence as a whole.

Though in this way the field has produced valuable results in the past decades, it still suffers from internal fragmentation [5] and “paradigmatic mess” [11], largely due to the

lack of a common theoretical foundation. There have been many debates on the nature or objective of the field, or on what type of theory it should or can have [8, 21, 42, 50].

Though the pursuit of unified theories of AI is widely considered as futile in the field, there is still a small number of AI researchers who believe that such a theory is possible, and worthwhile to be investigated. The best known work in this direction is the “Unified Theories of Cognition” by Newell [31], in which he argued for the necessity for AI and cognitive science to have unified theories, and proposed his theory, which attempts to cover both AI and human intelligence. Similar attempts include the works of Albus [1] and Pollock [36].

In recent years, the term “Artificial General Intelligence” (AGI) is adopted by a group of AI researchers to emphasize the general-purpose and holistic nature of the “intelligence” they are after [17, 49]. Since AGI treats intelligence as a whole, there are more efforts to establish unified theories [3, 4, 6, 9, 13, 15, 20, 41, 46], though none of them is mature or convincing enough to obtain wide acceptance in the field at the current moment [7].

Even though the AGI community is more “AI-theory-oriented” than mainstream AI, not every AGI project is based on some theory about intelligence. As in mainstream AI, a project is often guided by one, or more than one, of the following considerations:

Practical problem-solving demands: Since intelligence is displayed in the problem-solving capability of a system, many projects target problems that currently can be solved by humans only. Such a system is usually designed and analyzed according to the theory of computation [19, 24].

Knowledge about human intelligence: Since the human mind has the best-known form of intelligence, many projects aim at duplicating certain aspects of the human mind or brain. Such a system is usually designed and analyzed according to the theories in psychology or neuroscience [31, 38].

Available normative models: Since intelligence intuitively means “to do the right thing”, many projects are designed and analyzed as models of rationality or optimization, according to mathematical theories like classical logic and probability theory, though usually with extensions and/or revisions [26, 35].

Even the AGI projects that are based on certain theories on AI are moving in very different directions, mainly because of the difference in their theoretical foundations, as well as the influence of the above considerations. This collection provides a representative example of the diversity in the theoretical study of AGI.

Consequently, currently in the field of AI/AGI there are very different opinions on research goal [23, 47], roadmap [16, 27], evaluation criteria [2, 22], etc. Though each researcher can and should make decisions on the above issues for his/her own project, for the field as a whole this paradigmatic mess makes comparison and cooperation difficult, if not impossible.

In this chapter, I will not promote my own theory of AI (which is described in my previous publications [45, 46, 48]), nor to evaluate the other theories one by one, but to address the major *meta-level* issues about AI theories, such as

- What is the nature of an AI theory?
- How to evaluate an AI theory?
- Why do we lack a good theory?

This chapter attempts to clarify the related issues, so as to pave the way to a solid theoretical foundation for AGI, which is also the original and ultimate form of AI. For this reason, in the following “AI” is mainly used to mean “AGI”, rather than the current mainstream practice.

16.2 Nature and content of AI theories

In a field of science or engineering, a “theory” usually refers to a system of concepts and statements on the subject matter of the field. Generally speaking, there are two types of theory:

Descriptive theory: Such a theory starts with certain observations in the field. The theory provides a generalization and explanation of the observations, as well as predictions for future events, so as to guide people’s behaviors. The theories in natural science are the best examples of this type.

Normative theory: Such a theory starts with certain assumptions, then derives conclusions from them. When the assumptions are accepted as applicable in a field, all the conclusions should also be accepted as true. Mathematics and engineering theories are the best examples of this type.¹

¹In fields like economics and law, a “normative” theory or model specifies what people *should* do, often for *ethical* reasons. It is not what the word means here. Instead, in this chapter a “normative” theory specifies what people should do for *rational* reasons. This usage is common in the study of human reasoning and decision making, for example see [14].

Though it is possible for these two types of theory to interweave (in the sense that parts of a theory may belong to the other type), for a theory as a whole its type is still usually clear. For example, modern physics uses a lot of mathematics in it, but it does not change the overall descriptive nature of the theories in physics. On the contrary, computer science is mainly based on normative theories on how to build and use computer systems, even though empirical methods are widely used to test the systems.²

What makes a “Theory of AI” special on this aspect is that it needs to be *both* descriptive and normative, in a certain sense.

AI studies the similarity and the difference between “The Computer and the Brain”, as suggested by the title of [44]. This research is directly driven by the observation that though the computer systems can take over human’s mental labor in many situations (and often do a better job), there are nevertheless still many features of the human mental activities that have not been reproduced by computers. An AI theory should provide a bridge over this gap between “the Brain” and “the Computer”, so as to guide the designing and building of computer systems that are similar to the human mind in its “mental power”. “Intelligence” is simply the word whose intuitive meaning is the closest to the capability or property to be duplicated from the brain to the computer, though some people may prefer to use other words like “cognition”, “mind”, or “thinking”. The choice of word here does not change the nature of this problem too much.

Given this objective, an AI theory must identify the (known or potential) similarities between two entities, “the Brain” and “the Computer”, which are very different on many aspects. Furthermore, human intelligence is an existing phenomenon, while computer intelligence is something to be built, for which an accurate description does not exist at this moment. Consequently, an AI theory should be *descriptive* with respect to human intelligence (not in all details, but in basic principles, functions and mechanisms), and at the same time, be *normative* to computer intelligence. That is, on one hand, the theory should summarize and explain how the human mind works, at a proper level and scope of description; on the other hand, it should guide the design and development of computer systems, so as to make them “just like the human mind”, at the same level and scope of description.

A theory for this field is surely centered at the concept of “intelligence”. Accurately speaking, there are three concepts involved here:

Human Intelligence (HI), the intelligence as displayed by human beings;

Computer Intelligence (CI), the intelligence as to be displayed by computer systems;

²On this topic, I disagree with Newell and Simon’s opinion on “Computer science as empirical inquiry” [32].

General Intelligence (GI), the general and common description of both HI and CI.

For the current discussion, HI can also be referred to as “natural intelligence”, CI as “artificial intelligence”, and GI simply as “intelligence”, which also covers other concepts like “animal intelligence”, “collective intelligence”, “alien intelligence”, etc., as special cases [48].

