Declarative/Logic-Based Computational Cognitive Modeling[*]

Selmer Bringsjord
Rensselaer AI & Reasoning (RAIR) Lab
Department of Cognitive Science
Department of Computer Science
Rensselaer Polytechnic Institute (RPI)
Troy NY 12180 USA
`selmer@rpi.edu`

draft of 031607.0108NY

Abstract

This chapter is an esemplastic systematization of declarative computational cognitive modeling, a field that cuts across cognitive modeling based on cognitive architectures (such as ACT-R, Soar, and Clarion), human-level artificial intelligence (AI), logic itself, and psychology of reasoning (especially of the computational kind). The hallmarks of declarative computational cognitive modeling are the following two intertwined constraints: (1) The central units of information used in the approach are (at least in significant part) declarative in nature, and the central process carried out over these units is inference. (2) The approach to modeling the mind is top-down, rather than bottom-up. (These two points are interconnected because once one commits to (1), (2) becomes quite unavoidable, since bottom-up processing in the brain, as reflected in relevant formalisms (e.g., artificial neural networks), is based on units of information that are numerical, not declarative.) The systematization of declarative computational cognitive modeling is achieved by using formal logic, and hence declarative computational cognitive modeling, from the formal perspective, becomes *logic-based* computational cognitive modeling (LCCM).

The chapter covers some prior research that falls under LCCM; this research has been carried out by such thinkers as Johnson-Laird, Langley, Rips, Simon, and Sun. The material that follows is introductory in nature, and self-contained; it assumes only a modicum of previous exposure to discrete mathematics and computability theory. The key formal elements of LCCM are (a) a generalization of the concept of a *logical system*, central to mathematical logic, and (b) computing in such systems in the declarative programming paradigm, via logic-based computer programs, which are generalized versions of *logic programs* from mathematical logic and computer science. In LCCM, a (logic-based) computational cognitive model of some (or all) human cognition amounts to the execution of a logic-based computer program P_L in the context of a logical system selected from a family $\mathcal{F}$ of such systems. LCCM is designed to meet a number of challenges facing those wishing to devise computational simulations of human cognition. Three such challenges are discussed in the present chapter: the need to model and simulate sophisticated human reasoning (of the three, the one emphasized herein); the need to formulate a *transparently* unified theory of cognition; and the apparent need to achieve significant rigor in the computational simulation of human cognition — rigor which, in LCCM, emerges naturally from providing a formal syntax and semantics that precisely determines the structure and meaning of a logical system used to represent some part of human cognition, and determines as well the meaning of a computational cognitive model, or simulation, on the strength of the fact that the meaning of a logic-based computer program is easily made precise. The gain in precision offered by LCCM enables this field to be, like physics, mathematics, logic, and computer science, theorem-guided. Many regard such guidance to be desirable.

[*]Special thanks are due to my friend and colleague Konstantine Arkoudas for myriad suggestions, and for implemented systems that help make declarative computational cognitive modeling a concrete, rigorous reality.

Contents

1 Introduction

1.1 What is Logic-Based Computational Cognitive Modeling — In a Word?

This chapter is an esemplastic systematization of a particular approach to modeling the mind: *declarative* computational cognitive modeling. (In light of the fact that if an agent knows p, p must be a proposition or declarative statement, sometimes the term 'knowledge-based' is used in place of 'declarative.' Some writers even use the dangerously equivocal term 'symbolic.') Naturally enough, the basic units of such modeling are declarative in nature, or propositional: they are formal objects naturally associated with those particular sentences or expressions in natural languages (like English, German, Chinese) that are declarative statements (as opposed to expressions in the imperative or inquisitive mode) naturally taking values such as TRUE, FALSE, UNKNOWN, PROBABLE (sometimes to particular numerical degrees), and so on. The basic process over such units is inference, which may be deductive, inductive, probabilistic, abductive, or analogical. Because the basic units of declarative computational cognitive modeling are declarative, a hallmark of declarative computational cognitive modeling is a top-down, rather than bottom-up, approach. As Brachman & Levesque (2004) put it, when speaking of declarative computational cognitive modeling within the field of artificial intelligence:

> It is at the very core of a radical idea about how to understand intelligence: instead of trying to understand or build brains from the *bottom up*, we try to understand or build intelligent behavior from the *top down*. In particular, we ask what an agent would need to know in order to behave intelligently, and what computational mechanisms could allow this knowledge to be made available to the agent as required. (Brachman & Levesque 2004, p. iv)

The top-down approach is unavoidable, because, as reflected in relevant formalisms commonly associated with bottom-up approaches (e.g., artificial neural networks), the basic units in bottom-up processing are numerical, not declarative. The systematization of declarative computational cognitive modeling, which is the overarching purpose of the present chapter, is achieved by using formal logic, and hence declarative computational cognitive modeling, from the formal perspective, becomes *logic-based* computational cognitive modeling, sometimes abbreviated below to ease exposition as 'LCCM.' Correspondingly, to decrease verbosity and repetition of the phrase, 'computational cognitive modeling' will sometimes be abbreviated below as 'CCM.'

Logic-based computational cognitive modeling is an interdisciplinary field that cuts across: cognitive modeling based on cognitive architectures (such as ACT-R, Soar, CLARION, Polyscheme, etc.), logic itself, and computational psychology of reasoning. In addition, LCCM has a sister in logic-based human-level artificial intelligence (AI), and, being computational in nature, it inevitably draws heavily from computer science, which is itself, as has been explained (e.g., in Halpern, Harper, Immerman, Kolaitis, Vardi & Vianu 2001), based on formal logic. Specifically, and unsurprisingly, the declarative programming paradigm is naturally associated with declarative computational cognitive modeling. This paradigm, specifically as it applies to LCCM, will be explained later.

1.2 Absence of Advocacy in this Chapter

The present chapter is entirely free of advocacy; its purpose has nothing to do with supporting one group of practitioners in computational cognitive modeling over another, or one paradigm for computational cognitive modeling over another, or one particular cognitive architecture over others. Logic-based computational cognitive modeling, as set out herein, is not intended to be a description of the day-to-day *practice* of all cognitive modelers operating under the umbrella of declarative computational cognitive modeling. Such practice is dizzyingly heterogeneous. Logic-based computational cognitive modeling, as explicated herein, is directly analogous to the systematization of mathematics provided by many decades of formal exposition in books authored by Bourbaki[1] — exposition that shows, formally speaking, that discovery and confirmation

[1]A group allonym for the mathematicians who authored a collection of eight painstakingly rigorous, detailed books showing that all the archival results of mathematics, when pursued with a desire to divine and specify the underlying structure of these results, is fundamentally a derivation from axiomatic set theory using the logical system known as first-order logic, which is $\mathcal{L}_I$ in the family $\mathcal{F}$ of systems introduced and explained in the present chapter. An overview of the Bourbaki oeuvre would defeat its very purpose: Interested readers can start with the first volume: (Bourbaki 2004).

in mathematics consists, fundamentally, in the derivation and use of theorems all extractable from a small set of axioms (e.g., the Zermelo-Fraenkel axioms for set theory). The parallel in the present chapter is that all declarative computational cognitive modeling, formally speaking, is fundamentally the use of logical systems (as defined below) and logic-based computer programs (as also defined below) to model the human mind.

That all the archival products produced in mathematics are now known to consist, fundamentally and formally, of precise reasoning over a set of axioms of set theory is perfectly consistent with the fact that some mathematicians, in their day-to-day practice, exploit diagrams and sketches, whereas others have an exclusively linguistic orientation. Both groups are united, however, by the underlying formal structures that they are using and exploring. Likewise, in contemporary declarative computational cognitive modeling, one researcher may in daily practice use production rules, and another first-order logic, and another graphs to record probability distributions across declarative statements, and another semantic models, and yet another semantic networks, but they are all united by the fact that the structures and processes they produce, are all and only, at bottom, the formal structures explicated in this chapter. In a search for ultimate rigor and generality in the area of computational cognitive modeling that is declarative in nature, logic-based computational cognitive modeling marks the arrival at the desired destination.

The absence of advocacy characterizing the present chapter is seen also in the fact that the purpose of this chapter is certainly not to introduce a new competitor to extant, mature computational cognitive architectures such as Soar (Rosenbloom, Laird & Newell 1993), ACT-R (Anderson 1993, Anderson & Lebiere 1998, Anderson & Lebiere 2003), Clarion (Sun 2001), Icarus (Langley, McKusick, Allen, Iba & Thompson 1991), SNePS (Shapiro & Rapaport 1987), and Polyscheme (Cassimatis 2002, Cassimatis, Trafton, Schultz & Bugajska 2004), nor to declarative computational simulations of parts of human cognition, such as PSYCOP (Rips 1994), and programs written by Johnson-Laird and others to simulate various aspects of so-called mental models-based reasoning (a review is provided in Bucciarelli & Johnson-Laird 1999). These systems are all pitched at a level well above LCCM. Rather, again, the purpose of the present effort is to describe, systematically, what underlies the declarative approach to computational cognitive modeling exemplified by (at least significant parts of) these computational cognitive architectures and simulations, which partake, at least in part, of declarative representations and reasoning over these representations. In fact, the formal umbrella used for the systematization is such as to offer a way to understand and rationalize *all* computational cognitive architectures that are declarative; that is, that are, at least in part, rule-based, explicitly logic-based, predicate-and-argument-based, propositional, production-rule-based, and so on. The ancient roots of this kind of work, as we shall shortly see, run back to Aristotle. Though his theory of the syllogism was invented before the modern notion of a logical system, we can easily see that this theory, formally and foundationally speaking, *is* a (simple) logical system, and that LCCM, as set out in this chapter, therefore underpins Aristotle's declarative modeling. A parallel situation obtains between the content of this chapter and the specific architectures of today. Of course, some architectures overtly use some logic-based elements (e.g., Icarus and Polyscheme), and the present chapter thus directly provides a formal generalization of parts of such architectures.[2]

The formal foundations of declarative computational cognitive modeling, as will be seen, are remarkably simple: they rest only on a generalization of (a) the concept of *logical system*, used in mathematical logic, and (b) the notions of reasoning and computing in such systems, by way of logic-based computer programs. A computational simulation of some human cognition amounts to the execution of such a program in the context of certain selected parameters, where these parameters determine which logical system is operative. All of this will be explained in due course.

1.3 The Ancient Roots of LCCM

Declarative computational cognitive modeling is the oldest paradigm for modeling the mind. As shown in the standard timelines on such matters, over 300 years BC, and hence many, many centuries before the arrival of such things as probability theory and artificial neural networks, logic and logic alone was being used to model and predict human cognition. For example, consider the following argument:

[2]Another overtly logic-based computational cognitive architecture is the Rensselaer Advanced Synthetic Character Architecture for Living Systems, or just RASCALS, used in the Rensselaer AI & Reasoning Lab to build advanced synthetic characters for education and entertainment, such as the character known simply as 'E.' A summary of some of the research and engineering associated with RASCALS and E is found in (Bringsjord, Khemlani, Arkoudas, McEvoy, Destefano & Daigle 2005). The challenge of building advanced synthetic characters in these domains is discussed in this paper as well.

(1) All professors are pusillanimous people.
(2) All pusillanimous people are proud.
$\therefore$ (3) All professors are proud.

The symbol '$\therefore$', often read as 'therefore,' says that statement (3) can be logically inferred from statements (1) and (2); or in other words that, if statements (1) and (2) are true, then (3) must be true as well. Is that so? The odds are exceedingly good that you will see the answer is "Yes." The cognition that consists in your assimilating this argument, declaring it valid, and — were you requested to do so — providing a proof to justify your response, was modeled and predicted by Aristotle.[3] To use today's well-understood concept, which will soon turn out to be central to the present chapter, Aristotle's modeling was expressed in a primitive *logical system*. This system was the theory of the syllogism, according to which the schema

(1*) All As are Bs.
(2*) All Bs are Cs.
$\therefore$ (3*) All As are Cs.

is deductively valid, no matter what classes are denoted by A, B, and C. According to Aristotle, if you were now to be presented with an instantiation of this schema different from the one given about professors (e.g., if A = 'pigeons,' B = 'pongid,' C = 'smart') you would respond that it, too, is a valid inference (and you would of course be correct again). The remarkable thing about your response in the second case is that you will grasp the logical validity of the inference in question, despite the fact that, necessarily, no pigeons are pongid. In other words, Aristotle discovered that certain context-independent structures describe and predict human thinking: you don't assent to the second argument because you know the *meaning* of 'pigeon' and 'pongid,' but rather because you grasp that the abstract structure of the argument is what makes it a valid inference. Because computation was in its infancy 300 BC, and the concept of a general-purpose programmable computer would have to wait until logic made enough progress to give birth to it, it was far from clear to Aristotle how the schemas in his logical system were computational in nature, but in essence he had indeed presented a series of parameterized functions for computing the composite function from triples of formulas in the formal language he invented, to the set {VALID, INVALID.} If the function is s, then since the formulas are all and only of four types, viz.,

English:	All As are Bs.	No As are Bs.	Some As are Bs.	Some As are non-Bs.
Abbreviation:	All AB	No AB	I AB	O $A\bar{B}$

we can say that s returns VALID on the triple (All AB, All BC, All AC), with substitutions for $A - C$. For another example, notice that s returns VALID on the triple (I AB, All BC, I AC). Later, an INVALID triple will turn out to be relevant to modern-day research in psychology of reasoning (section 3.1.4).

Today, using modern experimental design and statistical analysis for the behavioral sciences, a large amount of data has been accumulated in support of large parts of Aristotle's model (e.g., see Newstead & Evans 1995). However, there are two serious problems with the theory of the syllogism. These two problems are in fact the main drivers that have brought LCCM to the level of maturity it enjoys today, and explaining the solution to them forms the heart of the present chapter, as will soon be seen. They are:

Problem 1 Some humans don't reason in normatively correct fashion. Tied to the ancient theory at hand, some human subjects fail to reason in conformity to valid syllogisms (i.e., to s), and in fact sometimes reason in conformity to provably *in*valid syllogisms. Aristotle, and his successors in the LCCM paradigm all the way up to and including Piaget (who held that in the course of normal development humans would acquire a capacity to think not only in accordance with the theory of the syllogism, but with the much more expressive, powerful, and complicated modern logical system known as first-order logic (Inhelder & Piaget 1958)), failed to realize this. The realization came when, in the 20th century AC, Wason and Johnson-Laird showed that normatively correct thinking is in surprisingly short supply among humans (see, e.g., Wason 1966), as can be seen when clever stimuli are devised and presented. (We visit such stimuli later, in section 3.1.4. They are syllogisms which, by s, are classified as INVALID, and yet many humans report that they are VALID.)

Problem 2 The theory of the syllogism was agreed by the relevant thinkers, even at the time of Aristotle, to be at best a model of only a smidgeon of the parts of human cognition that are obvious targets for declarative modeling (e.g., the specification of proofs, as routinely carried out by mathematicians). The specific evidence

[3]Aristotle's work on logic, including the theory of the syllogism, can be found in his *Organon*, the collection of his stunningly seminal logical treatises. This collection, and Aristotle's other main writings, are available in (McKeon 1941).

that gave rise to this agreement consisted of the brute fact that only a tiny part of Euclid's seminal logical and mathematical reasoning, published in his *Elements*, could be modeled as syllogistic reasoning.[4] Today, courtesy of modern logic, LCCM can model all that Euclid did — and more, as shall be seen.

1.4 LCCM's Brother: Logic-Based Human-Level AI

As stated at the outset, logic-based computational cognitive modeling cuts across a number of fields, draws deeply from computer science, and has, in logic-based human-level AI, a sister field. The purpose of the present chapter is to present and explain LCCM itself, not to characterize in depth the fields with which it intersects, draws from, or parallels. (That said, some of these fields, in what follows, will to a degree be explored. For example, in the course of explaining LCCM, the reader will learn quite a bit about logic in and of itself.) Nonetheless, in the case of its sister, it does make sense to say a few words, because the comparison sheds some light on logic-based computational cognitive modeling itself.

AI is the field devoted to building intelligent agents that map percepts (perceived information about the agent's environment) to actions that cause changes in the agent's environment, in the service of goals desired by the agent (Russell & Norvig 2002). This definition is consistent with attempts to build agents having no more intelligence than, say, an insect. (Some famous AI engineers have in fact strived to build robotic insects. Brooks 1991 is an example.) *Human-level* AI, as you can no doubt surmise, is AI focused not on insects, but on intelligent agents capable of human-level behavior. Recently a recrudescence of this form of AI has begun, as a number of writings confirm (e.g., see Cassimatis 2006, Nilsson 1995, Nilsson 2005, Brooks, Breazeal, Marjanovic, Scassellati & Williamson 1999). Of the authors just cited, Nilsson avowedly pursues logic-based human-level AI, while Brooks avowedly does not; Cassimatis straddles both camps.

How are logic-based computational cognitive modeling and human-level logic-based AI related? How similar are they? What makes them different? The encapsulated answer is straightforward: The two fields are largely based upon the same formalisms, both exploit the power of general-purpose programmable computing machines to process symbolic data, but LCCM targets computational simulations of human cognition, whereas human-level logic-based AI, as you might expect, strives to build beings that, at least behaviorally speaking, can pass for humans. While it's conceivable that both fields might well be on the same exact path (one that leads to building a computational system indistinguishable from a human), LCCM insists that the engineered system, at some suitably selected level of description, operate as a human does. Human-level AI would be content with artifacts that *seem* human, but "under the hood" really aren't. As to shared formalisms, interested readers are directed to treatments of logic-based AI that introduce the relevant technical material (summarized e.g. in Bringsjord & Ferrucci 1998*a*, Bringsjord & Ferrucci 1998*b*, Nilsson 1991). The present chapter provides more modern, systematic, and comprehensive treatment of the underlying formal content than provided in these publications.

1.5 Different Levels of Description

This chapter is based upon an ecumenical conception of what it is to computationally model human thinking, particularly human reasoning over declarative content. (Because of space constraints, the exposition herein leaves aside psychology of decision making, despite the fact this discipline is highly declarative, as revealed by the fact that seminal experiments in the field present subjects with declarative statements to be reasoned over, in order for decisions to be expressed. For exemplars, see the experiments carried out by Kahneman and Tversky to establish the so-called "framing effect." Nice coverage is provided in (Kahneman & Tversky 2000).) To explain, consider the structure of the kind of experiments traditionally used in psychology of reasoning.[5] Let S be some stimulus in some experiment involving a person (subject) P, and specifically assume that S is constituted by a list L of declarative statements, a query (or 'stem,' to use the argot of psychometrics) Q, and possibly a single declarative statement D to which Q refers. (If there is no D in S, then Q is simply: "What logically follows from L?") The last ingredient is simply a request for a justification. For example, one might present to P a stimulus such as the following.

[4]These issues are nicely chronicled by Glymour (1992).

[5]For a more thorough treatment of this structure, see (Bringsjord & Yang 2003). For prominent use of this structure in psychology of reasoning, one can read nearly any experiment-based work in that field. For an example of the structure in action, on a topic that bears directly on the present chapter, see e.g. (Johnson-Laird, Legrenzi, Girotto & Legrenzi 2000).

Consider L. $Q =$ Does the following proposition logically follow from L? D. Please provide a justification for your answer.

Now suppose that P gives a verdict ("Yes" or "No"), and provides justification J. In order to achieve a computational simulation of P in this context, given the inclusive orientation of this chapter, it suffices to produce a computer program that takes in S, produces the relevant verdict (e.g., the verdict given by the vast majority of subjects, the normatively correct verdict, etc.), and gives a proof or argument that matches the justification given. (This is not easy to do, because humans often give justifications, especially when erroneous, that depart considerably from established machine reasoning patterns.) The proof or argument itself, in the logic-based paradigm, constitutes the algorithm for transforming the stimulus in question into the output. Put in a way connected to traditional accounts of levels of description found in cognitive science, logic-based computational cognitive models are intended to be successful at Marr's (1982) *algorithmic* level, or Pylyshyn's (1984) *symbolic* level. In addition, please note that it is perfectly acceptable that justification be articulated by subjects on the basis of introspection, as long as established empirical techniques are used, such as verbal protocol analysis (Ericsson & Simon 1984). In the sequel, when we consider a series of specific puzzles as stimuli (in section 3.1), we suppress, in the interests of space, and consistent with the formal orientation of the present chapter, details concerning the normatively correct and incorrect justifications typically provided by subjects.