Roughly speaking, the content of the theory must cover certain mechanisms in the human mind (as the HI), then generalize and abstract them (to be the GI), and finally specify them in a computational form (to become the CI). No matter what names are used, the distinction and relationship among the three concepts are necessary for an AI theory, because the theory needs to identify the common properties between human beings and computer systems, while still to acknowledge their differences in other aspects.³

Now it is easy to see that in an AI theory, the part about HI is mostly descriptive, that about CI is mostly normative, and that about GI is both.

The human mind is a phenomenon that has been studied by many branches of science from different perspectives and with different focuses. There have been many theories about it in psychology, neuroscience, biology, philosophy, linguistics, anthropology, sociology, etc. When talking about HI, what AI researchers usually do is to selectively acquire concepts and conclusions from the other fields, and to reorganize them in a systematic way. As a result, we get a theory that summarizes certain observed phenomenon of the human mind. Such a theory is fundamentally *synthetic* and *empirical*, in that its conclusions are summaries of common knowledge on how the human mind works, so it is verified by comparing its conclusions to actual human (mental) activities. Here the procedure is basically the same as in natural science. The only special thing is the *selectivity* coming from the (different) understandings of the concept “intelligence”: different researchers may include different phenomena within the scope of HI, which has no “natural” boundary.

On the contrary, a theory about CI has to be normative, since this phenomenon does not exist naturally, and the main function of the theory is to tell the practitioners how to produce it. As a normative theory, its basic assumptions come from two major sources: knowledge of intelligence that describes what *should* be done, and knowledge of computer that describes what *can* be done. Combined together, this knowledge can guide the whole

³Some people may argue that AI researchers are only responsible for the CI part of the picture, because the HI part should be provided by psychologists, and the GI part should be covered by a “theory of general intelligence”, contributed by philosophers, logicians, mathematicians, and other researchers working on general and abstract systems. Though there is some truth in this argument, at the current time there is no established theory of GI that we AI researchers can accept as guidance, so we have to work on the whole picture, even though part of it is beyond our career training.

design and development process, by specifying the design objective, selecting some theoretical and technical tools, drawing a blueprint of the system's architecture, planning a development roadmap, evaluating the progress, and verifying the results. Here the procedure is basically the same as in engineering. The only special thing is the *selectivity* coming from the (different) understandings of the concept "intelligence": different researchers may define the concept differently, which will change everything in the following development.

As the common generalization of HI and CI, a theory of GI is both descriptive and normative. On one hand, the theory should explain how human intelligence works as a special case, and on the other hand, it should describe how intelligence works in general, so as to guide how an intelligent computer system should be designed. Therefore, this theory should be presented in a "medium-neutral" language that does not assume the special details of either the human brain or the computer hardware. At the same time, since it is less restricted by the "low-level" constraints, this part of the theory gives the researchers the largest freedom, compared to the HI and the CI part. Consequently, this is also where the existing theories differ most from each other — the differences among the theories are not much on *how* the brain, mind, or computer works, but on *where* the brain and the machine should be similar to each other [47].

In the textbook by Russell and Norvig [39], different approaches toward AI are categorized according to whether they are designed to be *thinking* or *acting* "humanly" or "rationally". It seems that the former is mainly guided by descriptive theories, while the latter by normative theories. Though such a difference indeed exists, it is more subtle than what these two words suggest. Since the basic assumptions and principles of all models of rationality come from abstraction and idealization of the human thinking process, "rationally" thinking/acting is actually a special type of "humanly" thinking/acting. For example, though the "Universal AI" model AIXI by Hutter [20] is presented in a highly abstract and mathematical form, its understanding of "intelligence" is still inspired and justified according to certain opinions about the notion in psychology [23]. On the other extreme, though Hawkins' HTM model of intelligence is based on certain neuroscientific findings, it is not an attempt to model the human brain in all aspects and in all details, but to *selectively* emulate certain mechanisms that are believed to be "the crux of intelligence" [18]. Therefore, the difference between AIXI and HTM, as well as among the other AGI models, is not on whether to learn from the human brain/mind (the answer is always "yes", since it is the best-known form of intelligence), or whether to idealize and simplify the knowledge obtained from the human brain/mind (the answer is also always "yes", since a computer

cannot become identical to the brain in all aspects), but on *where* to focus and *how much* to abstract and generalize.

From the same knowledge about the human mind, there are many meaningful ways to establish a notion of HI, by focusing on different aspects of the phenomena; from the same notion of HI, there are many meaningful ways to establish a notion of GI, by describing intelligence on different levels, with different granularities and scopes; from the same notion of GI, there are many meaningful ways to establish a notion of CI, by assuming different hardware/software platforms and working environments. The systems developed according to different notions will surely have different properties and practical applications, and are “similar to the human mind” in different senses. Unless one commits to a particular definition of intelligence, there is no absolute standard to decide which of these ways is “the correct way” to establish a theory of AI.

The current collection to which this chapter belongs provides a concrete example for this situation: though the chapter authors all use the notion of “intelligence”, and are explaining related phenomena, the theories they proposed are very different. It is not necessarily the case that at most one of the theory is “correct” or really captures intelligence “as it is”, while all the others are “wrong”, since each of them represents a certain perspective; nor can the issue be resolved by pooling the perspectives altogether, because they are often incommensurable, due to the usage of different concepts. This diversity is a major source of difficulty in theoretical discussions of AI.

16.3 Desired properties of a theory

Though there are reasons for different AI theories to be proposed, it does not mean that all of them are equally good. The following three desired properties of a scientific theory are proposed and discussed in my own theory of intelligence [48] (Section 6.2):

- *Correctness*: A theory should be supported by available evidence.
- *Concreteness*: A theory should be instructive in problem solving.
- *Compactness*: A theory should be simple.

Though these properties are proposed for scientific theories in general, in this chapter they will be discussed in the context of AI. Especially, let us see what they mean for an AI theory that is both descriptive (for human minds) and normative (for computer systems).

Correctness

Since the best-known form of *intelligence* is human intelligence, an AI theory is *correct* if it is supported by the available knowledge about the human mind. In this aspect, AI is not that different from any natural science, in that the correctness of a theory is verified empirically, rather than proved according to some priori postulates. Since the study of the human mind has been going on in many disciplines for a long time, AI researchers often do not need to carry out their own research on human subjects, but to inherit the conclusions from the related disciplines, including (though not limited to) psychology, linguistics, philosophy, neuroscience, and anthropology.

This task is not as simple as it sounds, since an AI theory cannot simply copy the concepts and statements from the related disciplines — to form the HI part of the theory, *selection* is needed; to form the GI part of the theory, *generalization* is needed.

“Intelligence” is not related to every aspect of a human being, and AI is not an attempt to clone a human. Even though the concept of intelligence has many different understandings, it is mainly about the *mental* properties of human beings, rather than their *physical* or *biological* properties (though those properties have impacts in the *content* of human thought). Furthermore, even only for lexical considerations, the notion of “Intelligence” should be more *general* than the notion of “Human Intelligence”, so as to cover the non-human forms of intelligence. Therefore, an AI theory needs to decide the boundary of its empirical evidence, by indicating which processes and mechanisms in the human mind/brain/body is directly relevant to AI, and which of them are not. In other words, an AI theory must specify the scope and extent to which a computer is (or will be) similar to a human.

The following two extreme positions on this issue are obviously improper — if HI is specified in such a “tight” way that is bounded to all aspects of a human being, non-human intelligence would be impossible by definition; if HI is specified in such a “loose” way that the current computer systems are already intelligent by definition, AI will be trivialized and deserves no attention.

This task uniquely belongs to AI theories, because even though there are many studies on the human mind in the past in the related disciplines, little effort is made to separate the conclusions about “intelligence” (or “cognition”, “mind”) in general from those about “*human* intelligence” (or “*human* cognition”, “*human* mind”) in specific.

For example, though there is a huge literature on the psychological study of human intelligence, which is obviously related to AI, an AI theory cannot use the conclusions indiscriminately. This is because the notion of “intelligence” is used in psychology as an

attribute where the difference *among human beings* is studied, while in AI it is an attribute where the difference *between humans and computers* is much more important. Many common properties among human beings are taken for granted in psychology, so they are rarely addressed in psychological theories. On the contrary, these properties are exactly what AI tries to reproduce, so they cannot be omitted in AI theories. For this reason, it is not helpful to directly use human IQ tests to evaluate the intelligence of a computer system. Similarly, the correctness of an AI theory cannot be judged in the same way as a theory in a related empirical discipline, such as psychology.

On the other hand, the human–computer difference cannot be used as an excuse for an AI theory to contain conclusions that are clearly inconsistent with the existing knowledge of the human mind. In the current context, even a theorem proved in a formal theory is not necessarily “correct” as a conclusion about intelligence, unless the axioms of the theory can be justified as acceptable in AI. If a normal human being is not “intelligent” according to an AI theory, then the theory is not really about intelligence as we know it, but about something else. This is especially the case for the GI part of the theory — even though generalization and simplification are necessary and inevitable, overgeneralization and oversimplification can cause serious distortion in a theory, to the extent that it is no longer relevant to the original problem.

For an AI theory to be correct, it does not need to explain every phenomenon of the human mind, but only those that are considered as essential for HI by the theory. Though each theory may select different phenomena, there are some obvious features that should be satisfied by every theory of intelligence. Suggested features are exemplified by the requirement of being *general* [7], or being *adaptive* and can work with *insufficient knowledge and resources* [48].

At the current time, the correctness of an AI theory is usually a matter of degree. The existence of certain counterevidence rarely falsifies a theory completely (as suggested by Popper [37]), though it does decrease its correctness, and therefore its competitiveness when compared with other theories. We will return to this topic later.

Concreteness

The practical value of a scientific theory shows in the guidance it provides to human activities. In the current context, this requirement focuses on the relation between an AI theory (especially the CI part) and the computer systems developed according to it.

Since the objective of AI is to build “thinking machines”, the content of an AI theory needs to be concrete enough to be applicable into system design and development, even though it does not have to specify all the technical details.

This requirement means that a pure descriptive theory about how human intelligence works will not qualify as a good AI theory. In the theoretical discussions of AI and Cognitive Science, there are some theories that sound quite correct. However, they are very general, and use fuzzy and ambiguous concepts, so seem to be able to explain everything. What is missing in these theories, however, is the ability of making *concrete, accurate, and constructive* suggestions on how to build AI systems.

Similarly, it is a serious problem if a theory of AI proposes a design of AI systems, but some key steps in it cannot be implemented — for example, the AIXI model is uncomputable [20]. Such a result cannot be treated as an unfortunate reality about intelligence, because the involved notion of “intelligence” is a construct in the theory, rather than a naturally existing phenomenon objectively described by the theory. The human mind has provided an existing proof for the possibility of intelligence, so there is no reason to generalize it into a notion that cannot be realized in a physical system.

In summary, a good AI theory should include a description of intelligence using the terminology provided by the existing computer science and technology. That is, the theory not only needs to tell people *what should be done*, but also *how to do it*.

“To guide the building of AI systems” does not necessarily mean these systems come with practical problem-solving capability. It again depends on the working definition of intelligence. According some opinion, “intelligence” means to be able to solve human-solvable problems [33], so an AI theory should cover the solutions to various practical problems. However, there are also theories that do not take “intelligence” as problem-solving capability, but learning capability [46]. According to such a theory, when an AI system is just built, it may have little problem-solving ability, like a human baby. What it has is the *potential* to acquire problem-solving ability via its interaction with the environment. The requirement of concreteness allows both of the previous understandings of intelligence — no matter how this concept is interpreted, it needs to be realized in computer systems.

To insist that the CI part of an AI theory must be presented using concrete (even computational) concepts does not mean that the theory of AI can be replaced by the existing theories of computer science. Not all computer systems are intelligent, and AI is a special type of computer systems that is designed according to a special theory. It is just like that a

theory of architecture cannot be replaced by a theory of physics, even though every building is constructed from physical components with physical relations among them. The claim that AI needs no theory beyond computer science [19] cannot explain the obvious difference between the human mind and the conventional computers.

Compactness

While the previous two properties (correctness and concreteness) are about the *external* relation between an AI theory and outside systems (human and computer, respectively), compactness is a property of the *internal* structure of the theory. Here the word “compactness” is used to mean the conceptual simplicity of a theory’s content, not merely on its “size” measured literally.