1.6 The Three Challenges, Briefly

Logic-based computational cognitive modeling addresses a number of challenges to the overall aim of computationally modeling human cognition. Three of them are discussed in this chapter, with emphasis falling on the first. The exposition that follows revolves around these three challenges to computational cognitive modeling. They are:

(C1) Human reasoning, though — in its primitive forms — uncovered and charted through decades of research in psychology of reasoning and psychology of decision making, and — in its more mature forms — through advances in the closely connected fields of logic, formal philosophy, mathematics, (parts of) economics, and computer science (the so-called *formal sciences*[6]), has for the most part not been modeled and computationally simulated in declarative computational cognitive modeling, as evidenced, for example, by what has been modeled in the declarative computational cognitive architectures associated with LCCM.

(C2) While a number of computational cognitive architectures have been developed in the striving for Newell's (Newell 1973, Newell 1990) original dream of providing a unified computational account of human cognition, the core underlying mechanisms that they each individually offer (e.g., production rules, representation and reasoning in the propositional or predicate calculus, Bayesian networks, artificial neural networks) seem to be insufficiently powerful for the task, for either of two reasons: Either the core mechanism, while logic-based, is insufficiently expressive to model the kind of sophisticated human reasoning referred to in (C1) (as happens, e.g., if the core mechanism for representation and reasoning is at the level of the propositional calculus or first-order logic); or the core mechanism, by its very nature, cannot directly model the high-level human reasoning referred to in (C1) (as happens in the case of neural networks and others non-declarative mechanisms). What is needed is a core mechanism that is *transparently* able to range from high-level reasoning and meta-reasoning, down to perception of, and action on, the external environment. This mechanism would constitute the comprehensive "logico-mathematical language" Ron Sun (2001) has said is missing in computational cognitive modeling.

(C3) The languages that most computational cognitive architectures (whether declarative or not) use for writing simulations do not have a clear and precise syntax and semantics, and the field of computational cognitive modeling is (with a few exceptions) bereft of theorems that could guide and inform the field. (Of course, some computational cognitive modelers may not want to be guided by theorems, as those in computer science and physics are. This issue is addressed later.) This adds another degree of vagueness to a field that is already quite nebulous by the standards of established, rigorous, theorem-based sciences (such as physics, (parts of) economics, computer science, mathematics, and logic itself). By contrast, models in logic-based computational cognitive modeling are not only fully declarative, but their meaning is mathematically precise by virtue of the formal syntax and semantics that is part and parcel of the logical systems on which they are based. In addition, LCCM is guided by theorems, and the promise of new ones in the future.

1.7 Structure of the Chapter

This chapter has the following structure. In the next section, (2), the context for logic-based computational cognitive modeling is set by taking note of the overarching goal of this field: the computational modeling of human personhood. In section 3, the three challenges (C1)–(C3) are described in more detail (again, emphasis is on (C1)). In section 4, the logico-mathematical foundation for LCCM is presented: a straightforward generalization of the concept of a logical system, as used in mathematical logic. As is explained (section 4.1), depending upon what aspect of human cognition is to be modeled and simulated, the appropriate logical system is selected. As to computation, that is handled by logic-based computer programs. Once the cognitive modeler has selected the appropriate logical system, a logic-based program relative to that selection is written, and of course executed. The execution produces a computational simulation of the cognition under scrutiny.[7]

Using the logico-mathematical foundation for logic-based computational cognitive modeling, the next section (5) explains how LCCM addresses the three aforementioned challenges.

In the penultimate section (6), the future and limitations of computational cognitive modeling are briefly discussed, in the context of what has been presented in this chapter.

The chapter ends with a brief conclusion, in which it is pointed out that while this chapter has introduced the reader to a frontier (modeling human cognition in unified fashion through logical systems and corresponding computer programs), logic requires issuance of the reminder that no argument has been presented in support of the claim that that frontier should be explored. In particular, this chapter is agnostic on whether one gains something by seeking to unify phenomena that can be separately formalized and modeled.

2 The Goal of CCM/LCCM

The goal of computational cognitive modeling (and by immediate implication, the goal of declarative computational cognitive modeling and systematization thereof in LCCM) is to understand the kind of cognition distinctive of human persons by modeling this cognition in information processing systems of some sort.[8] But how is one to understand 'human cognition,' pre-analytically? In other words, what is the field of computational cognitive modeling's provisional account of what it means to be creatures like us? Or to put the point in its simplest form: What are modelers seeking to model?

Clearly, no computational cognitive model is provided by merely noting the particular DNA structure of humans. When it is said that x is human just in case x has a particular genetic code, the perspective is not that of computational cognitive modeling. Likewise, our minds aren't modeled by charting the physiology of our brains. (After all, computational cognitive modeling is committed to the dogma that simulations can be produced in silicon-based substrates, not carbon-based ones.) Rather, computational cognitive modelers are asking what it means to be a human being, from the *psychological*, and indeed specifically the *cognitive*, perspective. That is, the question is: What does it mean to be a human *person*? For ambitious AI, the CCM's sister, the centrality of personhood is plain in the relevant literature. For example, here is the more than two-decade-old objective for AI announced by Charniak and McDermott (1985):

> The ultimate goal of AI, which we are very far from achieving, is to build a person, or, more humbly, an animal. (Charniak & McDermott 1985, p. 7)

One definition of human personhood has been proposed, defended, and employed by Bringsjord (1997, 2000). This definition is a fairly standard one; for example, it generally coincides with one given by Dennett (1978), and by others as well, for example Chisholm (1978, **?**). In addition, this definition is in line with the capacities covered, chapter by chapter and topic by topic, in surveys of cognitive psychology (e.g., see

[7]Human cognition over an interval of time may well shift between different logical systems, but in the interests of space this possibility is ignored in the present chapter. Such "fluid" logic-based processing will in the near future almost certainly be a very exciting and fertile area for LCCM and its relative, logic-based human-level AI.

[8]This chapter is confined to information processing systems that are no more powerful than Turing machines (= systems at or below the 'Turing Limit'). So-called *hypercomputers* (Bringsjord & Zenzen 2003) (a) are bona-fide machines (they are without question information-processing systems); (b) might be physically realizable; and (c) could, according to some authors, be profitably employed in modeling human cognition. For coverage of the mathematics of information processing above the Turing Limit, see, e.g., (Siegelmann 1995, Siegelmann & Sontag 1994, Siegelmann 1999, Bringsjord & Zenzen 2003, Hamkins & Lewis 2000).

Goldstein 2005, Ashcraft 1994). This definition essentially amounts to the view that x is a person if and only if x has the *capacity*

1. to "will," to make choices and decisions, set plans and projects — autonomously;
2. for consciousness, for experiencing pain and sorrow and happiness, and a thousand other emotions — love, passion, gratitude, and so on;
3. for *self*-consciousness, for being aware of his/her states of mind, inclinations, preferences, etc., and for grasping the concept of him/herself;
4. to communicate through a language;
5. to know things and believe things, and to believe things about what others believe (second-order beliefs), and to believe things about what others believe about one's beliefs (third-order beliefs), and so on;
6. to desire not only particular objects and events, but also changes in his or her character;
7. to reason (for example, in the fashion exhibited in the writing and reading/studying of this very chapter).

Given this list, which as you can see is indeed psychologically oriented, CCM and LCCM are fields devoted to capturing these seven capacities in computation.[9] This position on the ultimate objective of LCCM and CCM meshes seamlessly with a recent account of what CCM is shooting for given by Anderson & Lebiere (2003), who, instead of defining personhood, give an operational equivalent of this definition by describing "Newell's Program," an attempt to build computational simulations of human-level intelligence, where that intelligence is cashed out in the form of a list of abilities that correspond to those on the list just given. For example, part of Newell's Program is to build a computational simulation of natural-language communication at the normal, adult level. This is attribute 4 on the list above. As Anderson & Lebiere (2003) concede, CCM (whether or not logic-based in nature) is finding it rather difficult to mechanically simulate this attribute.

Attribute 4 isn't the only sticking point. An even more challenging problem is consciousness, the representation of which in third-person machine terms remains elusive (Yang & Bringsjord 2003, Bringsjord 1998*a*, Bringsjord 2001, Bringsjord 1995, Bringsjord 1999).

In this chapter, as the reader by now realizes, the emphasis is on attribute 7. Some of the other attributes are ones LCCM can apparently handle, as shown elsewhere. For example, the simulation of attribute 5 in accordance with the LCCM paradigm would seem attainable in light of the fact that this attribute, from the standpoint of AI, has been partially attained via the formalization and implementation given in (Arkoudas & Bringsjord 2005).

3 Three Challenges Facing Computational Cognitive Modeling

In this section a more detailed account of the three aforementioned challenges is provided. In accordance with the plan laid down above, once these problems have been presented, the reader will be in position to proceed to the next section, a description of logic-based computational cognitive modeling, and an explanation of how it is that this sub-field promises to meet the trio. Again, emphasis is decidedly on Challenge 1.

3.1 Challenge 1 (C1): CCM Data from Psychology of Reasoning

At least for the most part, computational cognitive architectures have not been designed in the light of what psychology of reasoning has taught us over many productive decades of empirical research, stretching back to Piaget. In addition, whereas computer science and AI have been driven by the powerful use of logic (Halpern et al. 2001), which is after all the science of reasoning, computational cognitive modeling, whether or not of the declarative type, has largely ignored powerful human reasoning. (Notice that it is said: *powerful* human reasoning. Such reasoning is normatively correct, and sometimes produces significant, publication-worthy results. When the reasoning in question is *not* powerful, it should be noted that Ron Sun's (2001) Clarion cognitive architecture, discussed separately below, has been used to model some human reasoning.) For example, there is no denying that while computer science has produced software capable of discovering non-trivial proofs, no such thing can be accomplished by any computational cognitive architecture. The situation

[9]Where 'computation' as used here covers standard Turing-level computation, and only *possibly* hyper-computation above the Turing Limit as well. See note 8.

is even more odd given that other parts — arguably *all* other parts — of cognitive psychology have been viewed as offering guidance and constraints to and on computational cognitive modeling.

The first challenge, (C1), is now expressed as a series of desiderata that any acceptable computational cognitive architecture would need to satisfy. Some of these desiderata come in the form of specific puzzles expressed in accordance with the experimental structure set out in section 1, where the challenge is to model the cognition (both normatively correct and incorrect) catalyzed by the attempt to solve these puzzles. The section ends with answers to two questions that will occur to a number of readers having some familiarity with psychology of reasoning and computational cognitive modeling.

3.1.1 Desideratum #1: Modeling System 1 vs System 2 Cognition

In an wide-ranging paper in *Behavioral and Brain Sciences*, Stanovich & West (2000) explain that there are two dichotomous systems for thinking at play in the human mind: what they call System 1 and System 2.[10] Reasoning performed on the basis of System 1 thinking is bound to concrete contexts and is prone to error; reasoning on the basis of System 2 cognition "abstracts complex situations into canonical representations that are stripped of context" (Stanovich & West 2000, p. 662), and when such reasoning is mastered, the human is armed with powerful techniques that can be used to handle the increasingly abstract challenges of the modern, symbol-driven marketplace. But before considering these challenges, it's wise to get a better handle on System 1 versus System 2 reasoning.

Psychologists have devised many tasks to illuminate the distinction between System 1 and System 2 (without always realizing, it must be granted, that that was what they were doing). One such problem is the Wason Selection Task (Wason 1966), which runs as follows. (This problem is gradually heading toward ubiquity in the quasi-popular literature on deductive reasoning; e.g., see (Devlin 2000).) Suppose that you are dealt four cards out of a larger deck, where each card in the deck has a digit from 1 to 9 on one side, and a capital Roman letter on the other. Here is what appears to you when the four cards are dealt out on a table in front of you:

$$\boxed{\text{E}} \qquad \boxed{\text{K}} \qquad \boxed{4} \qquad \boxed{7}$$

Now, your task is to pick just the card or cards you would turn over to try your best at determining whether the following rule is true:

($\mathbf{R}_1$) If a card has a vowel on one side, then it has an even number on the other side.

Less than 5% of the educated adult population can solve this problem (but, predictably, trained mathematicians and logicians are rarely fooled; it would indeed by well nigh *impossible* for them to be fooled by an accurate symbolization of the problem). This result has been repeatedly replicated over the past 15 years, with subjects ranging from 7th grade students to illustrious members of the Academy; see (Bringsjord, Bringsjord & Noel 1998). About 30% of subjects do turn over the E card, but that isn't enough: the 7 card must be turned over as well. The reason why is as follows. The rule in question is a so-called **conditional**, that is, a proposition having an if-then form, which is often symbolized as $\phi \rightarrow \psi$, where the Greek letters here are variables ranging over formulas from some logical system in the family $\mathcal{F}$ introduced below. As the truth-tables routinely taught to young math students make clear (e.g., see Chapter 1 of Bumby, Klutch, Collins & Egbers 1995), a conditional is false if and only if its antecedent, ϕ, is true, while its consequent, ψ, is false; it's true in the remaining three permutations. So, if the E card has an odd number on the other side, (R_1) is overthrown. However, if the 7 card has a vowel on the other side, this too would be a case sufficient to refute (R_1). The other cards are entirely irrelevant, and flipping them serves no purpose whatsoever, and is thus profligate.

This is the abstract, context-independent version of the task. But now let's see what happens when some System 1 context-*de*pendent reasoning is triggered in you, for there is incontrovertible evidence that *if the task in question is concretized*, System 1 reasoning can get the job done.[11] For example, suppose one changes rule (R_1) to this rule:

[10]Though sometimes given a different name, the dichotomy is affirmed by many psychologists of reasoning, as the commentary following (Stanovich & West 2000) reveals.

[11]A nice discussion can be found, e.g., in (Ashcraft 1994).

(R_2) If an envelope is sealed for mailing, it must carry a 20 cent stamp on it.

And now suppose one presents four envelopes to you (keeping in mind that these envelopes, like our cards, have a front and back, only one side of which will be visible if the envelopes are "dealt" out onto a table in front of you), viz.,

sealed envelope	unsealed envelope	env. w/ 20 cent stamp	env. w/ 15 cent stamp

Suppose as well that you are told something analogous to what subjects were told in the abstract version of the task, namely, that they should turn over just those envelopes needed to check whether (R_2) is being followed. Suddenly the results are quite different: Most subjects choose the sealed envelope (to see if it has a 20 cent stamp on the other side), *and* this time they choose the envelope with the 15 cent stamp (to see if it is sealed for mailing!). Such is the power of domain *de*pendent reasoning flowing from System 1.

The challenge to logic-based computational cognitive modeling will be to model *both* types of human reasoning. This challenge will be met if both normatively correct and incorrect responses to the stimuli (puzzles) used in psychology of reasoning are modeled. Prior research that can be plausibly viewed as setting out and tackling aspects of both System 1 and System 2 cognition is hard to find. One exception, to some degree, is Ron Sun's (2001) exploration of implicit versus explicit cognition. His exploration is discussed below (section 4.2.3).

3.1.2 Desideratum #2: Modeling Mental Logic-, Mental Models-, and Mental Metalogic-Based Reasoning

There is another thing the data in psychology of reasoning implies: While sometimes (logically untrained *and* trained) humans reason by explicitly manipulating linguistic entities (e.g., formulas, as when humans construct line-by-line linguistic proofs in proof construction environments like Barwise & Etchemendy (1999)'s Fitch; natural deduction of this linguistic variety is explained below, in section 4.1), they also sometimes reason by imagining and manipulating "mental models," non-linguistic entities capturing possible situations, *and* they sometimes reason in a fashion that involves both mental logic, mental models, and meta-reasoning over the structures posited in these two theories. This meta-reasoning uses rules of inference that at once range over formulas *and* mental models, and are obviously rules that cannot be independently modeled in simulations based either exclusively on mental logic theory, or exclusively on mental models theory.

The first kind of reasoning is explained, explored, and defended by proponents of the theory known as *mental logic* (Rips 1994, Braine 1990, Yang, Braine & O'Brien 1998, Braine 1998*b*, Braine 1998*a*). Mental logic has its roots in Piaget, who held (at least at one point in his intellectual life[12]) that humans naturally acquire the ability to reason at the level of the proof theory of first-order logic (Inhelder & Piaget 1958, Bringsjord et al. 1998). Quintessential cases of this kind of reasoning include giving a proof that from (say) 'If Gooker is a sequaat, then Peeves is a rooloy' and 'Gooker is a sequaat' one can infer 'Peeves is a rooloy' by the rule (*modus ponens*, or — to use the term introduced below — *conditional elimination*):

$$\frac{\text{If } \phi \text{ then } \psi,\ \phi}{\psi}$$

Note that with respect to arguments such as these, it would seem rather odd to say that those who produce them have a mental model of anything. (In connection with quoted description of mental models just below, it seems counter-intuitive to say that reasoners who reason correctly about the Peeves problem have in mind some kind of mental model of the world.) They surely seem to be working just from the surface-level pattern of the linguistic expressions in question. In fact, that is the justification they customarily give when confronted by stimuli of this sort.

The second type of reasoning has been discovered, explained, and defended by Johnson-Laird (1983), who characterizes mental models in their contribution to the present volume like this:

[12]Toward the end of his life, Piaget came to believe that this level of reasoning required the right sort of environmental conditions — a sort that many youth aren't lucky enough to receive. Piaget, also later in his life, held that there is a stage of development *beyond* the capacity to reason at the level of first-order proof theory. This level is one reached by logicians, mathematicians, and the like, who can *meta*-reason over various systems of formal operations, and in fact *create* such systems. This level is also a level at the core of the theory known as mental metalogic, discussed briefly just below.

> The theory of mental models postulates that when individuals understand discourse, they construct models of the possibilities consistent with the discourse. Each mental model represents a possibility. A frequent misunderstanding is that mental models are images. In fact, they are more akin to three-dimensional models of the world of the sort that underlie the phenomena of mental rotation [as introduced, e.g., by (Metzler & Shepard 1982)].

The third sort of reasoning is explained and explored in a third theory known as *mental meta-logic* (Yang & Bringsjord forthcoming, Rinella, Bringsjord & Yang 2001, Yang & Bringsjord 2001, Yang & Bringsjord 2006). According to mental meta-logic, human reasoners, both trained and untrained, often reason in ways that, at once, invoke representations and inference of the sort posited in mental logic *and* mental models, and also meta-inferential rules that manipulate these rules and these models.

Desideratum #2 is that both LCCM and CCM should provide the machinery for mechanizing human reasoning in all three of these modes.

The remaining desiderata each consist in the need to model human reasoning stimulated by a particular puzzle. Each of these puzzles, note, conforms exactly to the structure of experiments set out in section 1. Each of the variables (L, Q, D, etc.) in this structure can be directly instantiated by the specifics in each of the puzzles. Note as well that the puzzles are selected so as to ensure that the modeling of human reasoning in question will entail that the first two desiderata are satisfied.

3.1.3 Desideratum #3: Puzzle 1: The King-Ace Puzzle

The third desideratum is to model human reasoning triggered by the following puzzle, a slight variant[13] of a puzzle introduced by Johnson-Laird (1997).

> Assume that the following is true:
>
> 'If there is a king in the hand, then there is an ace in the hand,' or 'If there is not a king in the hand, then there is an ace in the hand,' — but not both of these if-thens are true.
>
> What can you infer from this assumption? Please provide a careful justification for your answer.

Subjects (logically untrained) almost invariably respond with: "That there is an ace in the hand." This response is entirely incorrect. In point of fact, what one can infer is that there is *not* an ace in the hand. Later, the reader will see exactly why this is the correct answer. The challenge to CCM and LCCM in the case of this second desideratum is to provide a mechanical simulation of both the normatively correct and normatively incorrect responses to this puzzle, and the justification of those responses.

3.1.4 Desideratum #4: Puzzle 2: The Wine Drinker Puzzle

Now let us consider an interesting puzzle devised by Johnson-Laird & Savary (1995) that relates directly to Aristotle's theory of the syllogism:

> Suppose:
>
> - All the Frenchmen in the restaurant are gourmets.
> - Some of the gourmets are wine drinkers.
>
> Does it follow that some of the Frenchmen are wine drinkers? Please provide a careful justification for your answer.

The vast majority of (logically untrained) subjects respond in the affirmative. Yet, the correct answer is "No." Some subjects (some of whom are logically untrained, but the vast majority of which have had significant formal training) respond in the negative, *and* offer a disproof, i.e., a proof that 'Frenchmen are wine drinkers' does *not* follow from the two suppositions. The disproof includes an example (following standard terminology in mathematical logic, a *countermodel*) in which the premises are true but the conclusion false, and the point that such an example establishes a negative answer to the wine drinker query.

The requirement to be met by both CCM and LCCM is that computational simulations of both types of responses be provided.

[13]The variation arises from disambiguating Johnson-Laird's 's or else s'' as 'either s or s', but not both.'