Since scientific theories are used to guide human behaviors, simple theories are preferred, because they are easier to use and to maintain (to verify, to revise, to extend, etc.). This opinion is well known in various forms, such as “Occam’s Razor” or “Mach’s Economy of Thought”, and is accepted as a cornerstone in several AGI theories [4, 20, 41].

To establish a compact theory in a complicated domain, two common techniques are *axiomatization* and *formalization*.

Axiomatization works by compressing the core of the theory into a small number of fundamental concepts and statements, to be taken as the basic notions and axioms of the theory. The other notions are defined recursively from the basic ones, and the other statements are proved from the axioms as theorems. Consequently, in principle the theory can be reduced to its axioms. Besides efficiency in usage, axiomatization also simplifies the verification of the theory’s consistency and applicability.

Formalization works by representing the notions in a theory by symbols in an artificially formed language, rather than by words in a naturally formed language. Consequently, the notions have relatively clear and unambiguous meaning. The same theory can also be applied to different situations, by giving its symbols different interpretations. Even though it looks unintuitive to outsiders, a formal theory is actually easier to use for various purposes.

Axiomatization and formalization are typically used in mathematics, as well as in logic, computer science, and other normative theories. The same idea can also be applied to empirical science to various degrees, though because the very nature of those theories, they cannot be *fully* axiomatized (because they must open to new evidence) or *fully* formalized

(because their key concepts already have concrete meaning associated, and cannot be taken as symbols waiting to be interpreted).

Since a theory of AI has empirical content, it cannot be fully axiomatized or formalized, neither. Even so, it is still highly desired for it to move in that direction as far as possible, by condensing its empirical content into a small set of assumptions and postulations, then deriving the other part of the theory from it using justifiable inference rules. To a large extent, it is what a “Serious Computational Science” demands, with the requirements of being “cohesive” and “theorem-guided” [7].

16.4 Relations among the properties

To summarize the previous discussions, a good AI theory should provide a *correct* description about how intelligence works (using evidence from human intelligence), give *concrete* instructions on how to produce intelligence in computer systems (using feasible techniques), and have a *compact* internal structure (using partial axiomatization and formalization).

These three requirements are *independent*, in the sense that in general there is no (positive or negative) correlation among them. All the three C’s are desired in a theory, for different reasons, and one cannot be reduced into, or replaced by, the others.

For example, a simpler theory is not necessarily more correct or less correct, when compared with other theories. On this topic, one usual misconception is about Occam’s Razor, which is often phrased as “Simpler theories are preferred, because they are more likely to be correct”. This is not proper, since the original form of this idea was just something like “Simpler theories are preferred”, and it is not hard to find examples where simpler theories are actually less correct. A common source of this misconception is the assumption that the only desired feature of a scientific theory is its correctness (or “truth”) — in that case, if simpler theories are preferred, the preference must come from its correctness. However, generally speaking, compactness (or simplicity) is a feature that is preferred *for its own sake*, rather than as an indicator of correctness. It is like when we compare several products, we prefer cheaper ones when everything else is about the same, though it does not mean that we prefer cheaper products because they usually have higher quality. Here “price” and “quality” are two separate factors influencing our overall preference, and additional information is needed to specify their relationship.⁴

⁴Some people may argue that a simpler theory is more correct because it is less likely to be wrong, but if a theory becomes simpler by saying less, such a simplification will make the theory covers less territory, so it will also

In certain special situations, it is possible for the requirements to be taken as correlated. One such treatment is Solomonoff's "universal prior", which assumes that without domain knowledge, the simpler hypotheses have higher probability to be correct [43]. Though Solomonoff's model of induction has its theoretical and practical values, the soundness of its application to a specific domain depends on whether the assumptions of the model, including the above one, can be satisfied (exactly or approximately) in the domain. For the related AGI models (such as AIXI [20]), such justifications should be provided, rather than taken for granted. After all, we often meet simple explanations of complex phenomena that turn out to be wrong, and Occam's Razor cannot be used as an argument for the *correctness* of a theory (though it can be an argument for why a theory is preferred). For the same reason, a formal theory is not necessarily more correct than an informal one, though the former is indeed preferred when the other features of the two theories are similar.

These three C's are arguably *complete*, because altogether they fully cover the subject matter: the descriptive ingredients of the theory need to be correct, the normative ingredients need to be concrete, and the whole theory needs to be compact. Of course, each of the three can be further specified with more details, while all of them must be possessed by a theory that is about intelligence, rather than only about one part or one aspect of it.

All three C's can only be *relatively* satisfied. As a result, though probably there will not be a *perfect* theory of AI, there are surely *better* theories and *not-so-good* ones. When a theory is superior to another one in all three dimensions, it is "generally better". If it is superior in one aspect, but inferior in another, then whether it is better for the current purpose depends on how big the differences are, as well as on the focus of the comparison. Intuitively speaking, we can think the overall "score" on the competitiveness of an AI theory as a multiplication of the three "scores" it obtains on the three C's, though we do not have numerical measurements for the scores yet. In this way, an acceptable theory must be acceptable in all the three dimensions. Even if a theory is excellent in two aspects, it still can be useless for AI if it is terrible in the third.

16.5 Issues on the properties

In the current theoretical explorations in AGI, a common problem is to focus on some desired property of a theory, while ignoring the others.

have less supporting evidence. To simply remove some conclusions from a theory does not make it more correct, unless "correctness" is defined according to Popper's falsification theory about science [37], so the existence of supporting evidence does not contribute to the correctness of a theory. Such a definition is not accepted here.

Issues on *correctness* typically happen in formal or computational models of intelligence. Sometimes people think as long as they make it clear that a model is based on “idealized assumptions”, they can assume whatever they want (usually the assumptions required by their available theoretical or technical tools). For example, Schmidhuber thought that for AI systems, the assumption of Markovian environments is too strong, so “We will concentrate on a much weaker and therefore much more general assumption, namely, that the environment’s responses are sampled from a computable probability distribution. If even this weak assumption were not true then we could not even formally specify the environment, leave alone writing reasonable scientific papers about it.” [41] It is true that every formal and computational model is based on some idealized assumptions, which are usually never fully satisfied in realistic situations. However, this should not be taken as an excuse to base the model on highly unrealistic assumptions or assumptions that can only be satisfied in special situations. Since the conclusions of the model are largely determined by its assumptions, an improper assumption may completely change the nature of the problem, and consequently the model will not be about “intelligence” (as we know it) at all, but about something else. One cannot force people to accept a new definition of “intelligence” simply because there is a formal or computational model for it — it reminds us the famous remark of Abraham Maslow: “If you only have a hammer, you tend to see every problem as a nail”. We do want AI to become a serious science, but to change the problem into a more “manageable” one is not the way to get there.