3.1.5 Desideratum #5 Puzzle 3: The Wise Man Puzzle (WMP)

Now to the next puzzle:

> Suppose there are three wise men who are told by their king that at least one of them has a white spot on his forehead; actually, all three have white spots on their foreheads. You are to assume that each wise man can see the others' foreheads but not his own, and thus each knows whether the others have white spots. Suppose you are told that the first wise man says, "I do not know whether I have a white spot," and that the second wise man then says, "I also do not know whether I have a white spot." Now consider the following questions:
>
> **(1)** Does the third wise man now know whether or not he has a white spot?
>
> **(2)** If so, what does he know, that he has one or doesn't have one?
>
> **(3)** And, if so, that is, if the third wise man does know one way or the other, provide a detailed account (showing all work, all notes, etc.; use scrap paper as necessary) of the reasoning that produces his knowledge.

In the case of this puzzle, only the challenge of modeling the (or at least *a*) normatively correct response will be explicitly considered.[14]

3.1.6 Desideratum #6 Puzzle 4: Infinitary DeMorgan

Here is the next puzzle:

> Consider a disjunction as big as the natural numbers,[15] i.e.,
>
> (1) $\phi_1 \vee \phi_2 \vee \phi_3 \vee \ldots \vee \phi_n, \phi_{n+1} \vee \ldots.$
>
> Suppose that (1) is true. Now suppose you also know that
>
> (2) $\phi_{4,599,223,811}$
>
> is false. What can you now conclude must be the case from (1) and (2)? Why?

As in the puzzle that immediately precedes this one, only concern for modeling the normatively correct answer will be present herein, which of course is from (1) and (2) it can be immediately deduced that

$$\phi_1 \vee \phi_2 \vee \ldots \vee \phi_{4,599,223,810} \vee \phi_{4,599,223,812} \vee \phi_{4,599,223,813} \vee \ldots.$$

3.2 Challenge 2 (C2): Unify Cognition via a Comprehensive Theoretical Language

The original dream of the founders of the field of computational cognitive modeling (a dream shared by the founders of modern-day AI) was to provide a core unifying representation scheme, and mechanical processes over this scheme, so as to cover all of human cognition. In the case of Soar and ACT-R, the core representation and process is intended to be essentially the same: chaining in a production system. Other computational cognitive architectures include different core processes. For example, in Clarion, core processing includes a sub-declarative dimension, carried out in artificial neural networks.

The second problem LCCM addresses is that the core processes at the heart of these and other in-progress cognitive architectures certainly don't *seem* well-suited to range across the entire gamut of human cognition. For example, while one can certainly easily enough imagine rapid-fire processing of production rules to cover simple rule-based thinking, it really does boggle the mind to think of formalizing, say, Gödel's declarative cognition in discovering and specifying his famous incompleteness results in the form of production rules. As the attempt to meet the first challenge will reveal later (see section 4.2.1), lots of cognition that seems to call

[14]There is insufficient space to fully explain why this is. In short, WMP is rather more difficult than the simple puzzles that have dominated psychology of reasoning, and this makes it somewhat difficult to see incorrect answers as *systematically* incorrect. The students that succeed can produce, in one form or another (i.e., in terminology introduced later, one **token** or another), the proof that cracks WMP. But the students who fail often fail for *myriad* reasons.

[15]It would be inappropriate in a chapter that doesn't presuppose knowledge of transfinite numbers to be more precise about "bigness" in the present context. It suffices to note that there is a rather obvious bijection holding between the disjuncts and the natural numbers.

for declarative modeling, specifically calls for logical systems much more expressive than those at the level of production rules. As will be seen, logical systems at the level of merely the propositional and predicate calculi (i.e., the logical systems $\mathcal{L}_{PC}$ and $\mathcal{L}_I$, resp.) suffice to formalize production rules and systems.

It can be explained with a bit more precision why it is that non-declarative cognitive architectures cannot transparently offer schemes that cut across the full range of human cognition — from high-level cognition (including meta-cognition, and meta-meta-cognition, ..., as reflected in attribute 5 on the goal list of section 2) to perception of, and actions that affect the, external environment. In the area of high-level cognition, the modeling challenge can be expressed by the puzzles listed above in our description of Challenge 1: How could one write down a production system that represents the declarative statements given in the above puzzles? More precisely (and to return to the abstract structure presented in section 1): We know that each puzzle presents the subject with n declarative statements in English: $s_1, s_2, \ldots, s_n$. Furthermore, it is known that many subjects, when giving normatively correct responses to the puzzles in question, reason over representations of each s_i. Let's denote these representations by $rep(s_i)$. The challenge, then, is to provide a computational simulation based on processing over $rep(s_1), rep(s_2), \ldots, rep(s_n)$. Notice that the challenge is *not* to provide a simulation that simply produces the desired verdict (often, as seen above, either a "Yes" or a "No"). In addition, a justification that explicitly contains each $rep(s_i)$ must be provided. Since, for example, production rules don't include modal operators, and $rep(s_i)$ must include such operators in the case of WMP, it is seen that the challenge is very serious, even within declarative computational cognitive modeling. If one turns to artificial neural network-based computational cognitive architectures, what single, determinate representation in such a network corresponds to $rep(s_2)$, which after all some subjects (as shall be seen later) explicitly write down on scrap paper, and manipulate in accordance with certain explicit rules of inference? And this is only the *start* of the challenge to non-logic-based computational cognitive modeling, for there is the specific process of reasoning that the human subjects in question go through to provide the needed proofs. How could *that* work in a neural network? Again, even if in principle this is possible, LCCM solves these puzzle in a *transparent* way, as is shown below. Given this, if LCCM can also be seen to *clearly* allow models of low-level interaction with the environment (what is later called external perception and action), then it would certainly appear that (C2) is met by LCCM.

To sum up, one can view (C2) as the search for the "unified theoretical language" Ron Sun correctly says is rather hard to come by:

> [I]t is admittedly highly desirable to develop a single, completely unified theoretical language, as a means of expressing fundamental theories of the mind and its various manifestations, components, and phenomena. In place of the classical formalism—symbolic computation, we would certainly like to see a new logico-mathematical formalism that is (1) more analyzable (e.g., in the form of mathematical entities, as opposed to computer programs, which are notoriously difficult to analyze), (2) more inclusive (for example, being able to include both symbols and numeric values, and both serial and parallel processing), and (3) properly constrained (that is, being able to express exactly what needs to be expressed). However, thus far, there is no such a single unified formalism in sight. (Sun 2001, p. 248)

LCCM, as defined herein, would seem to be the formalism Sun is looking for. On Sun's three points: (1) LCCM is in fact *fully* analyzable, and its programs are transparently so, because they are in the declarative mode and are themselves well-defined logico-mathematical objects (more about this when the third challenge, (C3), is shown to be met by LCCM). (2) LCCM, in virtue of logics infused with strength factors (e.g., see section 4.2.3) and probabilities, and of the fact that logic is ideally suited to parallelism, is fully inclusive. (3) The expressibility of LCCM is unparalleled: there is no competitor able to directly and easily express the declarative knowledge in the puzzles in the desiderata composing (C1).

3.3 Challenge 3 (C3): CCM Suffers From a Lack of Mathematical Maturity

In computational cognitive modeling (CCM), a cognitive model is a computational simulation produced by executing code written in some cognitive architecture. Computational cognitive modeling is thus clearly intimately related to computer science, which centrally involves algorithms, programs that are tokens of those algorithms, and the execution of these programs to produce computation. But given this clear connection between computational cognitive modeling and computer science, it's at least somewhat surprising that while the latter is today so rigorous as to be considered by many to be in large part a sub-field of formal logic (Halpern et al. 2001), which is theorem-based, computational cognitive modeling apparently counts nary a theorem among that which it has produced over the course of decades. Part of the root cause of this state of affairs is that the *meaning* of code written in computational cognitive modeling is often somewhat mysterious. Of course, one might retort that since code written in some computational cognitive architecture can be, and indeed sometimes is, written in some established programming language (L_{PL}) having a formal semantics, the meaning of a model can simply be identified with the meaning of the program P written in this L_{PL}. (E.g., L_{PL} could be Common Lisp.) Unfortunately, the meaning of a computational model obviously must be *cognitive* in nature. It is crucial that the meaning of a model relates to the goal of CCM, which is to model human cognition at the symbolic level (as, again, Marr and Pylyshyn would put it), and thereby advance the science of cognition. Put another way, a program written in CCM should have a radically different sort of meaning than, say, a program written in operations research. But if P is written in L_{PL} in the field of CCM, and P' is written in L_{PL} in the field of operations research (or any other field far removed from CCM), both programs will have exactly the same sort of meaning — and it will be a meaning divorced completely from the cognitive or psychological level.

This problem is solved decisively and cleanly by LCCM, as will be seen. Programs written in declarative form have an exact meaning, and that meaning accords with the categories that are constitutive of human cognition, for the simple reason that the declarative level is preserved in the relevant programs. Furthermore, the machinery that yields this precision in turn yields the result that logic-based computational cognitive modeling can be guided by theorems. This result, and the desirability thereof, are discussed later.

4 Logic-Based Computational Cognitive Modeling

4.1 Logical Systems

Logic-based computational cognitive modeling is based on a generalized form of the concept of *logical system* as defined rather narrowly in mathematical logic, where this concept stands at the heart of Lindström's Theorems (for details, see an excellent book covering many additional core concepts presupposed by LCCM: Ebbinghaus, Flum & Thomas 1994).[16] For LCCM, the generalized form of a logical system $\mathcal{L}$ is composed of the following six elements:

1. An object-level alphabet A, partitioned into those symbols that are invariant across the use of $\mathcal{L}$ for particular applications, and those that are included by the human for particular uses. The former are called *fixed* symbols, and the latter *application* symbols.
2. A grammar $\mathcal{G}$ that yields well-formed expressions (usually called *formulas*) L^A from A.
3. An argument theory $\vdash^M_X$ (called a *proof* theory when the reasoning in question is deductive in nature) that specifies correct (relative to the system $\mathcal{L}$) inference from one or more expressions to one or more expressions. The superscript is a placeholder for the *mode* of inference: deductive, abductive, inductive, probabilistic, analogical, etc. The subscript is a placeholder for *particular* inferential mechanisms. For example, in Aristotle's theory of the syllogism, visited at the beginning of the chapter, the first two declarative statements in a valid syllogism deductively imply the third. Where D is used to indicated the deductive mode of inference, and *Syll* the particular deductive scheme introduced by Aristotle, we can write

   $$\{\text{All } AB, \text{All } BC\} \vdash^D_{Syll} \text{All } AC$$

   to indicate that any declarative statement of the form All AC can be deductively inferred in Aristotle's syllogistic system.

   The space of deductive (D) mechanisms include various forms of deduction well beyond what Aristotle long ago devised (e.g., resolution, sequent calculus, Fitch-style natural deduction; they are explained below). Other modes of inference, as mentioned, include: probabilistic inference in Bayesian frameworks, inductive inference, non-monotonic or defeasible inference, and so on.
4. An *argument semantics* that specifies the meaning of inferences allowed by $\vdash^M_x$, which makes possible a mechanical verification of the correctness of arguments.

[16]In a word, these theorems express the fact that logics more expressive than first-order logic necessarily lose certain attributes that first-order logic possesses. It should be pointed out that there are a *number* of different *narrow* accounts of *logical system*; e.g., see (Gabbay 1994).

5. A *formula semantics* that assigns a meaning to members of L^A given announcements about what the application symbols are. The values traditionally include such things as TRUE, FALSE, INDETERMINATE, PROBABLE, numbers in some continuum (e.g., 0 to 1, as in the case of probability theory) and so on.
6. A metatheory that defines meta-mathematical attributes over the previous five components, and includes proofs that the attributes are or are not possessed. Examples of such attributes include soundness (inference in the argument theory from some subset Φ of L^A to ϕ, where $\phi \in L^A$, implies that if all of Φ are true, ϕ must be as well) and completeness (if ϕ is true whenever Φ is, then there is a way to infer ϕ from Φ).

The family $\mathcal{F}$ of logical systems populated by the setting of parameters in the sextet just given is infinite, and includes zero-, first-, and higher-order extensional logics (in Hilbert style, or sequent style, or natural deduction Fitch style, etc.); modal logics (including temporal, epistemic, deontic logics, etc.); propositional dynamic logics; Hoare-Floyd logics for reasoning about imperative programs; inductive logics that subsume probability theory; abductive logics; strength-factor-based and probabilistic logics; non-monotonic logics, and many, many others. Because all of classical mathematics, outside formal logic, is derivable from merely a small proper subset of these systems (with some specific axioms), the machinery of LCCM is enormous. Of necessity, this the scope must be strategically limited in the present chapter.

Accordingly, it is now explained how four logical systems (the first two of which are elementary) are based on particular instantiations of five of the six elements. (We leave aside argument semantics until we discuss logic-based computer programming.) In addition, two additional clusters of logical systems, non-monotonic logical systems and probabilistic logical systems, are briefly discussed after the quartet of logical systems is presented.

The first logical system is $\mathcal{L}_{PC}$, known as the propositional calculus. The second, more powerful logical system is $\mathcal{L}_I$, known as the 'predicate calculus,' or 'first-order logic,' or sometimes just 'FOL.' Every comprehensive introductory cognitive science or AI textbook provides an introduction to these two simple, limited systems, and makes it clear how they are used to engineer intelligent systems (e.g., see Russell & Norvig 2002). In addition, coverage of FOL is often included in surveys of cognitive science (e.g., see Stillings, Weisler, Chase, Feinstein, Garfield & Rissland 1995). Surveys of cognitive psychology, while rarely presenting FOL, often give encapsulated presentations of $\mathcal{L}_{PC}$ (e.g., see Ashcraft 1994). Unfortunately, it is usually the case that when these two logical systems are described, the reader is not told that this pair is but an infinitesimally small speck in the family $\mathcal{F}$.

In both both $\mathcal{L}_{PC}$ and $\mathcal{L}_I$, reasoning is deductive in nature. The third logical system introduced is a particular propositional modal logic, $\mathcal{L}_{KT}$, designed to allow modeling of possibility, necessity, belief, and knowledge. The fourth logical system is based on the simplest infinitary logic: $\mathcal{L}\omega_1\omega$.

4.1.1 The Alphabet and Grammar of $\mathcal{L}_{PC}$

The alphabet for propositional logic is simply an infinite list $p_1, p_2, \ldots, p_n, p_{n+1}, \ldots$ of propositional variables (according to tradition p_1 is p, p_2 is q, and p_3 is r), and the five familiar truth-functional connectives $\neg, \rightarrow, \leftrightarrow, \wedge, \vee$. The connectives can at least provisionally be read, respectively, as 'not,' 'implies' (or 'if then '), 'if and only if,' 'and,' and 'or.' In cognitive science and AI it is often convenient to use propositional variables as mnemonics that help one remember what they are intended to represent. For an example, recall Puzzle 1. Instead of representing 'There is an ace in the hand' as p_i, for some $i \in \mathbf{N} = \{0, 1, 2, \ldots\}$, it would no doubt be useful to represent this proposition as A, and this representation is employed below. Now, the grammar for propositional logic is composed of the following three rules.

1. Every propositional variable p_i is a well-formed formula (wff).
2. If ϕ is a wff, then so is $\neg\phi$.
3. If ϕ and ψ are wffs, then so is $(\phi \star \psi)$, where $\star$ is one of $\wedge, \vee, \rightarrow, \leftrightarrow$. (We allow outermost parentheses to be dropped.)

This implies, for example, that $p \rightarrow (q \wedge r)$ is a wff, while $\rightarrow q$ isn't. To represent the declarative sentence 'If there is an ace in the hand, then there is a king in the hand' we can use $A \rightarrow K$.

4.1.2 An Argument (= Proof) Theory for $\mathcal{L}_{PC}$

A number of proof theories are possible. One such system is an elegant Fitch-style system of natural deduction, F, fully explained in (Barwise & Etchemendy 1999). (Such systems are commonly referred to simply as "natural" systems.) In F, each of the truth-functional connectives has a pair of corresponding inference rules, one for introducing the connective, and one for eliminating the connective. Proofs in F proceed in sequence line by line, each line number incremented by 1. Each line not only includes a line number, but also a formula (the one deduced at this line) and, in the rightmost column, a rule cited in justification for the deduction. The vertical ellipsis

$$\vdots$$

is used to indicate the possible presence of 0 or more lines in the proof.

Here is the rule for eliminating a conjunction:

$$\begin{array}{r|ll}
\vdots & \vdots & \vdots \\
k & \phi \wedge \psi & \\
\vdots & \vdots & \vdots \\
m & \phi & k \ \wedge \ \text{Elim} \\
\vdots & \vdots & \vdots
\end{array}$$

Intuitively, this rule says that if at line k in some derivation you have somehow obtained a conjunction $\phi \wedge \psi$, then at a subsequent line m, one can infer to either of the conjuncts alone. Now here is the rule that allows a conjunction to be introduced; intuitively, it formalizes the fact that if two propositions are independently the case it follows that the conjunction of these two propositions is also true.

$$\begin{array}{r|ll}
\vdots & \vdots & \vdots \\
k & \phi & \\
\vdots & \vdots & \vdots \\
l & \psi & \\
\vdots & \vdots & \vdots \\
m & \phi \wedge \psi & k, l \ \wedge \ \text{Intro} \\
\vdots & \vdots & \vdots
\end{array}$$

A key rule in F is *supposition*, according to which you are allowed to assume any wff at any point in a derivation. The catch is that you must signal your use of supposition by setting it off typographically. Here is the template for supposition:

$$\begin{array}{r|l|ll}
\vdots & & \vdots & \vdots \\
k & & \underline{\phi} & \text{supposition} \\
\vdots & & \vdots & \vdots
\end{array}$$

Often a derivation will be used to establish that from some set Φ of propositional formulas a particular formula ϕ can be derived. In such a case, Φ will be given as suppositions (or, as it is sometimes said, *givens*), and the challenge will be to derive ϕ from these suppositions. To say that ϕ can be derived from a set of formulas Φ in F we follow the notation introduced above and write

$$\Phi \vdash_F^D \phi.$$

When it is clear from context which system the deduction is to take place in, the subscript on $\vdash$ can be omitted. Here is a proof that puts to use the rules presented above and establishes that $\{(p \wedge q) \wedge r\} \vdash_F^D q$:

$$\begin{array}{r|ll}
1 & (p \wedge q) \wedge r & \text{given} \\
\cline{2-2}
2 & (p \wedge q) & 1 \ \wedge \ \text{Elim} \\
3 & q & 2 \ \wedge \ \text{Elim}
\end{array}$$

Now here is a slightly more complicated rule, one for introducing a conditional. It basically says that if you can carry out a sub-derivation in which you suppose ϕ and derive ψ you are entitled to close this sub-derivation and infer to the conditional $\phi \to \psi$.

$$\begin{array}{r|l|ll}
\vdots & \multicolumn{1}{l}{\quad\vdots} & & \\
k & & \phi & \text{supposition} \\ \cline{3-3}
\vdots & & \vdots & \\
m & & \psi & \\
\vdots & \multicolumn{1}{l}{\quad\vdots} & & \\
n & \multicolumn{1}{l}{\phi \to \psi} & & k-m \ \to \ \text{Intro}
\end{array}$$

As stated above, in a Fitch-style system of natural deduction, the rules come in pairs. Here is the rule in F for eliminating conditionals:

$$\begin{array}{r|ll}
k & \phi \to \psi & \\
\vdots & \vdots & \vdots \\
l & \phi & \\
\vdots & \vdots & \vdots \\
m & \psi & k, l \ \to \ \text{Elim}
\end{array}$$

Here is the rule for introducing $\vee$:

$$\begin{array}{r|ll}
\vdots & \vdots & \vdots \\
k & \phi & \\
\vdots & \vdots & \vdots \\
m & \phi \vee \phi & k \ \vee \ \text{Intro} \\
\vdots & \vdots & \vdots
\end{array}$$

And here is the rather more elaborate rule for eliminating a disjunction:

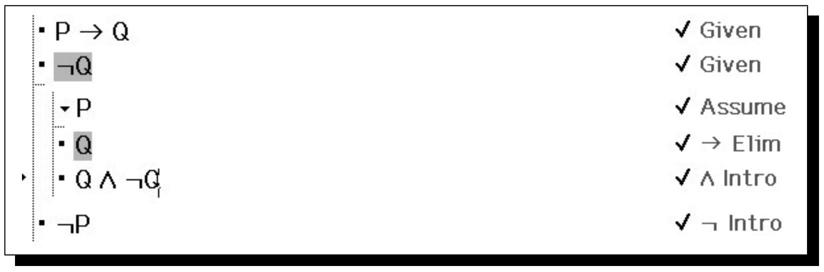
$$\begin{array}{r|l|ll}
\vdots & \multicolumn{1}{l}{\quad\vdots} & & \\
k & \multicolumn{1}{l}{\phi \vee \psi} & & \\
\vdots & \multicolumn{1}{l}{\quad\vdots} & & \\
l & & \phi & \text{supposition} \\ \cline{3-3}
\vdots & & \vdots & \\
m & & \chi & \\
\vdots & \multicolumn{1}{l}{\quad\vdots} & & \\
n & & \psi & \text{supposition} \\ \cline{3-3}
\vdots & & \vdots & \\
o & & \chi & \\
\vdots & \multicolumn{1}{l}{\quad\vdots} & & \\
p & \multicolumn{1}{l}{\chi} & & k, l-m, n-o \ \vee \ \text{Elim}
\end{array}$$

The rule $\vee$ Elim is also known as *constructive dilemma.* The core intuition behind this rule is that if one knows that either ϕ or ψ is true, and if one can show that χ can be proved from ϕ alone, and ψ alone, then χ follows from the disjunction.