On the other hand, to overemphasize correctness at the cost of the other requirements also leads to problems. The “Model Fit Imperative” analyzed by Cassimatis (Chapter 2 of this book) is a typical example. A theory of AI is not responsible for explaining or reproducing all the details of human intelligence. The most biologically (or psychologically) accurate model of the human brain (or mind) is not necessarily the best model for AI.

Issues on *concreteness* typically happen in theories that have rich philosophical content. Though philosophical discussions are inevitable in AI theories, to *only* present a theory at that level of description is often useless, and such a discussion quickly degenerates into word games, which is why among AI researchers “this is a philosophical problem” is often a way to say “It doesn’t matter” or “You can say whatever you want about it”. Similarly, if some theory contains descriptions that nobody knows how to implement or even to approximate, such a theory will not be very useful for AI. Just to solve the AI problem “in principle” is not enough, unless those principles clearly lead to technical decisions in design and development, even if not in all details.

Issues on *compactness* widely exist in AGI projects that are mainly guided by psychological/biological inspirations or problem-solving capabilities. Such a project is usually based on a theory that basically treats intelligence as a collection of “cognitive functions” that are organized into a “cognitive architecture” (see Chapters 7 and 8 of this book).

One problem about this approach is that the functions recognized in the human mind are not necessarily carried out by separate processes or mechanisms. In a psychological theory, sometimes it is reasonable to concentrate on one aspect of intelligence, but such a practice is not always acceptable in an engineering plan to realize intelligence, since to reproduce a single mechanism of intelligence may be impossible, given its dependency on the other mechanisms. For example, “reasoning” and “learning” may be two aspects of the same process [29, 46]; “perceiving” may be better considered as a way of “acting” [34]; “analogy” may be inseparable from “high-level perception” [10].

Though from an engineering point of view, a modular architecture may be used in an AI system, the identification and specification of the modules must follow an AI theory — the functions and modules should be the “theorems” of a theory that are derived from a small number of “axioms” or “principles”, so as to guarantee the coherence and integrity of the system as a whole. Without such an internal structure, a theory of AI looks like a grab bag of ideas — even when each idea in it looks correct and concrete, there is still no guarantee that the ideas are indeed consistent, nor guidance on how to decide if on a design issue different ideas point to different directions. Such an architecture often looks arbitrary, without convincing reason for the partition of the overall function into the modules. Consequently, the engineering practice will probably be full of trial-and-error, which should not happen if the theory is well-organized.

16.6 Conclusion

A major obstacle of progress in AI research is “theoretical nihilism” — facing the well-known difficulty in establishing a theory of AI, the research community as a whole has not made enough effort in this task, but instead either follows some other theories developed for certain related, though very different, problems, or carries out the research based on intuitions or practical considerations, with the hope that the theoretical problems can be eventually solved or avoided using technical tricks.

Though AI is indeed a very hard problem, and it is unlikely to get a perfect (or even satisfactory) theory very soon, to give up on the effort or to depend on an improper substitute

is not a good alternative. Even though the research can go ahead without the guidance of a theory, it may run in a wrong direction, or into dead alleys. Even an imperfect theory is still better than no theory at all, and a theory developed in another domain does not necessarily keep its authority in AI, no matter how successful it is in its original domain.

Given the special situation in the field, an AI theory must be descriptive with respect to the human mind, and be normative with respect to computer systems. To achieve this objective, it should construct a notion of general intelligence, which does not depend in the details of either the biological brain or the electrical computer. The desired properties of such a theory can be summarized by the Three C's: *Correctness*, *Concreteness*, and *Compactness*, and the overall quality of the theory depends on all the three aspects. Among the existing theoretical works, many issues are caused by focusing only on one (or two) of the properties, while largely ignoring the other(s).

To a large extent, the above issues come from the science–engineering duality of AI. A theory of AI is similar to a theory of natural science in certain aspects, while that of engineering in other aspects. We cannot work in this field like typical natural scientists, because “intelligent computers” are not existing phenomena for us to study, but something to be created; on the other hand, we cannot work like typical engineers, because we are not sure what we want to build, but have to find that out by studying the human mind. The theoretical challenge is to find a minimum description of the human mind at a certain level, then, with it as specification, to build computer systems in which people will find most of the features of “intelligence”, in the everyday sense of the word.

Though the task is hard, there is no convincing argument for its impossibility. What the field needs is to spend more energy in theoretical exploration, while keeping a clear idea about what kind of theory we are looking for, which is what this chapter attempts to clarify.

Acknowledgements

Thanks to Joscha Bach for the helpful comments.

Bibliography

- [1] Albus, J. S. (1991). Outline for a theory of intelligence, *IEEE Transactions on Systems, Man, and Cybernetics* **21**, 3, pp. 473–509.
- [2] Alvarado, N., Adams, S. S., Burbeck, S. and Latta, C. (2002). Beyond the Turing Test: Performance metrics for evaluating a computer simulation of the human mind, in *Proceedings of the 2nd International Conference on Development and Learning*, pp. 147–152.

- [3] Bach, J. (2009). *Principles of Synthetic Intelligence PSI: An Architecture of Motivated Cognition* (Oxford University Press, Oxford).
- [4] Baum, E. B. (2004). *What is Thought?* (MIT Press, Cambridge, Massachusetts).
- [5] Brachman, R. J. (2006). (AA)AI — more than the sum of its parts, 2005 AAAI Presidential Address, *AI Magazine* **27**, 4, pp. 19–34.
- [6] Bringsjord, S. (2008). The logicist manifesto: At long last let logic-based artificial intelligence become a field unto itself, *Journal of Applied Logic* **6**, 4, pp. 502–525.
- [7] Bringsjord, S. and Sundar G, N. (2009). Toward a serious computational science of intelligence, Call for Papers for an AGI 2010 Workshop.
- [8] Bundy, A. and Ohlsson, S. (1990). The nature of AI principles, in *The foundation of artificial intelligence—a sourcebook* (Cambridge University Press, New York), pp. 135–154.
- [9] Cassimatis, N. L. (2006). Artificial intelligence and cognitive science have the same problem, in *Papers from the AAAI Spring Symposium on Between a Rock and a Hard Place: Cognitive Science Principles Meet AI-Hard Problems*, pp. 27–32.
- [10] Chalmers, D. J., French, R. M. and Hofstadter, D. R. (1992). High-level perception, representation, and analogy: a critique of artificial intelligence methodology, *Journal of Experimental & Theoretical Artificial Intelligence* **4**, pp. 185–211.
- [11] Chandrasekaran, B. (1990). What kind of information processing is intelligence? in *The foundation of artificial intelligence—a sourcebook* (Cambridge University Press, New York), pp. 14–46.
- [12] Feigenbaum, E. A. and Feldman, J. (1963). *Computers and Thought* (McGraw-Hill, New York).
- [13] Franklin, S. (2007). A foundational architecture for artificial general intelligence, in B. Goertzel and P. Wang (eds.), *Advance of Artificial General Intelligence* (IOS Press, Amsterdam), pp. 36–54.
- [14] Gabbay, D. M. and Woods, J. (2003). Normative models of rational agency: The theoretical disutility of certain approaches, *Logic Journal of the IGPL* **11**, 6, pp. 597–613.
- [15] Goertzel, B. (2009). Toward a general theory of general intelligence, *Dynamical Psychology*, URL: <http://goertzel.org/dynapsyc/dynacon.html#2009>.
- [16] Goertzel, B., Arel, I. and Scheutz, M. (2009). Toward a roadmap for human-level artificial general intelligence: Embedding HLAI systems in broad, approachable, physical or virtual contexts, *Artificial General Intelligence Roadmap Initiative*, URL: <http://www.agi-roadmap.org/images/HLAIR.pdf>.
- [17] Goertzel, B. and Pennachin, C. (eds.) (2007). *Artificial General Intelligence* (Springer, New York).
- [18] Hawkins, J. and Blakeslee, S. (2004). *On Intelligence* (Times Books, New York).
- [19] Hayes, P. and Ford, K. (1995). Turing Test considered harmful, in *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pp. 972–977.
- [20] Hutter, M. (2005). *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability* (Springer, Berlin).
- [21] Kirsh, D. (1991). Foundations of AI: the big issues, *Artificial Intelligence* **47**, pp. 3–30.
- [22] Laird, J. E., Wray, R. E., Marinier, R. P. and Langley, P. (2009). Claims and challenges in evaluating human-level intelligent systems, in *Proceedings of the Second Conference on Artificial General Intelligence*, pp. 91–96.
- [23] Legg, S. and Hutter, M. (2007). Universal intelligence: a definition of machine intelligence, *Minds & Machines* **17**, 4, pp. 391–444.
- [24] Marr, D. (1982). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information* (W. H. Freeman & Co., San Francisco).
- [25] McCarthy, J. (1984). We need better standards for AI research, *AI Magazine* **5**, 3, pp. 7–8.
- [26] McCarthy, J. (1988). Mathematical logic in artificial intelligence, *Dædalus* **117**, 1, pp. 297–311.
- [27] McCarthy, J. (2007). From here to human-level AI, *Artificial Intelligence* **171**, pp. 1174–1182.

- [28] McCarthy, J., Minsky, M., Rochester, N. and Shannon, C. (1955). A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, URL: <http://www-formal.stanford.edu/jmc/history/dartmouth.html>.
- [29] Michalski, R. S. (1993). Inference theory of learning as a conceptual basis for multistrategy learning, *Machine Learning* **11**, pp. 111–151.
- [30] Minsky, M. (1985). *The Society of Mind* (Simon and Schuster, New York).
- [31] Newell, A. (1990). *Unified Theories of Cognition* (Harvard University Press, Cambridge, Massachusetts).
- [32] Newell, A. and Simon, H. A. (1976). Computer science as empirical inquiry: symbols and search, *Communications of the ACM* **19**, 3, pp. 113–126.
- [33] Nilsson, N. J. (2005). Human-level artificial intelligence? Be serious! *AI Magazine* **26**, 4, pp. 68–75.
- [34] Noë, A. (2004). *Action in Perception* (MIT Press, Cambridge, Massachusetts).
- [35] Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems* (Morgan Kaufmann Publishers, San Mateo, California).
- [36] Pollock, J. L. (2006). *Thinking about Acting: Logical Foundations for Rational Decision Making* (Oxford University Press, USA, New York).
- [37] Popper, K. R. (1959). *The Logic of Scientific Discovery* (Basic Books, New York).
- [38] Rumelhart, D. E. and McClelland, J. L. (eds.) (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1, Foundations* (MIT Press, Cambridge, Massachusetts).
- [39] Russell, S. and Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*, 3rd edn. (Prentice Hall, Upper Saddle River, New Jersey).
- [40] Schank, R. C. (1991). Where is the AI? *AI Magazine* **12**, 4, pp. 38–49.
- [41] Schmidhuber, J. (2007). The new AI: General & sound & relevant for physics, in B. Goertzel and C. Pennachin (eds.), *Artificial General Intelligence* (Springer, Berlin), pp. 175–198.
- [42] Simon, T. W. (1990). Artificial methodology meets philosophy, in *The foundation of artificial intelligence—a sourcebook* (Cambridge University Press, New York), pp. 155–164.
- [43] Solomonoff, R. J. (1964). A formal theory of inductive inference. Part I and II, *Information and Control* **7**, 1-2, pp. 1–22, 224–254.
- [44] von Neumann, J. (1958). *The Computer and the Brain* (Yale University Press, New Haven, CT).
- [45] Wang, P. (1995). *Non-Axiomatic Reasoning System: Exploring the Essence of Intelligence*, Ph.D. thesis, Indiana University.
- [46] Wang, P. (2006). *Rigid Flexibility: The Logic of Intelligence* (Springer, Dordrecht).
- [47] Wang, P. (2008). What do you mean by ‘AI’, in *Proceedings of the First Conference on Artificial General Intelligence*, pp. 362–373.
- [48] Wang, P. (2010). A General Theory of Intelligence, An on-line book under development. URL: <http://sites.google.com/site/narswang/EBook>.
- [49] Wang, P. and Goertzel, B. (2007). Introduction: Aspects of artificial general intelligence, in B. Goertzel and P. Wang (eds.), *Advance of Artificial General Intelligence* (IOS Press, Amsterdam), pp. 1–16.
- [50] Wilks, Y. (1990). One small head: models and theories, in *The foundation of artificial intelligence—a sourcebook* (Cambridge University Press, New York), pp. 121–134.