Next, here is a very powerful rule corresponding to *proof by contradiction* (sometimes called *indirect proof* or *reductio ad absurdum*). Notice that in F this rule is $\neg$*Intro*.

$$\begin{array}{r|l l}
\vdots & \vdots & \\
k & \quad \begin{array}{|l}\phi \\ \hline \end{array} & \text{supposition} \\
\vdots & \quad \begin{array}{|l}\vdots\end{array} & \\
m & \quad \begin{array}{|l}\psi \wedge \neg\psi\end{array} & \\
\vdots & \vdots & \\
n & \neg\phi & k - m \; \neg \text{ Intro}
\end{array}$$

Sometimes a natural deduction system can be a little obnoxious, because by insisting that inference rules come exclusively in the form of pairs for each truth-functional connective, it leaves out certain rules that are exceedingly useful. Two examples are *modus tollens* and DeMorgan's Laws. The former rule allows one to infer $\neg\phi$ from $\phi \rightarrow \psi$ and $\neg\psi$. This rule can be established through a proof in F, as is shown in Figure 1. This figure shows a screenshot of the completed proof as constructed in the HYPERPROOF proof construction environment, which accompanies the book by the same name authored by Barwise & Etchemendy (1994).[17] The core of this proof is *reductio ad absurdum*, or $\neg$ Intro. DeMorgan's Laws for propositional logic sanction moving from a formula of the form $\neg(\phi \wedge \psi)$ to one of the form $\neg\phi \vee \neg\psi$, and vice versa. The laws also allow an inference from a formula of the form $\neg(\phi \vee \psi)$ to one of the form $\neg\phi \wedge \neg\psi$, and vice versa. When, in constructing a proof in F, one wants to use *modus tollens* or DeMorgan's Laws, or any number of other timesaving rules, one can make the inference in question, using the rule of *tautological consequence* as a justification. This rule, abbreviated as TAUT CON in HYPERPROOF, is designed to allow the human proof constructor a way to declare that a given inference is obvious, and could with more work be fully specified using only the rules of F. HYPERPROOF responds with a check to indicate that an attempted inference is in fact correct. As you can see in Figure 2, HYPERPROOF approves of our use of TAUT CON, which, again, corresponds in this case not just to DeMorgan's Law in the first two occurrences of this rule, but to the useful inference of $\phi \wedge \neg\psi$ from $\neg(\phi \rightarrow \psi)$.



```
▪ P → Q                 ✓ Given
▪ ¬Q                    ✓ Given
   ▾ P                  ✓ Assume
   ▪ Q                  ✓ → Elim
   ▪ Q ∧ ¬Q             ✓ ∧ Intro
▪ ¬P                    ✓ ¬ Intro
```

Figure 1: A Proof of *Modus Tollens* in $\mathcal{F}$, Constructed in HYPERPROOF

This section ends with two more key concepts. A formula provable from the null set is said to be a *theorem*, and where ϕ is such a formula, customary notation is

$$\vdash^D_X \phi$$

to express such a fact, where of course the variable X would be instantiated to the particular deductive calculus in question. Here are two examples that the reader should pause to verify in his or her own mind: $\vdash^D_F (p \wedge q) \rightarrow q$; $\vdash^D_F (p \wedge \neg p) \rightarrow r$. It is said that a set Φ of formulas is *syntactically consistent* if and only if it's not the case that a contradiction $\phi \wedge \neg\phi$ can be derived from Φ.

4.1.3 Formal Semantics for $\mathcal{L}_{PC}$

The precise meaning of the five truth-functional connectives of the propositional calculus is given via truth-tables, which tell us what the value of a statement is, given the truth-values of its components. The simplest truth-table is that for negation, which informs us, unsurprisingly, that if ϕ is T (= TRUE) then $\neg\phi$ is F (= FALSE; see first row below double lines), and if ϕ is F then $\neg\phi$ is T (second row).

[17]For data on the power of HYPERPROOF to help teach logic, see (Rinella et al. 2001, Bringsjord et al. 1998).

ϕ	$\neg\phi$
T	F
F	T

Here are the remaining truth-tables.

ϕ	ψ	$\phi \wedge \psi$
T	T	T
T	F	F
F	T	F
F	F	F

ϕ	ψ	$\phi \vee \psi$
T	T	T
T	F	T
F	T	T
F	F	F

ϕ	ψ	$\phi \rightarrow \psi$
T	T	T
T	F	F
F	T	T
F	F	T

ϕ	ψ	$\phi \leftrightarrow \psi$
T	T	T
T	F	F
F	T	F
F	F	T

Notice that the truth-table for disjunction says that when both disjuncts are true, the entire disjunction is true. This is called *inclusive* disjunction. In *exclusive* disjunction, it's one disjunct or another, but not both. This distinction becomes particularly important if one is attempting to symbolize parts of English (or any other *natural language*). It would not do to represent the sentence

George will either win or lose.

as

$$W \vee L,$$

because under the English meaning there is no way both possibilities can be true, whereas by the meaning of $\vee$ it would be possible that W and L are *both* true. (As we shall soon see, inclusive versus exclusive disjunction is a key distinction in cracking the King-Ace Puzzle.) One could use $\vee_x$ to denote *exclusive disjunction*, which can be defined through the following truth-table.

ϕ	ψ	$\phi \vee_x \psi$
T	T	F
T	F	T
F	T	T
F	F	F

Before concluding this section, it is worth mentioning another issue involving the meaning of English sentences and their corresponding symbolizations in propositional logic: the issue of the "oddity" of *material conditionals* (formulas of the form $\phi \rightarrow \psi$). Consider the following English sentence.

If the moon is made of green cheese, then Dan Quayle will be the next President of the United States.

Is this sentence true? If you were to ask "the man on the street," the answer would likely be "Of course not!" — or perhaps you would hear: "This isn't even a meaningful sentence; you're speaking nonsense." These responses are quite at odds with the undeniable fact that when represented in the propositional calculus, the sentence turns out true. Why? The sentence is naturally represented as

$$G \rightarrow Q.$$

Since G is false, the truth-table for $\rightarrow$ classifies the conditional as true. Results such as these have encouraged some to devise better (but much more complicated) accounts of the "if – then's" seen in natural languages (e.g., see Goble 2001*a*). In fact, a substantial sub-space within the space $\mathcal{F}$ logical systems includes those devoted to just formalizing conditionals (Nute 1984). These accounts will be beyond the purview of this chapter, however: no such search will be embarked upon, so readers must for now be content with the conditional as defined by the customary truth-table for $\rightarrow$ presented above.

Given a truth-value assignment v (i.e., an assignment of T or F to each propositional variable p_i), one can say that v "makes true" or "models" or "satisfies" a given formula ϕ; this is standardly written

$$v \models \phi.$$

A formula such that there is some model that satisfies it is said to be *satisfiable*. A formula that cannot be true on any model (e.g., $p \wedge \neg p$) is said to be *unsatisfiable*. Some formulas are true on all models. For example, the formula $((p \vee q) \wedge \neg q) \rightarrow p$ is in this category. Such formulas are said to be *valid* and are sometimes referred to as *validities*. To indicate that a formula ϕ is valid we write

$$\models \phi.$$

Another important semantic notion is *consequence*. An individual formula ϕ is said to be a consequence of a set Φ of formulas provided that all the truth-value assignments on which all of Φ are true is also one on which ϕ is true; this is customarily written

$$\Phi \models \phi.$$

The final concept in the semantic component of the propositional calculus is the concept of consistency once again: we say that a set Φ of formulas is *semantically consistent* if and only if there is a truth-value assignment on which all of Φ are true. As a check of understanding, the reader may want to satisfy herself that the conjunction of formulas taken from a semantically consistent set must be satisfiable.

4.1.4 Some Metatheoretical Results for $\mathcal{L}_{PC}$

At this point it's easy enough to describe some key metatheory for the propositional calculus. In general, metatheory would deploy logical and mathematical techniques in order to answer such questions as whether or not provability implies consequence, and whether or not the reverse holds. When the first direction holds, a logical system is said to be *sound*, and this fact can be expressed in the notation that has now been introduced as

$$\text{If } \Phi \vdash \phi \text{ then } \Phi \models \phi.$$

Roughly put, a logical system is sound if it's guaranteed that true formulas can only yield (through proofs) true formulas; one cannot pass from the true to the false. When the "other direction" is true of a system it is said to be *complete*; in the notation now available, this is expressed by

$$\text{If } \Phi \models \phi \text{ then } \Phi \vdash \phi.$$

The propositional calculus is both provably sound and complete. One consequence of this is that all theorems in the propositional calculus are valid, and all validities are theorems. This last fact is expressed more formally as:

$$\models \phi \text{ if and only if } \vdash \phi$$

4.1.5 The Alphabet and Grammar of $\mathcal{L}_I$

For $\mathcal{L}_I$, our alphabet will now be augmented to include

$=$	the identity or equality symbol
variables $x, y, \ldots$	like variables in elementary algebra, except they can range of anything, not just numbers
constants $c_1, c_2, \ldots$	you can think of these as proper names for objects
relation symbols $R, G, \ldots$	used to denote properties, e.g., W for *being a wine-drinker*
functors $f_1, f_2, \ldots$	used to refer to functions
quantifiers $\exists, \forall$	the first (existential) quantifier says that "there exists at least one ...," the second (universal) quantifier that "for all ..."
truth-functional connectives $(\neg, \vee, \wedge, \rightarrow, \leftrightarrow)$	now familiar to you, same as in the propositional calculus

Predictable *formation rules* are introduced to allow one to represent propositions like those seen above in Puzzle 2. In the interests of space, the grammar in question is omitted, and the reader is simply shown "in action" the kind of formulas that can be produced by this grammar. You will recall that these three propositions are relevant to Puzzle 2:

1. All the Frenchmen in the restaurant are gourmets.
2. Some of the gourmets are wine drinkers.
3. Some of the Frenchmen in the restaurant are wine drinkers.

With these rules, we can now represent the first of these propositions as

$$\forall x(Fx \rightarrow Gx),$$

which says that for every thing x, if it has property F (is a Frenchman), then it has property G (is a gourmet). The second of the two propositions becomes

$$\exists x(Gx \wedge Wx)$$

and the third is represented as

$$\exists x(Fx \wedge Wx)$$

4.1.6 Argument (= Proof) Theory of $\mathcal{L}_I$

As in propositional logic, sets of formulas (say Φ), given certain *rules of inference*, can be used to prove individual formulas (say ϕ); such a situation is expressed by meta-expressions having exactly the same form as those introduced above, e.g., $\Phi \vdash_X^D \phi$, where of course X will be instantiated to a particular deductive calculus. The rules of inference for FOL in such systems as F include those we saw for the propositional level, and new ones: two corresponding to the existential quantifier $\exists$, and two corresponding to the universal quantifier $\forall$. For example, one of the rules associated with $\forall$ says, intuitively, that if you know that everything has a certain property, then any particular thing a has that property. This rule, known as *universal elimination* or just $\forall$Elim (or, sometimes, *universal introduction*, $\forall$I) allows one to move from some formula $\forall x \phi$ to a formula with $\forall x$ dropped, and the variable x in ϕ replaced with the constant of choice. For example, from 'All Frenchman in the room are wine-drinkers,' that is, again,

$$\forall x(Fx \rightarrow Wx),$$

one can infer by $\forall$ Elim that, where a names some particular object,

$$Fa \rightarrow Wa,$$

and if one happens to know that in fact Fa, one could then infer by familiar propositional reasoning that Ra. The rule $\forall$ Elim in $\mathcal{F}$, when set out more carefully, is

$$\begin{array}{r|ll} k & \forall x \phi & \\ \vdots & \vdots & \vdots \\ l & \phi(\frac{a}{x}) & k\ \forall \text{ Elim} \end{array}$$

where $\phi(\frac{a}{x})$ denotes the result of replacing occurrences of x in ϕ with a.

4.1.7 Semantics of $\mathcal{L}_I$

FOL includes a semantic side which systematically provides meaning (i.e., truth or falsity) for formulas. Unfortunately, the formal semantics of FOL gets quite a bit more tricky than the truth table-based scheme sufficient for the propositional level. The central concept is that in FOL formulas are said to be true (or false) on *interpretations*; that some formula ϕ is true on an interpretation is often written as $\mathcal{I} \models \phi$. (This is often read, "$\mathcal{I}$ satisfies, or models, ϕ.") For example, the formula $\forall x \exists y Gyx$ might mean, on the standard interpretation for arithmetic, that for every natural number n, there is a natural number m such that $m > n$. In this case, the *domain* is the set of natural numbers, that is, $\mathbf{N}$; and G symbolizes 'greater than.' Much more could of course be said about the formal semantics (or *model theory*) for FOL — but this is an advanced topic beyond the scope of the present, brief treatment. For a fuller but still-succinct discussion using the traditional notation of model theory see (Ebbinghaus et al. 1994). The scope of the present discussion does allow the reader to appreciate that FOL, like the propositional calculus, is both sound and complete; proofs can be found in (Ebbinghaus et al. 1994). This fact entails a proposition that will prove useful momentarily: that if ϕ isn't a consequence of Φ, then ϕ cannot be proved from Φ. In the notation introduced earlier, this is expressed as:

$$\Phi \not\models \phi \text{ then } \Phi \nvdash \phi$$

4.1.8 Alphabet and Grammar for $\mathcal{L}_{KT}$

This logical system $\mathcal{L}_{KT}$ adds the modal operators $\Box$ and $\Diamond$ to the grammatical machinery of $\mathcal{L}_{PC}$, with subscripts on these operators to refer to agents. Because the concern here is with what agents believe and know (i.e., with what is called epistemic or doxastic logic; a nice overview is provided in (Goble 2001*b*)), the focus is on the box, and therefore $\Box_\alpha$ is rewritten as $\mathbf{K}_\alpha$. So, to represent that 'Wise man A knows he doesn't have a white spot on his forehead.' one can write $\mathbf{K}_A(\neg\text{White(A)}$. Here's the grammar for $\mathcal{L}_{KT}$.

1. All ordinary wffs are wffs.
2. If ϕ is a closed wff, and α is a constant, then $\Box_\alpha\phi$ is a wff. Since the concern in WMP is with **doxastic** matters, that is, matters involving believing and knowing, one says that $\mathbf{B}_\alpha\phi$ is a wff, or, if one is concerned with 'knows' rather than 'believes,' that $\mathbf{K}_\alpha\phi$ is a wff.
3. If ϕ and ψ are wffs, then so are any strings which can be constructed from ϕ and ψ by the usual propositional connectives (e.g., $\rightarrow, \wedge, \ldots$).

4.1.9 Semantics for $\mathcal{L}_{KT}$

The formal semantics for $\mathcal{L}_{KT}$ can be achieved via three steps. The cornerstone of these steps is the concept of a *possible world*. Intuitively, the idea, which goes back to (Hintikka 1962), and can arguably be traced back as far as Aristotle's treatment of the logic of knowledge and belief in his *De Sophisiticis Elenchis* and in the *Prior* and *Posterior Analytics* (McKeon 1941), is that some agent α knows some declarative statement (= some proposition) ϕ provided that, in all possible worlds compatible with what α knows, it is the case that ϕ. The compatibility between worlds can be regimented by way of an *accessibility relation* between them. Here are the three steps:

1. Associate with each interpretation (which now includes a set, A, of agents) a **possible world**.
2. Establish a relation — the **accessibility relation** — $k \subseteq A \times W \times W$ where W denotes the set of all possible worlds.
3. Now it is said that $\mathbf{K}_\alpha\phi$ is true in some possible world w_i iff ϕ is true in every world w_j such that $< \alpha, w_i, w_j > \in k$. We write this as $\models_{w_i} \mathbf{K}_\alpha\phi$.

For a full, modern treatment of epistemic logic in connection with computationally modeling the mind (from the standpoint of AI), see (Fagin, Halpern, Moses & Vardi 2004).

4.1.10 The Simplest Infinitary Logical System: $\mathcal{L}\omega_1\omega$

Because $\mathcal{L}_I$ is so limited (most interesting mathematical statements cannot be expressed in FOL; e.g., the concept of finitude, central to mathematics, provably cannot be expressed in FOL), logicians have studied infinitary logics like $\mathcal{L}_{\omega_1\omega}$, the definition of which is now provided.

The basic idea behind $\mathcal{L}_{\omega_1\omega}$ is straightforward. This logical system allows for infinite disjunctions and conjunctions,[18] where these disjunctions and conjunctions are no longer than the size of the set of natural numbers (let's use ω to denote the size of the set of natural numbers).[19] This fundamental idea is effortlessly regimented: First one simply adds to the customary alphabet for first-order logic the symbols $\bigvee$ and $\bigwedge$. To the ordinary formation rules for building grammatically correct first-order formulas, one then adds

- If Φ is a set of well-formed formulas $\{\phi_1, \phi_2, \ldots\}$ no larger than ω, then $\bigvee \Phi$ ($\bigwedge \Phi$) is also a well-formed formula, viz., the disjunction (conjunction) of the formulas in Φ.

The conditions under which an infinite formula is true is fixed by extending the notion of truth in ordinary first-order logic:

[18]Of course, even finitary logics have underlying alphabets that are infinite in size (the propositional calculus comes with an infinite supply of propositional variables). $\mathcal{L}_{\omega_1\omega}$, however, allows for formulas of infinite length — and hence allows for infinitely long derivations. More about such derivations in a moment.

[19]This chapter, as stated at the outset, is aimed at an audience assumed to have familiarity with but elementary logic. So this isn't the place to baptize readers into the world of cardinal numbers. Hence the size implications of the subscripts in $\mathcal{L}_{\omega_1\omega}$, and other related niceties, such as the precise meaning of ω, are left to the side. For a comprehensive array of the possibilities arising from varying the subscripts, see (Dickmann 1975).

- A possibly infinite disjunction, $\bigvee \Phi$, is true on an interpretation $\mathcal{I}$ (written $\mathcal{I} \models \bigvee \Phi$) if and only if there is a formula ϕ in Φ which is true on $\mathcal{I}$.
- A possibly infinite conjunction, $\bigwedge \Phi$, is true on an interpretation $\mathcal{I}$ (written $\mathcal{I} \models \bigwedge \Phi$) if and only if every formula ϕ in Φ is true on $\mathcal{I}$.

Proofs (= derivations) in $\mathcal{L}_{\omega_1\omega}$ can, as the relevant literature states, be "infinitely long" (Ebbinghaus, Flum & Thomas 1984). This is because in addition to classical cornerstones like *modus ponens* covered above,

$$\text{from}\, \phi \to \psi \,\text{and}\, \phi \,\text{infer to}\, \psi,$$

$\mathcal{L}_{\omega_1\omega}$ allows rules of inference like

$$\text{from}\, \phi \to \psi \,\text{for all}\, \psi \in \Phi, \text{ infer to}\, \phi \to \bigwedge \Psi.$$

This rule says that if in a derivation you have an infinite list of if-thens (i.e., formulas of the form $\phi \to \psi$) where each consequent (ψ) in each if-then is an element of some infinite set Φ, then you can infer to an if-then whose consequent is the infinite conjunction obtained by conjoining all the elements of Φ. It may be worth pausing a bit to create a picture of the sort of derivation which is here permitted: Suppose that Γ is an infinite set of the same size as $\mathbf{N}$, the natural numbers. So Γ is $\{\gamma_1, \gamma_2, \ldots, \gamma_n, \gamma_{n+1}, \gamma_{n+2}, \ldots\}$. Then here is one possible picture of an infinite derivation:

$$\begin{array}{|l|}\hline \phi \to \gamma_1 \\ \phi \to \gamma_2 \\ \phi \to \gamma_3 \\ \vdots \\ \phi \to \gamma_n \\ \phi \to \gamma_{n+1} \\ \vdots \\ \hline \phi \to \gamma_1 \land \gamma_2 \land \ldots \land \gamma_n \land \gamma_{n+1} \land \gamma_{n+2} \ldots \\ \hline \end{array}$$

It should be clear from this that derivations in $\mathcal{L}_{\omega_1\omega}$ can indeed be infinitely long.