Index

- λ -Calculus, 175, 179, 180, 183, 184, 187, 188, 190, 192
- “Ladder” of Intelligence, 53
- a-priori
 - knowledge, 79
- action, 75
 - human, 72
 - rational, 72
- action selection, 108, 114
- actions, 74, 78
- actuation, 5
- agent, 75
 - artificial, 72
 - building one, 84
 - framework, 72
 - rational, 81
- Agent’s Body, 270
- AGI theories, 3
- AIXI, 311, 315, 318
 - approximation, 82
 - complete, 76
 - CTW, 82
 - definition, 75
 - optimal, 76, 81
 - rational, 76
 - sound, 76
 - theory, 76
 - universal, 76
- Albus, James, 123, 126, 129, 134, 144
- altruism, 80
- analogical reasoning, 25, 44, 45
- analogico-deduction, 25, 27
- analogico-deductive reasoning, 39
- analogy, 7, 219–224, 230, 232–234, 237, 238
- analysis mechanism, 288, 289, 292–298, 300, 302–304
- animal cultural reasoning, 74
- anthropology, 72
- approximation
 - AIXI, 82
- architecture, 5
- Arel, Itamar, 126, 144
- artificial
 - agent, 72
 - general intelligence (AGI), 89, 173
 - intelligence, 68, 81
- artificial intelligence
 - context, 71
 - embodied, 79
 - foundations, 69, 70
 - friendly, 80
 - funding, 68
 - history, 68
 - information theory, 69
 - optimists, 68
 - paradigms, 69
 - pessimists, 68
 - philosophy, 70
 - social questions, 79
 - the dream, 68
 - the problem, 70
 - universal, 74
- asocial
 - ontologies, 58
 - reasoning, 53
- association, 77
- attentional memory (ATM), 107
- attitude, 80
- axiomatization, 316

- Baars' Global Workspace Theory (GWT), 103
 Baars, Bernard, 126, 129, 144
 Bach, Joscha, 123, 126, 130, 144
 Bayes, 75
 Bayesian
 statistics, 81
 behaviorism, 71
 Bellman, 75
 blending, 219, 220, 222–224, 226–228,
 232–238
 bottom-up, 72
 building
 agents, 84

 C-test, 82
 childhood, 80
 Chinese Room, 268
 chronological environment, 76
 civilization
 post-human, 68
 civilization-scale
 ontologies, 62
 reasoning, 55
 classification, 77
 clustering, 77
 CogAff, 126, 144
 cognitive
 architecture, 320
 decision-making, 117
 mechanisms, 219, 220, 222, 232, 236–238
 modeling, 4
 science, 72
 semantics, 299, 300
 synergy, 6, 123, 125, 131, 137–140,
 142–144
 CogPrime, 143
 compactness, 316, 320
 complete
 AIXI, 76
 complexity
 Kolmogorov, 81
 compression
 contest, 83
 prize, 83
 compressor
 intelligent, 83
 smart, 83
 computation, 69
 computationalism, 269
 Computer Intelligence, 309

 computer science, 72
 concreteness, 314, 319
 consciousness, 7, 79, 264, 283–290, 294,
 296–298, 300–304
 constructivist, 6
 contest
 compression, 83
 context
 artificial intelligence, 71
 continuation, 175–177, 179, 181–183, 186–192
 correctness, 313, 319
 creativity, 7, 78, 219–223, 226, 228, 232, 235,
 237, 238, 277
 cross-domain reasoning, 220, 222, 223,
 236–238
 Crows, 60
 CTW
 AIXI, 82
 cultural ontologies, 59
 curiosity, 80

 Dörner, Dietrich, 123, 126, 144
 decision theory, 75, 81
 decisions, 75, 78
 deduction, 73, 78
 Deep Blue, 26
 Deep learning networks, 126, 127
 Deep Machine Learning, 89, 90, 92
 definition
 intelligence, 72, 74, 82
 definition of AGI, 1
 definition of intelligence, 312
 Descartes, 34
 descriptive theory, 308, 311
 deterministic
 environment, 76
 drugs, 80
 dualism, 266

 economics, 72
 ego(t)ism, 80
 eliminativism, 267
 embodiment, 79, 271
 emergent
 intelligence, 77
 emotion, 7
 enlarged mind, 272
 environment, 75
 chronological, 76
 deterministic, 76

- mixture, 76
 - noisy, 76
- Epicurus, 75
- epiphenomenalism, 266
- episodic memories, 106
- evolution, 72
- Experience, 275
- expert systems, 69
- externalist, 272

- facets
 - intelligence, 77
- feature selection
 - reinforcement learning, 83
- Fifth Generation Computer Systems, 1
- formalization, 316
 - intelligence, 83
- foundations
 - artificial intelligence, 69, 70
- framework
 - agent, 72
- Franklin, Stan, 123, 126, 129, 144
- Free Will, 274
- friendly
 - artificial intelligence, 80
- funding
 - artificial intelligence, 68

- g factor, 4
- Gödel Machine, 173–178, 185, 191, 192, 194
- General Intelligence, 310
- general intelligence, 1, 3, 4, 7, 8, 68
- General Problem Solver, 1
- general-purpose systems, 2
- generalization, 77, 220, 223, 224, 226, 227, 231, 232, 237, 313, 314
- generalizations, 219, 220, 222
- global optimality, 176
- goals, 78
- grand goal, 68
- grand unification, 73