4.1.11 Nonmonotonic Logical Systems

Deductive reasoning is monotonic. That is to say, if ϕ can be deduced from some knowledge base Φ of formulas (written, recall, $\Phi \vdash^d_x \phi$), then for any formula $\psi \notin \Phi$, it remains true that $\Phi \cup \{\psi\} \vdash^d_x \phi$. In other words, when the reasoning in question is deductive in nature, new knowledge never invalidates prior reasoning. More formally, where Φ is some set of formulas, the closure of this set under standard deduction (i.e., the set of all formulas that can be deduced from Φ), denoted by $\Phi^{\vdash}$, is guaranteed to be a subset of $(\Phi \cup \Psi)^{\vdash}$, for all sets of formulas Ψ. This is not how real life works, at least when it comes to humans; this is easy to see. Suppose that at present, Jones knows that his house is still standing as he sits in it, typing. If, later in the day, while away from his home and working at his office, he learns that a vicious tornado passed over the town in which his house is located, he has new information that probably leads him to at least suspend judgment as to whether or not his house still stands. Or to take the much-used example from AI, if Smith knows that Tweety is a bird, he will probably deduce that Tweety can fly, on the strength of a general principle saying that birds can fly. But if he learns that Tweety is a penguin, the situation must be revised: that Tweety can fly should now not be in Smith's knowledge base. Nonmonotonic reasoning is the form of reasoning designed to model, formally, this kind of *defeasible* inference.

There are many different logic-based approaches that have been designed to model defeasible reasoning, and each one is associated with a group of logical systems, as such systems have been defined above. Such systems include: default logic, circumscription, argument-based defeasible reasoning, and so on. (The *locus classicus* of a survey can be found in (Genesereth & Nilsson 1987). An excellent survey is also provided in the Stanford Encyclopedia of Philosophy.[20]) In the limited space available in the present chapter, the wisest

[20]At

http://plato.stanford.edu/entries/logic-ai course is to briefly explain one of these approaches. Argument-based defeasible reasoning is selected, because it seems to accord best with what humans actually do as they adjust their knowledge through time.[21]

Let us return to the tornado example. What is the argument that Jones might give to support his belief that his house still stands, while he sits within it, typing? There are many possibilities, one respectable one is what can be labeled 'Argument 1,' where the indirect indexical refers of course to Jones:

(1) I perceive that my house is still standing.
(2) If I perceive ϕ, ϕ holds.
$\therefore$ (3) My house is still standing.

The second premise is a principle that seems a bit risky, perhaps. No doubt there should be some caveats included within it: that when the perception in question occurs, Jones is not under the influence of drugs, not insane, and so on. But to ease exposition, let's leave aside such clauses. So, on the strength of this argument, we assume that Jones' knowledge base includes (3), at time t_1.

Later on, as we have said, he finds himself working in his office, away from hom. A tornado passes over his building. Jones quickly queries his web browser once the roar and rumble dies down, and learns from the National Weather Service this very same tornado has touched down somewhere in the town T in which Jones' house is located. At this point (t_2, assume), if Jones were pressed to articulate his current position on (3), and his reasoning for that position, and he had sufficient time and patience to comply, he might offer something like this (Argument 2):

(4) A tornado has just (i.e., at some time between t_1 and t_2) touched down in T, and destroyed some houses there.
(5) My house is located in T.
(6) I have no evidence that my house was *not* struck to smithereens by a tornado that recently passed through the town in which my house is located.
(7) If a tornado has just destroyed some houses in (arbitrary) town T', and house h is located in T, and one has no evidence that h is not among the houses destroyed by the tornado, then one ought not to believe that h wasn't destroyed.
$\therefore$ (8) I ought not to believe that my house is still standing. (I.e., I ought not to believe (3).)

Assuming that Jones meet all of his "epistemic obligations" (in other words, assuming that he's rational), he will not believe (3) at t_2. Therefore, at this time, (3) will not be in his knowledge base. (If a cognitive system s doesn't believe ϕ, it follows immediately that s doesn't know ϕ.) The nonmonotonicity should be clear.

The challenge is to devise formalisms and mechanisms that model this kind of mental activity through time. The argument-based approach to nonmonotonic reasoning does this. While the details of the approach must be left to outside reading (see Pollock 1992, Pollock 2001), it should be easy enough to see that the main point is to allow one argument to shoot down another (and one argument to shoot down an argument that shoots down an argument, which revives the original, etc.), and to keep a running tab on which propositions should be believed at any particular time. Argument 2 above rather obviously shoots down Argument 1; this is the situation at t_2. Should Jones then learn that only two houses in town T were leveled, and that they are both located on a street other than his own, Argument 2 would be defeated by a third argument, because this third argument would overthrow (6). With Argument 2 defeated, (3) would be reinstated, and back in Jones' knowledge base. Clearly, this ebb and flow in argument-versus-argument activity is far more than just straight deductive reasoning.

4.1.12 Probabilistic Logical Systems

While, as we have seen, declarative/logic-based computational cognitive modeling was being pursued in earnest more than 2,300 years ago, probability theory is only about 200 years old; it emerged from technical

[21] From a purely formal perspective, the simplest way to achieve non-monotonicity is to use the so-called *closed world assumption*, according to which, given a set Φ of initially believed declarative statements, what an agent believes after applying the closed world assumption (CWA) to the set is not only what can be deduced from Φ, but also the negation of every formula that *cannot* be deduced. It is easy to verify that it doesn't always hold that CWA(Φ) $\subset$ CWA($\Phi \cup \Psi$), for all sets Ψ. I.e., monotonicity doesn't hold.

philosophy and logic (Glymour 1992, Skyrms 1999). Kolmogorov's axioms, viz.,

1. All probabilities fall between 0 and 1. I.e., $\forall p(0 \leq P(p) \leq 1)$.
2. Valid (in the traditional logic-based sense explained earlier in the present chapter) propositions have a probability of 1; unsatisfiable (in the traditional logic-based sense explained earlier) propositions have a probability of 0.
3. $P(p \vee q) = P(p) + P(q) - P(p \wedge q)$

are simple formulas from a simple logical system, but modern probability theory can be derived from them in straightforward fashion.

The reader may wonder where probabilistic *inference* enters the picture, since traditional deduction is not used for inference in probability theory. Probabilistic inference consists in computing, from observed evidence expressed in terms of probability theory, posterior probabilities of propositions of interest. In the relevant class of logical systems, the symbol to be used is $\vdash^P rob_X$, where X would be the particular way of computing over prior distributions to support relevant posterior formulas. Recently, the assignment to X has received much interest, because some strikingly efficient ways of representing and computing over distributions have arrived due to the use of graph-theoretic structures, but the expressiveness of probability theory is ultimately bounded by the logical system with which it's associated, and the two systems in question (the propositional calculus and first-order logic, both of course introduced above as $\mathcal{L}_{PC}$ and $\mathcal{L}_I$, resp.) are rather inexpressive, from the mathematical point of view afforded by $\mathcal{F}$. In fact, extending probability theory to the first-order case is a very recent achievement, and things are not settled (Russell & Norvig 2002).

Because another chapter in the present handbook covers probabilistic computational cognitive modeling (see "Bayesian Models of Cognition," by Griffiths et al. in this volume), no more is said about such logical systems here. The interested reader is also directed to (Skyrms 1999, Russell & Norvig 2002, Bringsjord in-press) for additional coverage of probabilisitc formalisms and modeling.

4.2 Sample Declarative Modeling in Conformity to LCCM

Though the purpose of this chapter is to present logic-based computational cognitive modeling itself, and to show it at work directly, it is appropriate to present a few examples of declarative computational cognitive modeling, and to see how this kind of modeling is formalized by LCCM. Three such examples will now be provided.

4.2.1 Production Rule-Based Modeling

Much computational cognitive modeling is based on production-rules. For example, Soar, ACT-R, and EPAM are based on such rules, and, accordingly, are often called *production systems*. But what is a production rule? In a seminal paper dealing with the relationship between logic and production rule-based modeling, Eisenstadt & Simon (1997) tell us that

> A production [rule] can be represented in the form **C** $\rightarrow$ **A**, where the **C** represents a set of *conditions*, which are knowledge elements, either stored in memory or derived from current stimuli; and the **A** represents a set of *actions*, that either alter internal symbol structures or initiate more responses, or both. (Eisenstadt & Simon 1997, p. 368)

Given the technical content shared with the reader earlier, what should come to mind when seeing '**C** $\rightarrow$ **A**' is a conditional in $\mathcal{L}_I$ (or even, in some simple cases, in $\mathcal{L}_{PC}$), and in point of fact this logical system does provide a precise formalization of activity in collections of interconnected production rules. In any case where **A** is performed, the declarative content **C** is satisfied, and there exists a mechanical sequence of deductive inference (a proof) that produces **C** as a conclusion, and indeed a proof that produces a declarative representation of **A** as a conclusion.[22]

[22]More generally, the production rules used to specify the operation of a Turing machine (in some formalizations of these machines, such rules are used), and executions of these rules, can be entirely algorithmically replaced with deduction over these rules expressed exclusively in $\mathcal{L}_I$. A readable proof of this is found in (Boolos & Jeffrey 1989). Going from the abstract and mathematical to the concrete, any production rule-based activity in an *implemented* system (such as Soar), can be directly matched by the corresponding execution of a program in the general space $P_{\mathcal{L}_{PC}}$ of such program (see the coverage below of logic-based computer programs).

Let us consider an example to make this clearer; the example parallels one given — for different purposes — by Eisenstadt & Simon (1997). Three consequences result from a dog chasing a cat, where this event consists in the instantiation of **Chasing(x,y)**, **Dog(x)**, and **Cat(y)**. The consequences are certain actions, denoted, respectively, by **Consequence1(x)**, **Consequence2(y)**, and a general consequence **Consequence3**; the third consequence, Eisenstadt & Simon (1997) tell us, consists in the cat knocking over a bowl. The idea is that if instantiations of the three conditions appear in memory (i.e., that if **Chasing(a,b)**, **Dog(a)**, **Cat(b)** are in memory), the production is executed and the consequences ensue.

Where

$$Happens(Consequence1(x)) \wedge Happens(Consequence2(y)) \wedge Happens(Consequence3))$$

expresses in $\mathcal{L}_I$ that the three consequences do in fact transpire, this proposition combined with the following four formulae allows any standard automated prover (ATP) to instantly prove exactly what is desired.

1. $\forall x \forall y((Chasing(x,y) \wedge Dog(x) \wedge Cat(y)) \rightarrow$

   $$(Happens(Consequence1(x) \wedge Happens(Consequence2(y)) \wedge Happens(Consequence3))))$$

2. $Chasing(a,b)$
3. $Dog(a)$
4. $Cat(b)$

As we shall see, ATPs stand to logic-based computer programming as, say, built-in functions like addition stand to an established programming language like Common Lisp (Steele 1984) (which happens to be long associated with computational cognitive modeling and AI).[23] In Lisp, `(+ 4 5)`, when executed, returns `9`. Likewise, a standard ATP, upon given the five formulas above, and a request to prove whether the second consequence obtains, brings the correct answer back immedidately. For example, here is a proof instantly returned by the well-known and long-established ATP known as Otter (Wos, Overbeek, e. Lusk & Boyle 1992), once the five formulae are asserted, and the query is issued.

```
---------------- PROOF ----------------
2 [] -Chasing(x,y)| -Dog(x)| -Cat(y)|Happens(consequence2(y)).
4 [] -Happens(consequence2(b)).
5 [] Dog(a).
6 [] Cat(b).
7 [] Chasing(a,b).
9 [hyper,7,2,5,6] Happens(consequence2(b)).
10 [binary,9.1,4.1] $F.
------------ end of proof -------------
```

This particular proof uses a mode of deductive inference called *resolution*, a mode that, using the notation introduced above, can be labeled $\vdash^d_{res}$, The core rule of inference in resolution is simply that from $\phi \vee \psi$ and $\neg\phi$ it can be inferred that ψ. In this inference, it can be accurately said that ϕ and $\neg\phi$ "cancel each other out," leaving ϕ. The careful reader can see this "cancellation" at work in the inference at the line containing `hyper`, and in the inference at the line containing `binary`.

It's important to know that while all that is represented in a production system, and all processes over those representations, corresponds, formally speaking, to representation and reasoning in simple logical systems, the converse does not hold. The reason for this is simply that some declarative information exceeds the particular structure for expressing declarative information available in the production paradigm. Whereas every production rule maps directly to a particular formula in some logical system, and every firing of production rules maps directly to a machine-generated proof, many formulas in many logical systems exceed the expressive power of production rules and systems. This is without question true for an entire sub-class infinitary logical systems are in this category. Even the smallest infinitary logical system ($\mathcal{L}_{\omega_1\omega}$, visited above) allows for declarative statements that exceed production rules. Even more mundane declarative statements, while easily expressed by particular formulas in particular logical systems, at the very least pose

[23]For cognoscenti to see that the analogy between these built-in functions and ATPs holds firmly, it must be stipulated that calls like $\Phi \vdash^M \phi$? include an interval of time n beyond which the machine's deliberation will not be allowed to pass.

an extreme challenge to production systems. For example, while ϕ = 'Everyone loves anyone who loves at least three distinct people' is trivially mapped to a formula in $\mathcal{L}_I$,[24] it is impossible to devise one production rule to correspond directly to this declarative statement. Things get even harder when expressivity must increase. For example, operators can range over ϕ as when, in quantified epistemic logic ($\mathcal{L}_{QKT}$), we say such things as that Jones believes that Smith believes ϕ.

4.2.2 Rips' PSYCOP and Johnson-Laird-Written Models

Rips (1994) describes a system (PSYCOP) designed to model normatively incorrect human reasoning at the level of the propositional calculus (i.e., at the level of $\mathcal{L}_{PC}$), and to some degree (if for no other reason than that PSYCOP includes normatively correct deductive rules of inference) normatively correct reasoning at this level as well. This means that PSYCOP is couched in terms of the logical system $\mathcal{L}_{PC}$, discussed above (from which it follows that whatever declarative statement(s) $\mathcal{L}_{PC}$ cannot express, PSYCOP cannot express).

PSYCOP reflects the driving dogma of the aforementioned theory of human reasoning known as *mental logic* — the dogma being that (logically untrained) human reasoners reason by following rules of inference similar to those used in the argument theory of $\mathcal{L}_{PC}$. For example, while the inference rule mentioned in a moment is absent from the system, *modus ponens* is in it, as are a number of other rules. (PSYCOP and its underpinnings are critiqued by Johnson-Laird, in his entry in the present volume.) It's important to note that PSYCOP is not an architecture designed to computationally model all of human cognition. In fact, PSYCOP can't be used to model the human reasoning triggered by the puzzle-based desiderata listed above. This is so because PSYCOP is insufficiently expressive (e.g., it has no modal operators like those in $\mathcal{L}_{KT}$, and they are needed for WMP, as we shall soon see; nor does PSYCOP even have the basic quantifiers or first-order models of $\mathcal{L}_I$), and doesn't allow trivial normatively correct inferences that good deductive reasoners make all the time. For example, in PSYCOP, you can't infer from the falsity of (where ϕ and ψ are well-formed formulas of $\mathcal{L}_{PC}$) "If ϕ, then ψ" to the falsity of ψ, but that is an inference that even logically untrained reasoners do sometimes make. For example, they sometimes say, when faced with such declarative sentences as

It's false that: If the cat is not on the mat, then Jones is away.

that if the if-then is false, the "if part" (= the antecedent) must be true while the "then part" (= the consequent) isn't — which immediately implies here that Jones isn't away.

Interestingly enough, Rips explicitly considers the possibility of a "deduction-based" cognitive architecture (in Chapter 8, "The Role of Deduction in Thought," in Rips 1994). This possibility corresponds to a proper subset of what is realized by LCCM. Rips has in mind only a particular simple extensional logic as the core of this possibility (viz., $\mathcal{L}_{PC}$), whereas LCCM , as you now know, is based on the literally infinitely broader concept of the family $\mathcal{F}$ of logical systems; on not just deduction, but other forms of reasoning as well (e.g., induction, abduction, non-monotonic reasoning, etc.); and on a dedicated programming paradigm tailor-made for implementing such systems.

Now, what about Johnson-Laird's (1983) mental models theory, and specifically some computer programs that implement it? Does this work also fall under LCCM? Since Johnson-Laird's work is declarative in nature, it does indeed fall under logic-based computational cognitive modeling.

Mental models theory has been, at least in part, implemented in the form of various computer programs (see the contribution from Johnson-Laird and Yang in the present volume), but none of these programs constitute across-the-board computational cognitive models of the human cognizer. Instead, the reasons offered as to why such programs have been written include the standard (but very compelling) ones — such as that the computer implementation of a psychological theory can reveal ambiguities and inconsistencies in that that theory. However, the programs in question obviously fall under LCCM. After all, these are programs that produce models in the logic-based sense, and then reason over these models in accordance with rules naturally expressed in logic. At the level of $\mathcal{L}_{PC}$, mental models in mental models theory correspond to rows in truth tables that yield TRUE for the formula, and where only the true literals in those rows are included (a literal is either some p_i, or $\neg p_i$). For example, the mental models corresponding to "The Yankees

[24]In $\mathcal{L}_I$ it's simply $\forall x \forall y((\exists z_1 \exists z_2 \exists z_3(z_1 \neq z_2 \wedge z_2 \neq \wedge z_1 \neq z_3 \wedge Lyz_1 \wedge Lyz_2 \wedge Lyz_3)) \rightarrow Lxy)$.

will win and the Red Sox will lose," assuming a symbolization of $Y \wedge R$ for this English sentence, yields one mental model:

$$Y \quad R$$

In the case of disjunction, such as "Either the Yankees will win the World Series, or the Red Sox will," the models would be three in number, viz.,

$$\begin{array}{ll} Y & \\ R & \\ Y & R \end{array}$$

Since on mental models theory a conclusion is necessary if it holds in all the models of the premises, some deductively valid inferences, such as that "The Yankees will win" follows from "The Yankees will win and the Red Sox will lose," should be made by logically untrained subjects. This is a normatively correct inference. (However, standard normatively correct justifications are not available on Johnson-Laird's theory. This is so because standard justifications are proofs of the sort seen in formal logic and mathematics.) What about normatively *in*correct reasoning? Clearly, such reasoning will frequently occur, according to the theory. Think back to the formal structure of experiments as set out at the beginning of this chapter (section 4.3). Suppose the human reasoner assimilates premises in the list L yielding the mental models in a set S. The reasoner will declare a purported conclusion D to follow if $D \in S$, and such membership can obviously hold independent of formally valid inference.

4.2.3 Rule-Based, Similarity-Based, and Commonsense Reasoning in LCCM

The Clarion cognitive architecture models human declarative reasoning in some very interesting non-logic-based ways. For example, Clarion models two apparently distinct forms of (simple) reasoning detected in the logically untrained. While the distinction between these two forms can't be modeled through naive use of first-order logic ($= \mathcal{L}_I$), since both forms of reasoning are declarative in nature, it is effortless to model the distinction using the full arsenal of LCCM; that is, using logical systems in the space $\mathcal{F}$ that are more expressive than $\mathcal{L}_I$. This is now shown.

The two forms of reasoning in question are what Sun and Zhang (2006) call 'rule-based reasoning' (RBR) and 'similarity-based reasoning' (SBR)." In order to look a bit more closely at the situation, one can turn to the specific stimuli on which Sun and Zhang focus, which are taken from (Sloman 1998). Stimuli consisted of pairs of arguments. Some pairs are said to be in the form of "premise specificity," as for instance in this pair:

All flowers are susceptible to thrips $\Rightarrow$ All roses are susceptible to thrips.
All plants are susceptible to thrips $\Rightarrow$ All roses are susceptible to thrips.

Other pairs are in the form of what is called "inclusion similarity." Examples include:

All plants contain bryophytes. $\Rightarrow$ All flowers contain bryophytes.
All plants contain bryophytes. $\Rightarrow$ All mosses contain bryophytes.

Subjects were directed to pick the stronger argument from each pair.[25] In response, the vast majority of subjects, for both types of pairs, selected as stronger the "more similar argument," as Sun and Zhang put it. By this they mean that the vast majority of subjects chose, from each pair, the argument whose subjects are intuitively regarded to be more similar. For example, the assumption is that roses are more similar to flowers than they are to plants.[26]

[25]Presumably one should assume that subjects were told to pick *what they perceived to be* the stronger argument from each pair. Since all the arguments in question are fallacious, it is not possible, from the mathematical point of view taken by LCCM, that any argument be stronger than the other one in a pair. All the arguments are *enthymemes*; all enthymemes, by definition, are formally invalid. The other horn of the dilemma is that if the missing premise in each argument had been supplied (e.g., 'All roses are flowers'), then each would become formally deductively valid, and it would once again be inappropriate to deem one argument stronger than another. Though these issues are somewhat worrisome, due to space constraints that are left aside.

[26]For lack of space, formal reasons for doubting that such similarity is in any way actual must be left aside. Presumably, once again, the issue is *perceived* similarity, or some such thing. And this perceived similarity is presumably supposed to be generally true of the population in question.

It should be apparent that if only RBR (e.g., based on logics) was used, then similarity should not have made a difference, because the conclusion category was contained in the premise category, and thus both arguments in each pair should have been equally, perfectly strong. Therefore, the data suggested that SBR (as distinct from RBR or logics capturing category inclusion relations) was involved to a significant extent. (Sun & Zhang 2006)

Of course, by 'RBR' Sun and Zhang here mean "barebone" RBR, which involves only category inclusion relations.

Now, Sun and Zhang proceed to show that Clarion can be used to model the distinction between RBR and SBR. While that modeling is impressive, the point relevant to the present chapter is that the RBR-is-distinct-from-SBR phenomenon, since it is declarative in nature, is also easily enough modeled in the LCCM approach, since this approach is based on $\mathcal{F}$, the infinite family of logical systems, not on a particular logic. Sun and Zhang, in the parenthetical in the quote immediately above, indicate that RBR is "based on logics." The use of the plural here is wise. For it would be exceedingly peculiar for any proponent of LCCM to maintain that human reasoning, let alone human cognition, can be modeled by a *particular* logic. This would be even more peculiar than maintaining that in modeling various phenomena, mathematicians and physicists can restrict themselves to only one specific branch of mathematics for purposes of modeling. (The tensor calculus is a thing of beauty when it comes to relativity, but of what good is it in, say, doing axiomatic set theory?) Following logical systems introduced by Chisholm (1966, 1977, 1987), and Pollock (Pollock 1974), one can assign strength factors (note: not probabilities; strength factors) to non-deductive inferential links, as in fact has been done in the LCCM-based Slate interactive reasoning system (Bringsjord, Arkoudas, Clark, Shilliday, Taylor, Schimanski & Yang 2007).[27] In Slate, these factors include, in descending strength, **certain** (4), **evident** (3), **beyond reasonable doubt** (2), **probable** (1), and **counter-balanced** (0), and then the negative counterparts to the first four of these (yielding, in the numerical shorthand, -1, -2, -3, and -4. These strength factors can be effortlessly associated with knowledge about the similarity of the subjects involved in such arguments as those studied by Sloman, Sun, and Zhang. Given, then, a pair such as

All flowers are susceptible to thrips $\Rightarrow_2$ All roses are susceptible to thrips.

All plants are susceptible to thrips $\Rightarrow_1$ All roses are susceptible to thrips.

and a straightforward selection algorithm for strength of argument that works by simply selecting the argument whose inferential link is associated with a higher number, LCCM has no trouble in the least modeling the phenomenon in question (i.e., the distinction between barebone RBR and SBR is formally captured). Moreover, since the cognitive plausibility of such strength factors rather obviously inheres in the fact that the average subject assigns a higher strength factor to the hidden premise (so that, e.g., 'All roses are flowers' is epistemically stronger than 'All roses are plants'), LCCM would allow us to model human reasoning that leads subjects to prefer even the more similar pairs in the *non-enthymematic* versions of Sloman's arguments (see note 25).

The reach of logic-based computational cognitive modeling can be shown to formalize not just RBR and SBR, but the overall phenomenon of *commonsense reasoning*, as this phenomenon is characterized by Sun (1995). Sun tells us that commonsense reasoning, while not strictly speaking definable, can be taken to include

informal kinds of reasoning in everyday life regarding mundane issues, where speed is oftentimes more critical than accuracy. The study of commonsense reasoning as envisaged here is neither about the study of a particular domain, nor about idiosyncratic reasoning in any particular domain. It deals with commonsense reasoning *patterns*; that is, the recurrent, domain-independent basic forms of reasoning that are applicable across a wide range of domains. (Sun 1995, p. 242)

Sun then goes on to show that such reasoning, insofar as a particular set of eleven examples can be taken as exemplars of this reasoning, can be modeled in the CONSYDERR architecture. While obviously there is insufficient space to show here how LCCM can also model commonsense reasoning characterized in this ostensive way, it can be indicated how such modeling would run.

[27]Information about Slate, and a for-teaching version of the system itself, can be obtained at

http://www.cogsci.rpi.edu/research/rair

.

The first of the eleven examples, all of which are taken from (Collins & Michalski 1989, Collins 1978), is as follows.

> Q: Do you think they might grow rice in Florida?
> A: Yeah. I guess they could, if there were an adequate fresh water supply, certainly a nice, big, warm, flat area.

About this example, Sun writes:

> There is a rule in this example: if a place is big, warm, flat, and has an adequate fresh water supply, then it is a rice-growing area. The person answering the question deduced an uncertain conclusion based on partial knowledge, although a piece of crucial information (i.e., the presence of fresh water) is absent. Sun 1995, p. 244)

One interesting aspect of this example is that the subject not only answers the question in the affirmative, but also sketches a justification. (You will recall that the structure of experiments designed to uncover the nature of reasoning and decision making are assumed herein to request justifications. See section 4.3.) The response is an enthymematic deductive argument (see note 25) easily expressed in $\mathcal{L}_I$, under the parameter $\vdash_F^D$ defined earlier in the chapter, as follows (with obvious meanings for the predicate letters).

	(1)	$\forall x((Place(x) \land Big(x) \land Warm(x) \land Flat(x) \land Water(x)) \rightarrow GrowRice(x))$	premise
	(2)	$Place(fl) \land Big(fl) \land Warm(fl) \land Flat(fl)$	premise
	(3)	$Water(x)$	probable
	(4)	$Place(fl) \land Big(fl) \land Warm(fl) \land Flat(fl) \land Water(fl)$	from (2), (3) BY $\land$I
	(5)	$Place(fl) \land Big(fl) \land Warm(fl) \land Flat(fl) \land Water(fl)) \rightarrow GrowRice(fl)$	from (1) BY $\forall$E
$\therefore$	(6)	$GrowRice(fl)$	from (5), (6) by $\rightarrow$Elim

Of course, while (4) follows deductively from {(1), (2), (3)} using the rules of natural deduction introduced earlier (as shown in the proof immediately above), (3) is only probable, which means that, overall, the strength of the argument for (6) is itself probable.

There are other formal niceties that would be included in a full exposition, but the point should be clear: Commonsense reasoning, as a declarative phenomenon, that is, as reasoning over declarative statements, can, from the formal point of view, be modeled in an illuminating way by LCCM. The subject in this case has given an answer, and a justification, whose formal essence can be expressed with the machinery of LCCM. All of the remaining ten examples in (Sun 1995) can be modeled in LCCM as well. Moreover, while it is not shown herein, using the concrete computational techniques covered in the next section, logic-based computer programs can be written and executed to produce rapid, real-time simulations of the commonsense reasoning in question. Finally, whatever empirical data might be associated with human commonsense reasoning (e.g., response time for answer to be produced and justification to be articulated) can be formally expressed in the framework of logic-based computational cognitive modeling.

4.3 Logic-Based Computer Programming

Needless to say, none of the foregoing has much value unless a program can be written which, when executed, produces a computational simulation of some cognition. After all, the subject being rationalized in this chapter is a particular paradigm for *computational* cognitive modeling, and computation is by definition the movement through time of a computer; and now and for the foreseeable future such movement is invariably caused and regulated by the creation (manual or automatic in nature, or a hybrid of the two) and execution of computer programs.[28] Given this, an obvious question is: With what same-level programming paradigm, and languages within it, is LCCM naturally associated? The answer to this question is straightforward, and comes in three parts, as follows.

First, in order to remind readers of the context, note that there are three programming paradigms (*procedural*, reflected, e.g., in Turing machines themselves, and in various "minimalist" languages like those seen in foundational computer science texts (e.g., Davis & Weyuker 1994, Pascal, etc.); *functional*, reflected, e.g., in Scheme, ML, and purely functional Common Lisp (Shapiro 1992, Abelson & Sussman 1996); and

[28]Even in the case of so-called hypercomputers, the machines in question must receive instructions. E.g., even Infinite Time Turing machines (Hamkins & Lewis 2000) are driven by instructions (= programs).

declarative, reflected, albeit weakly, in Prolog (Clocksin & Mellish 2003)), and unsurprisingly, the same-level programming paradigm, from the formal point of view, naturally associated with logic-based computational cognitive modeling is the declarative one.[29] This is not surprising because we noted at the outset that declarative/logic-based computational cognitive modeling takes declarative statements as fundamental units of information.

It's important to appreciate the adjective "*same-level*" in the previous paragraph. In theory, any Turing-computable function can be implemented through code written in any Turing-complete programming language. There is nothing in principle precluding the possibility of writing a program in assembly language that, at a higher level of abstraction, processes information in accordance with inference in many of the logical systems in the family $\mathcal{F}$ explained above. (In fact, as is well-known, the other direction is routine, as it occurs when a high-level computer program in, say, Prolog, is compiled to produce corresponding to low-level code; assembly language, for example.) However, this would not be *same-level* programming, and it would require a programming mindset that doesn't correspond to the declarative representational and design mindset reflected by logical systems in $\mathcal{F}$.

The second part of the answer to the question of what same-level programming paradigm is associated with logic-based computational cognitive modeling is this: Just as a generalization of the concept of *logical system* from mathematical logic was used to arrive at the family $\mathcal{F}$ of logical systems for LCCM, programs in logic-based computational cognitive modeling are written in programming languages from a family $\mathcal{P}$ composed of languages that are generalizations of the long-established concept of a *logic program* in computer science (succinctly presented, e.g., in the chapter "Logic Programming" in Ebbinghaus et al. 1994). For each system S in $\mathcal{F}$, there is a corresponding programming language P_S in $\mathcal{P}$. In the interests of space, and to simplify the exposition, the focus here will be on $P_{\mathcal{L}_{PC}}$ and $P_{\mathcal{L}_I}$, in which programs are written based on the computation of the two central relations in $\mathcal{L}_{PC}$ and $\mathcal{L}_I$, viz., $\vdash^D_X$ and $\models$, both defined earlier. (As will be recalled, $\Phi \vdash^D_X \psi$ holds iff ψ can be deductively inferred from Φ in deductive calculus X, and $\mathcal{I} \models \psi$ holds iff ψ is true on interpretation $\mathcal{I}$.) Fortunately, there are many well-established programming environments for writing programs based on the computing of these relations. For example, $\vdash^D_{Res}$ is computed by Vampire (Voronkov 1995) and Otter (Wos 1996, Wos et al. 1992), $\vdash^D_F$ by Oscar (Pollock 1989, Pollock 1995) and Athena and NDL (Arkoudas 2000, Bringsjord, Arkoudas & Bello 2006), among other such systems. As to $\models$, a number of mature, readily available systems now compute this relation as well, for example, Hyperproof (Barwise & Etchemendy 1994), and Paradox and Mace (Claessen & Sorensson 2003) at the level of $\mathcal{L}_I$, and at the level of $\mathcal{L}_{PC}$, many SAT solvers (e.g., see Kautz & Selman 1999).

Enough information is now provided to enable the reader to see how simulations that model human reasoning (triggered by the desiderata given in section 3.1) can be produced by programs written in $P_{\mathcal{L}_{PC}}$ and $P_{\mathcal{L}_I}$. To produce these simulations, we need to give to a program in either of these languages declarative knowledge corresponding to what the human knows, and then execute this code to produce the desired answer (the answer that corresponds to the answer given by the human cognizer), *and* the desired justification (the justification given by the human cognizer). It is important to realize that not just any justification will do: the justification produced by the machine must match that produced by the human.

In order to make all of this more concrete for readers, what follows is a program (or, better, an evaluable program) written in $P_{\mathcal{L}_{PC}}$, specifically in the generic, easy-to-understand denotational proof language NDL (Arkoudas 2000, Bringsjord et al. 2006) that corresponds directly to the system of natural deduction F presented in detail earlier in the chapter. This deduction, upon evaluation, produces a theorem in $\mathcal{L}_{PC}$ as output — a theorem that Newell and Simon's Logic Theorist, to great fanfare (because here was a machine doing what "smart" humans did), was able to muster at the dawn of AI in 1956, at the original Dartmouth AI conference. For a particular syntax, we follow the argument theory $\vdash^D_F$ introduced earlier: Fitch-style natural deduction, first presented in 1934 by two thinkers working independently to offer a format designed to capture human mathematical reasoning as it was and is expressed by real human beings: Gentzen (1935) and Jaskowski (1934). Streamlining of the formalism was carried out by Fitch (1952). The hallmark of this sort of deduction is that assumptions are made (and then discharged) in order to allow reasoning of the sort that human reasoners engage in. Now here is the deduction, commented to make it easy to follow.

```
// Here is the theorem to be proved,
```

[29]Readers should understand that each of the three programming paradigms corresponds to a seminal system invented long ago: Turing machines in the procedural case, the λ-calculus in the functional case, and first-order logic and associated provability in the declarative case.

```
// Logic Theorist's ``claim to fame'':
// (p ==> q) ==> (~q ==> ~p)

Relations p:0, q:0. // Here we declare that we have two
                    // propositional variables, p and q.
                    // They are defined as 0-ary relations.

// Now for the argument.  First, the antecedent (p ==> q)
// is assumed, and then, for contradiction, the antecedent
// (~q) of the consequent (~q ==> ~p).
assume p ==> q
   assume ~q
      suppose-absurd p
          begin
            modus-ponens p ==> q, p;
            absurd q, ~q
          end
```

If, upon evaluation, the desired theorem is produced, the program is successful. In the present case, sure enough, after the code is evaulated, one receives this back:

```
Theorem: (p ==> q) ==> (~q ==> ~p)
```

Now let us move up to programs written in $P_{\mathcal{L}_I}$ for simulations of cognition based on $\mathcal{L}_I$. As you will recall, this entails that the quantifiers $\exists x$ ('there exists at least one thing x such that ...') and $\forall x$ ('for all x ...') are admitted. In addition, there is now a supply of variables, constants, relations, and function symbols; these were discussed above. What follows is a simple NDL deduction in $P_{\mathcal{L}_I}$ that illuminates a number of the concepts introduced to this point. The code in this case, upon evaluation, yields the theorem that Tom loves Mary, given certain helpful information. It is important to note that both the answer and the justification have been assembled, and that the justification, since it is natural deduction, corresponds to the kinds of arguments often given by human beings.

```
Constants mary, tom.  // Two constants announced.

Relations Loves:2. // This concludes the simple signature, which
                   // here declares Loves to be a two-place relation.

// That Mary loves Tom is asserted:
assert Loves(mary, tom).

// 'Loves' is a symmetric relation, and this is asserted:
assert (forall x (forall y (Loves(x, y) ==> Loves(y, x)))).

//Now the evaluable deduction proper can be written:
suppose-absurd ~Loves(tom, mary)
   begin
      specialize (forall x (forall y (Loves(x, y) ==> Loves(y, x)))) with mary;
      specialize (forall y (Loves(mary, y) ==> Loves(y, mary))) with tom;
      Loves(tom,mary) BY modus-ponens Loves(mary, tom) ==> Loves(tom, mary), Loves(mary, tom);
      false BY absurd Loves(tom, mary), ~Loves(tom, mary)
   end;
Loves(tom,mary) BY double-negation ~~Loves(tom,mary)
```

When this program is evaluated, one receives the desired result back: `Theorem:  Loves(tom,mary)`. Once again, it is important to note that both the answer and the justification have been assembled, and that the justification, since it is natural deduction, corresponds to the kinds of proofs often given by human beings.

So far we have conceived of programs as proof-like entities. This takes care of the $\vdash_X^D$ in a logical system. But what about the semantic side? What about $\models$? What about interpretations, or models? In $P_{\mathcal{L}_I}$, programs can be written to produce, and to manipulate, models. Returning back once again to the abstract structure of problems given at the outset of the present chapter (see section), when the declarative information in some stimulus presented to a subject in the form of a list L of declarative statements doesn't allow a featured proposition D to be deductively inferred, a normatively correct justification is a *dis*proof: a proof the includes the presentation of a model on which all of the entries in L are true, but on which D is false. (Such models, for obvious reasons, are traditionally called *counter*models.) Rather than give examples of such processing here, we describe how programs written in $P_{\mathcal{L}_I}$ can produce and manipulate models in the next section (5), by turning to the Hyperproof (Barwise & Etchemendy 1994) system.

5 Meeting the Challenges

It's time now to turn to showing how the problems composing (C1) can be solved in LCCM in a manner that matches the human normatively incorrect and normatively correct responses returned after the relevant stimuli are presented. Recall, yet again, the ecumenical experimental structure to which declarative/logic-based computational cognitive modeling must conform (section 4.3).

5.1 Meeting (C1), the Challenge of Mechanizing Human Reasoning

Let's begin by reviewing the desiderata under (C1): Desideratum 1 is modeling both System 1 and System 2. Desideratum 2 is modeling reasoning that is emphasized by the three theories. Desiderata 3–6 consist of the sequence of four puzzles: King-Ace, Wine Drinker, Wise Man, and Infinitary DeMorgan. Now, how can it be shown that logic-based computational cognitive modeling can meet the six requirements? By providing the following six demonstrations:

D1 a normatively correct solution to King-Ace can be modeled by LCCM;

D2 a normatively *in*correct, mental logic-based response to King-Ace can be modeled by LCCM;

D3 a normatively correct mental meta-logic-based solution to Wine Drinker can be modeled by LCCM;

D4 a normatively incorrect mental models-based response to Wine Drinker can be modeled by LCCM;

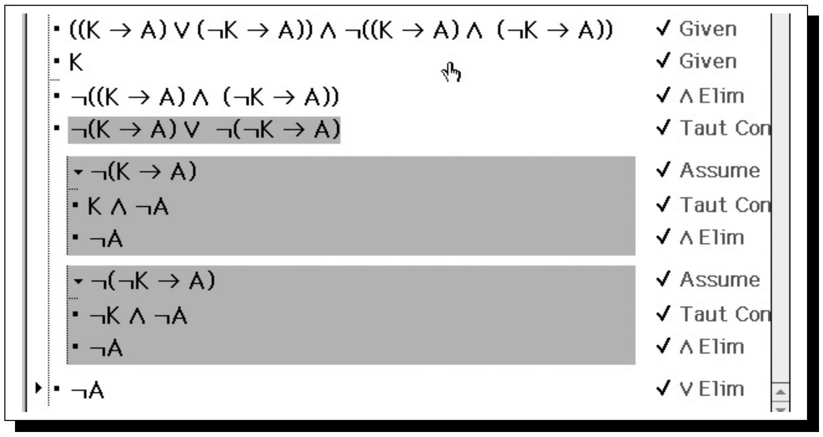
D5 a normatively correct solution to Wise Man can be modeled by LCCM.

D6 a normatively correct solution to Infinitary DeMorgan can be modeled by LCCM.

The reader, by elementary deduction, can satisfy herself that once these things are demonstrated, all desiderata are satisfied. These demonstrations, recall, are to be carried out at the "algorithmic" level in Marr's (1982) tripartite scheme, or, equivalently, the "symbolic" in Pylyshyn's (1984) corresponding three-level view of the computational modeling of cognition. Using Marr's language, that means that what is sought is a representation for the input and output, and the algorithm for the transformation from the former to the latter. The algorithm for transformation corresponds directly to the argument or proof provided. (Recall that what the algorithms in question are was provided in section 4.3, when logic-based computer programming was defined. The programming is based directly on arguments or proofs returned by subjects.)