- hard problem, 283, 285–287, 296, 297
- Hawkins, Jeff, 126, 144
- Heuristic-Driven Theory Projection (HDTP), 223
- history, 76
 - artificial intelligence, 68
- HTM, 311
- human
 - action, 72
 - identity, 68
 - intelligence, 309, 315
 - knowledge, 83
 - mind, 68
 - thinking, 72
- human intelligence, 4
- human-level intelligence, 5, 68, 91
- hypothetico-deduction, 25

- identity
 - human, 68
- immortality, 80
- induction, 73, 75, 77
 - problem, 81
 - universal, 81
- information, 69, 78
- information integration, 266
- information theory, 81
 - artificial intelligence, 69
- integrative architecture, 6
- integrative diagram, 124–133, 136–140, 143, 144
- intelligence
 - artificial, 68
 - C-test, 82
 - definition, 72, 82
 - emergent, 77
 - facets, 77
 - formal definition, 74
 - formalization, 83
 - general, 68
 - human-level, 68
 - measure, 74, 82
 - order relation, 74
 - philosophy, 83
 - rational, 81
 - test, 82
 - universal, 83
- intelligent
 - compressor, 83
- internalism, 272

- knowledge, 78
 - a-priori, 79
 - human, 83
- Kolmogorov, 75
 - complexity, 81
- ladder of intelligence, 5

- language, 80
- laws of thought, 72
- learning, 69, 79
 - universal, 73
- learning algorithm, 5
- Learning Intelligent Distribution Agent, 103
- LIDA, 103–118, 123, 126, 127, 129–131, 136, 144
- linguistics, 71
- LISA, 38
- Lisp, 175, 183, 184
- literate ontologies, 61
- literate reasoning, 54
- logic, 69, 72, 78

- MacGyver, 27
- Mach's Economy of Thought, 316
- Machine Consciousness, 263
- machine learning, 72
- manipulation, 80
- meaning, 284, 289, 292, 294, 298–300
- measure
 - intelligence, 82
- memory, 78
- meta-circular evaluator, 184–186, 192
- metaphors, 221–225, 236
- mind
 - human, 68
 - philosophy, 71
- mixture environment, 76
- molecular framework, 289–291
- Monte-Carlo AIXI, 82
- motor skills, 80
- multi-memory system, 130, 138

- neural
 - correlate, 266, 286
 - nets, 69
- neuroscience, 71
- noisy environment, 76
- Normative theory, 308
- normative theory, 310, 311, 316
- nurturing environment, 79

- observation, 75
- Occam's Razor, 316–318
- Ockham, 75
- octopus, 50
- ontology, 78
- OpenCog, 125, 126, 132, 144

- optimal
 - AIXI, 76, 81
 - control, 90, 91
- optimists artificial intelligence, 70
- oral linguistic ontologies, 61
- oral linguistic reasoning, 54
- order relation intelligence, 83

- Pacman, 82
- paradigms
 - artificial intelligence, 69
- paradigms artificial intelligence, 68
- Parrots, 60
- pattern recognition, 77
- perception, 5, 75
- Perceptual Associative Memory (PAM), 106, 113
- perceptual symbol systems, 105
- pessimists artificial intelligence, 70
- phenomenology, 285–287, 295–297, 300, 301, 304
- philosophy
 - artificial intelligence, 70
 - intelligence, 83
 - of the mind, 71
- Piaget, 35, 44
- Piaget-MacGyver, 25
- planning, 78
- Poker, 82
- post-human
 - civilization, 68
- prediction, 77
- predictions, 75
- primary oral cultures, 61
- prize
 - compression, 83
- probability, 69
- problem
 - induction, 81
- problem solving, 78
- procedural memory, 108
- procreation, 80
- proof search, 178
- Psi, 123, 126, 129–131
- psychology, 71, 313
- Psychometric AGI, 25, 28
- psychopath, 80

- qualia, 285, 288, 294, 296, 302, 303

- rational
 - action, 72
 - agent, 81
 - AIXI, 76
 - intelligence, 81
 - thinking, 72
- reality, 288
- reasoning, 78
- recognition, 6
- reductionist, 266
- reflection, 178, 183, 186, 187, 191
- Reinforcement Learning, 89–91, 95–97, 99, 101, 102, 175
- reinforcement learning, 5, 73, 79, 81
 - feature selection, 83
- reward, 75
 - shaping, 80
- scheduler, 177, 185
- Scheme, 175, 183–188, 190, 191, 194
- schooling, 80
- search
 - universal, 82
- selection, 313
- self-awareness, 79
- self-improvement, 80
- self-modification, 6
- self-modifying code, 174
- self-preservation, 80
- self-reflection, 173, 174, 178, 179, 183–185, 191, 192
- sensory input, 79
- sensory-motor memory, 108
- sequence
 - training, 80
- shaping
 - reward, 80
- situated cognition, 104
- situatedness, 271
- Slate, 39
- Sloman, Aaron, 123, 126, 128, 129
- smart
 - compressor, 83
- social
 - reasoning, 54
- social behavior
 - super-intelligence, 80
- social ontologies, 59
- social questions, 79
- socializing, 80
- soft approaches, 69
- Solomonoff, 75, 318
- sound
 - AIXI, 76
- state of the art
 - UAI, 81
- statistics
 - Bayesian, 81
- stigmergy, 59
- suicide, 80
- Super String theory, 73
- super-intelligence
 - social behavior, 80
- supervised learning, 79
- Symbol Grounding Problem, 55
- system, 72
- target theorem, 175, 177, 178
- teacher, 80
- test
 - intelligence, 82
 - Turing, 72, 82
- the AI problem, 70
- the problem
 - artificial intelligence, 70
- theory
 - AIXI, 76
 - UAI, 81
- theory of AI, 306
- thinking
 - human, 72
 - rational, 72
- three-hierarchy model of intelligence, 123, 126
- TicTacToe, 82
- Time, 273
- top-down, 72, 73
- training
 - sequence, 80
- trickiness, 123, 125, 140, 142–144
- Turing, 75
 - test, 72, 82
- UAI
 - state of the art, 81
 - theory, 81
- Unified Theories of Cognition, 307
- universal
 - AIXI, 76
 - artificial intelligence, 74
 - induction, 81

- intelligence, 83
- learning, 73
- search, 82
- universal artificial intelligence, 174
- universal prior, 318
- utility function, 174–177

vision, 80

WATSON, 63

Watson, 26

zombie, 285