5.1.1 D1 (N.C. King-Ace Modeled)

A normatively correct solution to Puzzle 1 that follows what human cognizers do when succeeding on the puzzle is effortless to construct in LCCM, with the logical system in question set to $\mathcal{L}_{PC}$. In a recent experiment in our laboratory (to test hypotheses outside the scope of the present chapter), 40 subjects were given Puzzle 1. The subjects were divided into two groups, one that was given a paper-and-pencil version of Puzzle 1, and one that was given an electronic version encoded in our Slate system. In both cases, a justification for the given answer was requested. A number of subjects did in fact answer correctly, *and* give a normatively correct justification, that is, a proof in a particular deductive calculus, namely the calculus F defined earlier. Figure 2 shows a proof in $\mathcal{F}$, constructed in HYPERPROOF, that follows the reasoning given by some students in proving that in Puzzle 1 one can correctly conclude $\neg A$. It is important to note that there are an unlimited number of deductive calculi that could be used to a proof establishing the correct answer "There is not an ace in the hand". The normative correct solution provided here is a direct match to the justification given by human subjects. For many examples of such normatively correct solutions produced by human subjects, see (Rinella et al. 2001).



```
▪ ((K → A) ∨ (¬K → A)) ∧ ¬((K → A) ∧ (¬K → A))   ✓ Given
▪ K                                              ✓ Given
▪ ¬((K → A) ∧ (¬K → A))                          ✓ ∧ Elim
▪ ¬(K → A) ∨ ¬(¬K → A)                           ✓ Taut Con
    ▾ ¬(K → A)                                   ✓ Assume
    ▪ K ∧ ¬A                                     ✓ Taut Con
    ▪ ¬A                                         ✓ ∧ Elim
    ▾ ¬(¬K → A)                                  ✓ Assume
    ▪ ¬K ∧ ¬A                                    ✓ Taut Con
    ▪ ¬A                                         ✓ ∧ Elim
▪ ¬A                                             ✓ ∨ Elim
```

Figure 2: A Proof That There is No Ace in the Hand in $\mathcal{F}$

5.1.2 D2 (N.I. Mental Logic-Based King-Ace Modeled)

The same experiment as mentioned in the previous section, combined with a number of predecessors relevantly like it, have enabled us to acquire an archive of "justifications" in support of A. The vast majority of these express reasoning that is in fact formally valid reasoning in conformity with mental logic theory — but in this reasoning, the declarative information is incorrectly represented. How does the reasoning run, specifically? It's perfectly straightforward; here is a sample:

"We know that if there's a king in the hand, then there's an ace in the hand. And we know that if there isn't a king in the hand, then there is an ace in the hand. But, there are only two possibilities here. Either there is a king in the hand, or there isn't. But both of these possibilities let us conclude that there is an ace in the hand."

Obviously, such reasoning accords well with mental logic, and a simulation of this reasoning in the LCCM approach is trivial. One need only write a program $P_{\mathcal{L}_{PC}}$ such that, when evaluated, the reasoning quoted immediately above is produced. Here's a simple NDL deduction that does the trick:

```
// The signature for this simple example (normatively incorrect
// deductive reasoning given in response to the king-ace puzzle)
// contains to propositional variables, A and K:
Relations A:0, K:0.

// One asserts the conjunction consisting of the claim that if there's
// a king in the hand, then there's an ace in the hand, and the claim
// that if there isn't a king in the hand then there's an ace in the
// hand.
assert ((K ==> A) & (~K ==> A))

// Either there's a king in the hand, or there isn't:
assert K \/ ~K

// And now for the argument, which mechnizes the idea that no
// mather which if-then one goes with, in either case one can
// show that there is an ace in the hand.
left-and ((K ==> A) & (~K ==> A));
right-and ((K ==> A) & (~K ==> A));
cases K \/ ~K, K ==> A, ~K ==> A
```

When evaluated, this returns a theorem exactly in line with what the normatively incorrect reasoning is supposed to produce, viz.,

```
Theorem: A
```

Once again, note that it is not just that the desired answer is produced. The structure of the justification directly models what is given by human subjects who fall prey to the puzzle.

5.1.3 D3 (N.C. Mental Meta-Logic-Based Wine Drinker Modeled)

How does $\mathcal{L}_I$ allow us to solve Puzzle 2? Recall yet again the three relevant statements, in English:

1. All the Frenchmen in the restaurant are gourmets.
2. Some of the gourmets are wine drinkers.
3. Some of the Frenchmen in the restaurant are wine drinkers.

The simplest solution to the puzzle is to note that one can find an interpretation $\mathcal{I}$ (in the logical system $\mathcal{L}_I$) in which the first two statements are true, but the third isn't. This will show that the third isn't a deductive consequence of the first two, from which it will immediately follow that the third cannot be proved from the first two. Here is an interpretation that fits the bill: First, assume that everyone we're talking about is in the restaurant. Now, suppose that Alvin is a wine-drinker and a gourmet, and not a Frenchman. Bertrand is a Frenchman and a gourmet, but not a wine drinker. No one else, in this imaginary scenario, exists. In this situation, all Frenchmen are gourmets, and there exists someone who is a wine-drinker and a gourmet. This ensures that both the first two statements are true. But it's *not* true that there exists someone who is both a Frenchman and a wine drinker. This means that third proposition is false; more generally, it means that the third isn't a consequence of the first two, which in turn means that (using the list of the three just given)

$$\{(1), (2)\} \vdash (3),$$

and the full solution is accomplished. Please note that logic-based programming at the level of $\mathcal{L}_I$ allows for countermodels to be produced, and they can be rendered in visual form to be more quickly grasped. Figure 3 shows such a countermodel relevant to the present case, produced by the aforementioned Paradox system (Claessen & Sorensson 2003), and translated and visually displayed by the Slate system (Bringsjord et al. 2007). For studies in which subjects respond to stimuli like the Wine Drinker in normatively correct fashion, see (Bringsjord et al. 1998, Rinella et al. 2001).

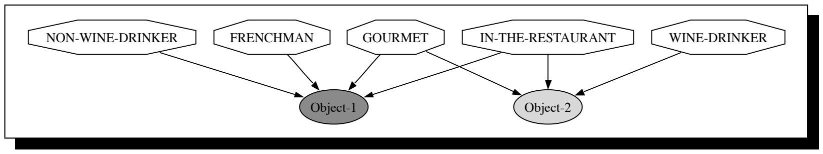


Figure 3: Visual Countermodel in Wine Drinker Puzzle (provided by Andrew Shilliday and Joshua Taylor)

5.1.4 D4 (N.I. Mental Models-Based Wine Drinker Modeled)

This is effortless to model in LCCM, as follows. First, most subjects who succumb to this problem see not the list of English sentences as written, but rather

1. All the Frenchmen in the restaurant are gourmets.
2. All of the gourmets are wine drinkers.
3. There are some Frenchman.
4. Some of the Frenchmen in the restaurant are wine drinkers.

The deduction of the last of these from the first three in a natural calculus is straightforward. Here is an NDL deduction that, once evaluated, produces exactly the human-produced output (i.e., exactly `(exists x (Frenchmen(x) & Winedrinker(x)))`, by exactly the human-produced reasoning:

```
// There are three obvious relations to declare:
Relations Frenchman:1, Gourmet:1, Winedrinker:1.

assert (forall x (Frenchman(x) ==> Gourmet(x)))  // The first proposition is asserted.

assert (forall x (Gourmet(x) ==> Winedrinker(x)))  // The second proposition is asserted.

assert (exists x Frenchman(x))  // There are some Frenchmen.
```

```
// Now for the reasoning corresponding to the normatively incorrect response.
// The reasoning itself, note, is formally valid.

pick-witness z for (exists x Frenchman(x))  // An arbitrary individual z is picked
                                               to facilitate the reasoning.
  begin
    specialize (forall x (Frenchman(x) ==> Gourmet(x))) with z;
    specialize (forall x (Gourmet(x) ==> Winedrinker(x))) with z;
    assume Frenchman(z)
      begin
        modus-ponens Frenchman(z) ==> Gourmet(z), Frenchman(z);
        modus-ponens Gourmet(z) ==> Winedrinker(z), Gourmet(z)
      end;
    modus-ponens Frenchman(z) ==> Winedrinker(z), Frenchman(z);
    both Frenchman(z), Winedrinker(z);
    ex-generalize (exists x Frenchman(x) & Winedrinker(x)) from z
  end
```

5.1.5 D5 (N.C. Wise Man Modeled)

To ease exposition, the solution is restricted to the two-wise man version. In this version, the key information consists in these three facts:

1. A knows that if A doesn't have a white spot, B will know that A doesn't have a white spot.
2. A knows that B knows that either A or B has a white spot.
3. A knows that B doesn't know whether or not B has a white spot.

Next, here are some key axioms and rules of inference:

K $\Box(\phi \Rightarrow \psi) \Rightarrow (\Box\phi \Rightarrow \Box\psi)$

T $\Box\phi \Rightarrow \phi$

LO ("logical omniscience") From $\phi \vdash^* \psi$ and $\mathbf{K}_\alpha\phi$ infer $\mathbf{K}_\alpha\psi$

We are now positioned to appreciate a traditional-style proof in $\mathcal{L}_{KT}$ that solves this problem, and which is the direct correlate given by (the few) subjects who, when WMP is given, provide a normatively correct justification:

1. $\mathbf{K}_A(\neg\text{White(A)} \Rightarrow \mathbf{K}_B(\neg\text{White(A)}))$
2. $\mathbf{K}_A(\mathbf{K}_B(\neg\text{White(A)} \Rightarrow \text{White(B)}))$
3. $\mathbf{K}_A(\neg\mathbf{K}_B(\text{White(B)}))$
4. $\neg\text{White(A)} \Rightarrow \mathbf{K}_B(\neg\text{White(A)})$ 1, T
5. $\mathbf{K}_B(\neg\text{White(A)} \Rightarrow \text{White(B)})$ 2, T
6. $\mathbf{K}_B\neg(\text{White(A)}) \Rightarrow \mathbf{K}_B(\text{White(B)})$ 5, K
7. $\neg\text{White(A)} \Rightarrow \mathbf{K}_B(\text{White(B)})$ 4, 6
8. $\neg\mathbf{K}_B(\text{White(B)}) \Rightarrow \text{White(A)}$ 7
9. $\mathbf{K}_A(\neg\mathbf{K}_B(\text{White(B)}) \Rightarrow \text{White(A)})$ 4–8, 1, LO
10. $\mathbf{K}_A(\neg\mathbf{K}_B(\text{White(B)})) \Rightarrow \mathbf{K}_A(\text{White(A)})$ 9, K
11. $\mathbf{K}_A(\text{White(A)})$ 3, 10

To see how this can be rendered in computational form, implemented, and efficiently run in a logic-based computer program, see (Arkoudas & Bringsjord 2005).

5.1.6 D6 (N.C. Infinitary DeMorgan)

Given the reach of LCCM through $\mathcal{L}_{\omega_1\omega}$, this puzzle is strikingly easy to solve, as some humans realize. The disjunction in question can be denoted by $\bigvee\Phi$. We then simply invoke the infinitary analogue to the inference rule known as disjunctive syllogism, which sanctions deducing ψ from the two formulas $\phi \vee \psi$ and $\neg\phi$. The analogue is

$$\text{from } \bigvee\Phi, \text{ where } \phi \in \Phi, \text{ and } \neg\phi, \text{ infer to } \bigvee\Phi - \{\phi\}$$

It's as easy as that.

5.2 Meeting (C2), the Perception/Action Challenge

(C2) can be solved if logic-based computational cognitive modeling can *transparently* model, on the strength of the core mechanical processes given by the families $\mathcal{F}$ and $\mathcal{P}$, the range of high-level cognition all the way down to non-deliberative interaction with the environment, or what, following contemporary terminology, can be called *external perception and action.*[30] In Ron Sun's words, discussed earlier, one can meet challenge (C2) if $\mathcal{F}$ and $\mathcal{P}$ constitute the unifying logico-mathematical language he says is sorely missing.

It has been shown above that logic-based computational cognitive modeling can model high-level reasoning. If it can be shown that LCCM can meet the perception-and-action challenge, large steps will have been taken toward showing that (C2) can be met by LCCM.[31]

Of course, there is a general feeling afloat that logic is unacceptably slow. Can LCCM handle rapid, non-deliberative perception and action, in an exchange with the physical environment? For example, can logic be used to model a human making his or her way through a rapid-fire first-person shooter computer game? In this section it is explained why this challenge (a) may be beside the point of modeling human personhood, (b) needs to be distinguished from so-called *transduction*, (c) can be met in at least two logic-based ways, one of which has already been successfully pursued to some degree, and one of which would be based on *visual* logic, an area of growing and great future importance to LCCM.

5.2.1 Is Perception and Action Beside the Point?

Note that non-deliberative, external perception and action is not part of the definition of human personhood given earlier in the chapter (section 2). The reason for that is well-known: In general, it seems entirely possible for us to *be* persons over a stretch of time during which no external perception and action occurs.[32] There is no reason why Smith can't spend three hours in a sensory deprivation tank, during which time he cracks a math problem, or writes a story in his head, or does any number of intellectual tasks. Moreover, it certainly seems mathematically possible that human persons could be brains in vats, having no exchange with the environment of the type that is supposed to be a challenge to logic-based computational cognitive modeling (Bringsjord & Zenzen 1991).

Nonetheless, it is charitably assumed that LCCM is challenged with having to model external perception and action. An explanation that this challenge can apparently be met is now provided.

5.2.2 Separating Out Transduction

It is important to distinguish between perception and action, and transduction. Transduction is the process by which data hitting sensors is transformed into information that can processed by an agent, and by which information processed by an agent is transformed into data emitted by effectors. Arguably, computational cognitive modeling should not be charged with having to capture transduction. Transduction is a purely physics- and engineering-relevant process having nothing to do with cognition. In other words, transduction is a process peripheral to human personhood. The quickest way to see this is to note that the transducers we currently have can be replaced with others, while the pre-replacement and post-replacement persons remain numerically identical despite this replacement. If you go blind, and doctors replace your eyes with artificial cameras, it's still you who thanks the surgeons after the procedure has brought your sight back. It's *your* sight they have brought back, after all. (The overall picture just described is articulated in the context of human-level logic-based AI in (Nilsson 1991). The picture transfers directly to the specific case of human persons.)

[30]The term 'external' is used because human persons do routinely engage in introspection (perceive internal things), and do carry out all sorts of mental (= internal) actions.

[31]Astute readers may wonder about learning. Please note that the notion that logic is inappropriate for modeling learning, which because of limited space isn't discussed in earnest herein, has certainly evaporated. This is so for two reasons. The first is that logic-based machine learning techniques are now well-established (for a nice survey, see Russell & Norvig 2002). The second reason is that machine learning by reading, which has never been pursued in AI or cognitive science, is now a funded enterprise — and is logic-based. For example, see the start of Project Halo (Friedland, Allen, Matthews, Witbrock, Baxter, Curtis, Shepard, Miraglia, Angele, Staab, Moench, Oppermann, Wenke, Israel, Chaudhri, Porter, Barker, Fan, Chaw, Yeh, Tecuci & Clark 2004), and logic-based machine reading research sponsored by the US government (e.g. see Bringsjord et al. 2007).

[32]*Internal* perception and action is another story: In a sensory deprivation tank, one can perceive all sorts of mathematical objects (e.g.), and can take all kinds of mental actions.

So, the assumption is made that for LCCM information from the environment is cast as expressions in some logical system from $\mathcal{F}$, and the challenge is to process those expressions with sufficient speed and accuracy to match human performance. This can be accomplished in one of two ways. The first way is briefly described in the next section. The second way is still experimental, and on the very frontier of LCCM and human-level logic-based AI, and will not be discussed here.[33]

5.2.3 A Calculus (Situation, Event, ...) with ATPs

While logic has been criticized as too slow for real-time perception-and-action-heavy computation, as you might see in the computational modeling of a human playing first-person shooter game (as opposed to a strategy game, which for obvious reasons fits nicely with the paradigm of LCCM), it has been shown that computation produced by the execution of programs in $P_{\mathcal{L}_I}$ is now so fast that it can enable the real-time behavior of a mobile robot simulating human behavior in a robust environment. This has been shown by having a logic-based mobile robot successfully navigate the wumpus world game, a staple in AI (Bringsjord et al. 2005), and a game that humans have long played. (See Figures 4 and 5.) This work parallels work done in John McCarthy's (logic-based) AI Lab that has shown it to be possible to control a real robot, operating in a realistic office environment in real time (Amir & Maynard-Reid 2001, Amir & Maynard-Reid 2000, Amir & Maynard-Reid 1999).[34] In this approach, a calculus is used to represent time and change. Usually the calculus is the situation calculus, but the event calculus can also be used; both are summarized in (Russell & Norvig 2002). It's important to know that such work is far from peripheral and tentative: logic-based AI is starting to reveal that even in the area of perception and action, the speed demands can be met via well-established techniques that are part of the standard toolkit for the field, as seen by such textbooks as (Reiter 2001).

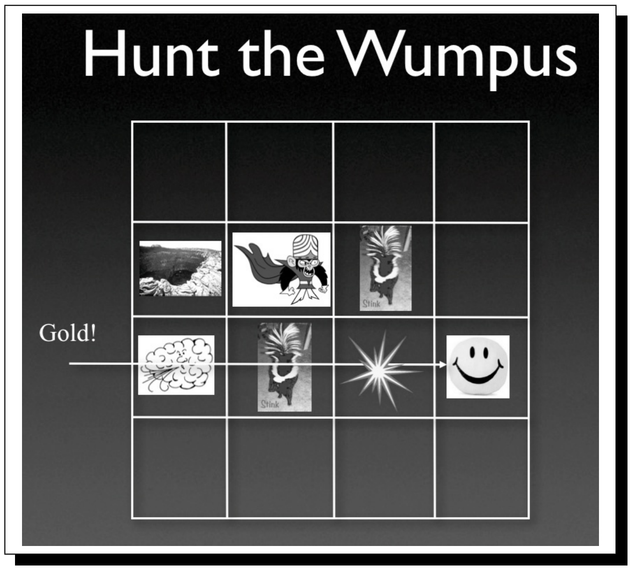


Figure 4: The Wumpus World Game. *In the wumpus world, a robot must navigate a work in matrix form, where cells in the grid may contain pits or a monster (the Wumpus). The robot must shoot and kill the Wumpus, and retrieve the gold.*

[33]In the second way, information from the environment is not transduced into traditional linguistic logics, but is rather left in visual form, and represented in visual logics. For a discussion of visual logic, in the context of the study of the mind from a computational perspective, see (Bringsjord in-press).

[34]This research can be found online at: http://www-formal.stanford.edu/eyal/lsa.

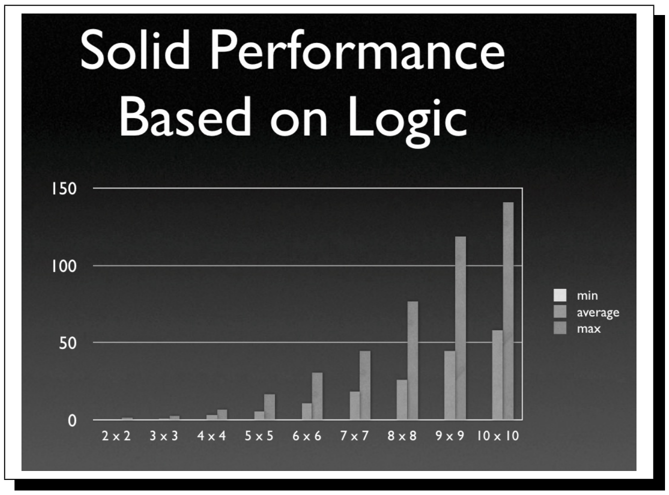


Figure 5: Performance of a RASCALS-Powered Robot in the Wumpus World. *This graph shows the time (in secs) it takes the logic-powered robot to succeed in the wumpus world, as a function of the size of the world (i.e., the size of the grid). The speed is really quite remarkable. Engineering was carried out by Matt Daigle.*

5.3 Meeting (C3), Meeting the Rigor Challenge

It is now briefly explained why it is that every computational model produced in accordance with logic-based computational cognitive modeling has a precise meaning, which allows LCCM to be theorem-guided.[35] Space does not permit a sampling of relevant theorems to be canvassed in any detail, but a few are cited at the end of the present section. For example, it is explained how it can be determined whether two different logic-based programs P and P' have the same meaning.

Let P_L be a logic-based computer program from the space $\mathcal{P}$. It follows immediately by definitions given above that this program conforms to what has been called the *argument semantics* for the logical system L in $\mathcal{F}$. That is, every inference made when P_L is executed has a precise mechanical meaning in terms of the effect this inference has on the knowledge base associated with this program. (This has been seen firsthand by the reader earlier, in the sample logic-based computer programs that have been provided.) This meaning perfectly coincides with corresponding inferences made when reasoning is carried out in the logical system L using a particular mode of inference, and a particular calculus. To make this clear, consider the precise meaning (albeit in English) of one of the inferences made use of in one of the sample logic-based computer programs presented above; that inference is `cases`, as used for example in

```
    cases K \/ ~K, K ==> A, ~K ==> A
```

The meaning of this inference is that, assuming that the knowledge base contains the three formulas in question (the disjunction `K    K` and the two conditionals `K ==> A` and `K ==> A`), its application will add to the knowledge base the formula `A`. This is the kind of meaning that is regimented through the *argument semantics* element in the six elements that are needed for each logical system; recall section 4.1. For each and every inference form, there is likewise a definition of this sort to fixes the meaning of that inference. As a result, any sequence of inferences has an absolutely clear meaning. Since every logic-based computer program is nothing more than the specification of a sequence of inferences, the meaning of the operation of a logic-based program is likewise entirely clear.

But what about the meaning of the formulas that are involved *in* the inferences? Here too precision is guaranteed. This is so because each and every formula appearing in a logic-based program is given a precise meaning via the formal semantics of the logical system that the formula is expressed in. As to why this is so, you have only to look back at the formula semantics for $\mathcal{L}_I$, given above. One can determine, for every

[35]Of course, though every model has a clear meaning, there is nothing intrinsic to the logic-based paradigm that would necessarily prevent a human modeler from midescribing human cognition. If Jones believes that everyone loves someone, and Smith, in seeking to model this belief, codes Jones' belief as $\forall x \forall y Lxy$, then no amount of precision in the paradigm will rescue the situation.

formula in every logic-based computer program, what the meaning of this formula is, because one has on hand an interpretation specifying the formal meaning of all the elements in the signature of every program. Look back to any of the sample logic-based programs given above, and see there, at the beginning of the file, declarations of relations (and sometimes constants). One has on hand an interpretation $\mathcal{I}$ telling us what these relations and constants mean. This pattern holds not just for programs written under $\mathcal{L}_I$ (i.e., programs from $P_{\mathcal{L}_I}$), but for any logical system in the family $\mathcal{F}$. In short, while in declarative computational cognitive modeling it may happen that a declarative statement ϕ is employed, in the formalization of such modeling in LCCM, the machinery must be in place for mechanically determining the meaning of ϕ.

Given the logico-mathematical precision of LCCM, declarative computational cognitive modeling can, thankfully, be guided by theorems. Of course, theorem guidance is not something that can be counted upon to be met with universal acclaim. There may well be those of the heterodox view that guidance by the light of mathematics is unwanted. However, there can be no denying the *effectiveness* of mathematics in not only describing, but predicting, the natural world, whether that description and prediction is pitched at the level of the 'algorithmic' (like formal economics, computer science, and computational cognitive modeling), or at the lower levels at which physics and chemistry operate (as has been famously pointed out in the 20th century; e.g., see Wigner 1960, Hamming 1980). To the extent that computational cognitive modeling takes the cognition distinctive of human persons to be natural phenomena that ought not only be carefully described, but predicted as well, theorem guidance would certainly be a welcome development; and in the case of at least declarative computational cognitive modeling, this development is achieved by virtue of LCCM.

There is not sufficient space, and this is not the right venue, to begin to give some interesting theorems, but it should be pointed out that many such theorems can now be proved in connection with the discussion above. For example, one can prove without much effort that simulations (i.e., computational cognitive models) in LCCM produced by programs at the level of $\mathcal{L}_{KT}$ will never go into infinite loops (assuming no syntactic bugs). On the other hand, because $\mathcal{L}_I$ is only semi-decidable (the theorems are in any decent textbook on intermediate mathematical logic, e.g., see Boolos & Jeffrey 1989), simulations in LCCM produced by programs at the level of this logical system can enter infinite loops, and explicit timeout catches must be included for all but very simple programs. For a more general example, note that given the foregoing, it is now known exactly when two logic-based computer programs $P_{\mathcal{L}}$ and $P'_{\mathcal{L}}$ have the same meaning under some interpretation $\mathcal{I}$: This equivalence holds provided that (i) both programs, given declarative input Φ (declarative sentences expressed as formulae in logical system $\mathcal{L}$), once executed, produce the very same theorems as output; and (ii), the formulas in Φ, as well as those used in the execution of the two programs, have the same meaning under $\mathcal{I}$.

6 Limitations and the Future

What can be said about the future of computational cognitive modeling, and in particular declarative/logic-based computational cognitive modeling? As readers well know, the future of any field is notoriously difficult to predict. Nonetheless, in the present case, present-day deficiencies in computational cognitive modeling, and specifically in LCCM, clearly point the way toward what cognitive modelers will in the future attempt to do. So, in a limited sense, the future can be accurately predicted, as follows.

What are the deficiencies? First, while Turing (1950) predicted over half a century back that by now we would be able to engineer machines linguistically indistinguishable from us (i.e., machines able to pass his so-called "Turing Test"), the fact of the matter is that, today, a bright toddler's conversational reach still exceeds that of any and all computers on our planet. This situation parallels the sad state of computational cognitive modeling when it comes to language: No robust computational cognitive models of human-level communication (attribute 4 in the list of capacities constitutive of personhood, given in section 2) exist. Even Anderson (2003), a sanguine champion of ACT-R, concedes that the linguistic side of computational cognitive modeling has essentially gone nowhere; that in this regard "Newell's Program" has not succeeded. Not only that, but no light can even be seen at the end of the tunnel. There are those (e.g., Moravec 1999) who hold that, relatively soon, person-level communication will be mechanized. Unfortunately, such writers are confident because of the continuous increase in processing speed produced by Moore's Law, but raw processing speed is not the problem (as explained in Bringsjord 2000): the challenge, surely, is to discover the information-processing procedures that enable human persons to communicate in natural languages. However fast the hardware, it does little good unless there are procedures to run upon it. It can therefore be said with confidence that computational cognitive modeling will in the future see sustained work in the area of language-based communication. Breakthroughs are waiting to be made in this area.

What are the implications of this specifically for declarative/logic-based computational cognitive modeling? At the dawn of AI in the States, when AI was what is today called human-level AI, and for at least three decades thereafter, the dream was to capture natural languages like English, German, and Norwegian completely in first-order logic (= in $\mathcal{L}_I$) (e.g., see the FOL-based Charniak & McDermott 1985). Unfortunately, this specific logic-based approach has not succeeded. In fact, some originally logic-based experts in computational language processing have turned their backs on logic, in favor of purely statistical approaches. Charniak is an example. In 1985, his comprehensive-at-the-time *Introduction to Artificial Intelligence* gave a strikingly unified presentation of AI, including natural language processing. This unification was achieved via first-order logic (= $\mathcal{L}_I$), which runs throughout the book and binds things together. But Charniak abandoned logic in favor of purely statistical approaches (Charniak 1993).

To this point, despite the richness of the families $\mathcal{F}$ and $\mathcal{P}$, natural language has resisted attempts to model is in logico-computational terms. However, it seems clear that some traction has taken hold in the attempt to model *fragments* of natural language in formal logic (e.g., see Fuchs, Schwertel & Schwitter 1999), and this direction is certain to see more investment, and at least some progress. Only time will tell is this research and development will be able to scale up to all of natural language.

A second present-day deficiency in computational cognitive modeling is consciousness. That is, there are today no simulations of consciousness (attribute 2 in the list of capacities constitutive of personhood; again, recall section 2). (There *are* simulations that encourage humans seeing these simulations to ascribe consciousness to them. But that is quite different.) As was pointed out in section 2, no one has a third-person account of what it is to (say) experience the taste of deep, dark chocolate, or what it is to *be* you (Bringsjord 1998*a*). Absent such an account, mechanization — indeed, taking just initial steps toward some mechanization — is impossible. Notice that while you may disagree about what is *ultimately* mechanizable, you must concede that, at least as of *now*, we have no third-person formalization of consciousness. In other words, property dualism (the view that such properties as "experiencing the taste of deep, dark chocolate" are incorporeal, though they may be correlated with certain physical properties of the brain) may be false, but it currently can't be overthrown by providing the third-person description of the physical properties that are identical to the psychological (or, as they are sometimes called in philosophy, "Cartesian") ones. Given the importance of consciousness in human cognition (after all, the reason humans seek to continue to live is to continue to have conscious experiences), there is little doubt that in the future computational cognitive modeling will be marked by a persistent attempt to express consciousness in computation. Again, breakthroughs are waiting to be made. Unfortunately, declarative/logic-based computational cognitive modeling would appear to be ill-suited to producing those breakthroughs. The reason is that such modeling is by definition focused on content that is fully and completely third-person in form: declarative representations are paradigmatically third-person in kind. It must be confessed that, at least at present, it is frankly hard to see how LCCM will rise above the limitation that subjective consciousness seems impossible to formalize in the form of propositions, and reasoning over those propositions.

The present chapter has emphasized human reasoning, as the reader well knows by now. But only reasoning in connection with specific puzzles has been considered. What about the future of attempts to computationally simulate *robust* human reasoning within the declarative/logic-based paradigm? Here it would seem that two prudent predictions can be made, given the current state of the art, and severe limitations seen within it.

As to trends, there is a growing desire to engineer simulations not only of the sort of relatively simple reasoning required to solve the puzzles analyzed above, but of the sort of real-life reasoning seen in the proofs of historic theorems. Gödel's incompleteness theorems are in this category, and recently some attempts have been made to build computational simulations of the reasoning involved (Sieg & Field 2005, Quaife 1988, Shankar 1994). When one looks closely at this work, it becomes clear that these efforts are based on giving the computer, at the start of its computation, knowledge the formation of which was the majority of the battle in Gödel's own case (as explained in Bringsjord 1998*b*). In the future, researchers will increasingly attempt to construct computational simulations of the production of such theorems, where the starting point involves only some basic knowledge. In other words, the attempt will be made to simulate the human ability to invent or create from scratch, rather than to simply process pre-defined representations. Declarative/logic-based computational cognitive modeling, as the present chapter shows, can provide impressive simulations when the declarative content is provided ahead of time. But what about the process of *generating* such content in the first place? This, currently, is a serious limitation, and it points toward a future in which much effort will be expended to surmount it.

7 Conclusion

This chapter has explained logic-based computational cognitive modeling as a formal, esemplastic rationalization of declarative computational cognitive modeling. It has also presented the attempt to build computational simulations of all, or at least large portions of, human cognition, on the basis, fundamentally, of logic and logic alone, where 'logic' here denotes the sense of 'logical system' explained above, and the vast family $\mathcal{F}$. This chapter has not argued that this engineering should be pursued; it has not argued that something is gained by pursuing a full-blown unified theory of cognition. The absence of unification has been famously bemoaned rather long ago by Newell (1973). Though such complaints are generally regarded to be compelling even to this day, it must be admitted that they are not sensitive to the fact that, in other fields more mature and (at least hitherto) more rigorous (e.g., physics) than computational cognitive modeling, unification is regarded to be of little or no value by many, if not most, researchers in these fields. There may be no grand synthesis between quantum mechanics and special relativity, but that doesn't stop physics from advancing year by year, and the benefits of that advance, from medical imaging to space exploration, are myriad. It's frankly a bit old-fashioned nowadays to yearn, in physics, for a grand synthesis.

To put the point another way, *if* one ought to pursue declarative computational models of all of human cognition in a unified fashion, logic-based computational cognitive modeling provides a rigorous route for the pursuit. But the antecedent of this conditional has not been established in the present chapter. In the end, only the use of logic would allow one to rationally determine if the antecedent should be affirmed: after all, only rational arguments either way could carry the day, since we are engaged in science. In fact, only the use of a particular logical system in which to express and evaluate this argumentation will allow the issue to be rationally settled. Ergo, logic-based computational cognitive modeling is configured, by virtue of the formalisms that constitute it, to allow for the construction of cognitive models of the cognition stimulated by this very chapter.

References

Abelson, H. & Sussman, G. (1996), *Structure and Interpretation of Computer Programs (2nd Edition)*, MIT Press, Cambridge, MA.

Amir, E. & Maynard-Reid, P. (1999), Logic-based subsumption architecture, *in* 'Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence'.

Amir, E. & Maynard-Reid, P. (2000), Logic-based subsumption architecture: Empirical evaluation, *in* 'Proceedings of the AAAI Fall Symposium on Parallel Archittectures for Cognition'.

Amir, E. & Maynard-Reid, P. (2001), LiSA: A robot driven by logical subsumption, *in* 'Proceedings of the Fifth Symposium on the Logical Formalization of Commonsense Reasoning'.

Anderson, J. & 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. & Lebiere, C. (1998), *The Atomic Components of Thought*, Lawrence Erlbaum, Mahwah, NJ.

Arkoudas, K. (2000), Denotational Proof Languages, PhD thesis, MIT.

Arkoudas, K. & Bringsjord, S. (2005), Metareasoning for multi-agent epistemic logics, *in* 'Fifth International Conference on Computational Logic In Multi-Agent Systems (CLIMA 2004)', Vol. 3487 of *Lecture Notes in Artificial Intelligence (LNAI)*, Springer-Verlag, New York, pp. 111–125.

Ashcraft, M. (1994), *Human Memory and Cognition*, HarperCollins, New York, NY.

Barwise, J. & Etchemendy, J. (1994), *Hyperproof*, CSLI, Stanford, CA.

Barwise, J. & Etchemendy, J. (1999), *Language, Proof, and Logic*, Seven Bridges, New York, NY.

Boolos, G. S. & Jeffrey, R. C. (1989), *Computability and Logic*, Cambridge University Press, Cambridge, UK.

Bourbaki, N. (2004), *Elements of Mathematics: Theory of Sets*, Verlag, New York, NY. This is a recent release. The original publication date was 1939.

Brachman, R. J. & Levesque, H. J. (2004), *Knowledge Representation and Reasoning*, Morgan Kaufmann/Elsevier, San Francisco, CA.

Braine, M. (1998*a*), How to investigate mental logic and the syntax of thought, *in* M. Braine & P. O'Brien, eds, 'Mental Logic', Lawrence Erlbaum, Mahwah, NJ, pp. 45–61.

Braine, M. (1998*b*), Steps toward a mental predicate-logic, *in* M. Braine & D. O'Brien, eds, 'Mental Logic', Lawrence Erlbaum Associates, Mahwah, NJ, pp. 273–331.

Braine, M. D. S. (1990), 'On the relation between the natural logic of reasoning and standard logic', *Psychological Review* **85**, 1–21.

Bringsjord, S. (1995), 'In defense of impenetrable zombies', *Journal of Consciousness Studies* **2**(4), 348–351.

Bringsjord, S. (1997), *Abortion: A Dialogue*, Hackett, Indianapolis, IN.

Bringsjord, S. (1998*a*), 'Chess is too easy', *Technology Review* **101**(2), 23–28.

Bringsjord, S. (1998*b*), 'Is Gödelian model-based deductive reasoning computational?', *Philosophica* **61**, 51–76.

Bringsjord, S. (1999), 'The zombie attack on the computational conception of mind', *Philosophy and Phenomenological Research* **59.1**, 41–69.

Bringsjord, S. (2000), 'A contrarian future for minds and machines', *Chronicle of Higher Education* p. B5. Reprinted in *The Education Digest* **66.6**: 31–33.

Bringsjord, S. (2001), 'Is it possible to build dramatically compelling interactive digital entertainment (in the form, e.g., of computer games)?', *Game Studies* **1**(1). This is the inaugural issue. Url: `http://www.gamestudies.org`.

Bringsjord, S. (in-press), Artificial intelligence, *in* E. Zalta, ed., 'The Stanford Encyclopedia of Philosophy', CSLI, Palo Alto, CA. http://plato.stanford.edu.

Bringsjord, S., Arkoudas, K. & Bello, P. (2006), 'Toward a general logicist methodology for engineering ethically correct robots', *IEEE Intelligent Systems* **21**(4), 38–44.

Bringsjord, S., Arkoudas, K., Clark, M., Shilliday, A., Taylor, J., Schimanski, B. & Yang, Y. (2007), Reporting on some logic-based machine reading research, *in* 'Proceedings of the 2007 AAAI Spring Symposium: Machine Reading', Menlo Park, CA.

Bringsjord, S., Bringsjord, E. & Noel, R. (1998), In defense of logical minds, *in* 'Proceedings of the 20^{th} Annual Conference of the Cognitive Science Society', Lawrence Erlbaum, Mahwah, NJ, pp. 173–178.

Bringsjord, S. & Ferrucci, D. (1998*a*), 'Logic and artificial intelligence: Divorced, still married, separated...?', *Minds and Machines* **8**, 273–308.

Bringsjord, S. & Ferrucci, D. (1998*b*), 'Reply to Thayse and Glymour on logic and artificial intelligence', *Minds and Machines* **8**, 313–315.

Bringsjord, S., Khemlani, S., Arkoudas, K., McEvoy, C., Destefano, M. & Daigle, M. (2005), Advanced synthetic characters, evil, and E, *in* M. Al-Akaidi & A. E. Rhalibi, eds, 'Game-On 2005, 6th International Conference on Intelligent Games and Simulation', European Simulation Society, Ghent-Zwijnaarde, Belgium, pp. 31–39.

Bringsjord, S., Noel, R. & Caporale, C. (2000), 'Animals, zombanimals, and the total Turing test: The essence of artificial intelligence', *Journal of Logic, Language, and Information* **9**, 397–418.

Bringsjord, S. & Yang, Y. (2003), Logical illusions and the welcome psychologism of logicist artificial intelligence, *in* D. Jacquette, ed., 'Philosophy, Psychology, and Psychologism: Critical and Historical Essays on the Psychological Turn in Philosophy', Kluwer, Dordrecht, The Netherlands, pp. 289–312.

Bringsjord, S. & Zenzen, M. (1991), In defense of hyper-logicist AI, *in* 'IJCAI 91', Morgan Kaufman, Moutain View, CA, pp. 1066–1072.

Bringsjord, S. & Zenzen, M. (2003), *Superminds: People Harness Hypercomputation, and More*, Kluwer Academic Publishers, Dordrecht, The Netherlands.

Brooks, R. (1991), 'Intelligence without representation', *Artificial Intelligence* **47**, 139–159.

Brooks, R. A., Breazeal, C., Marjanovic, M., Scassellati, B. & Williamson, M. M. (1999), 'The cog project: Building a humanoid robot', *Lecture Notes in Computer Science* **1562**, 52–87.

Bucciarelli, M. & Johnson-Laird, P. (1999), 'Strategies in syllogistic reasoning', *Cognitive Science* **23**, 247–303.

Bumby, Klutch, Collins & Egbers (1995), *Integrated Mathematics Course 1*, Glencoe/McGraw Hill, New York, NY.

Cassimatis, N. (2002), Polyscheme: A Cognitive Architecture for Integrating Multiple Representation and Inference Schemes, PhD thesis, Massachusetts Institute of Technology (MIT).

Cassimatis, N. (2006), 'Cognitive substrate for human-level intelligence', *AI Magazine* **27**(2), 71–82.

Cassimatis, N., Trafton, J., Schultz, A. & Bugajska, M. (2004), Integrating cognition, perception and action through mental simulation in robots, *in* 'Proceedings of the 2004 AAAI Spring Symposium on Knowledge Representation and Ontology for Autonomous Systems'.

Charniak, E. (1993), *Statistical Language Learning*, MIT Press, Cambridge, MA.

Charniak, E. & McDermott, D. (1985), *Introduction to Artificial Intelligence*, Addison-Wesley, Reading, MA.

Chisholm, R. (1966), *Theory of Knowledge*, Prentice-Hall, Englewood Cliffs, NJ.

Chisholm, R. (1977), *Theory of Knowledge 2nd ed*, Prentice-Hall, Englewood Cliffs, NJ.

Chisholm, R. (1978), 'Is there a mind-body problem?', *Philosophic Exchange* **2**, 25–32.

Chisholm, R. (1987), *Theory of Knowledge 3rd ed*, Prentice-Hall, Englewood Cliffs, NJ.

Claessen, K. & Sorensson, N. (2003), New techniques that improve Mace-style model finding, *in* 'Model Computation: Principles, Algorithms, Applications (Cade-19 Workshop)', Miami, Florida.

Clocksin, W. & Mellish, C. (2003), *Programming in Prolog (Using the ISO Standard; 5th Edition)*, Springer, New York, NY.

Collins, A. (1978), Fragments of a theory of human plausible reasoning, *in* D. Waltz, ed., 'Theoretical Issues in Natural Language Processing II', University of Illinois Press, Urbana, IL, pp. 194–201.

Collins, A. & Michalski, R. (1989), 'The logic of plausible reasoning: A core theory', *Cognitive Science* **82**, 1–49.

Davis, M., Sigal, R. & Weyuker, E. (1994), *Computability, Complexity, and Languages: Fundamentals of Theoretical Computer Science*, Academic Press, New York, NY.

Dennett, D. (1978), Conditions of personhood, *in* 'Brainstorms: Philosophical Essays on Mind and Psychology', Bradford Books, Montgomery, VT, pp. 267–285.

Devlin, K. (2000), *The Math Gene*, Basic Books, New York, NY.

Dickmann, M. A. (1975), *Large Infinitary Languages*, North-Holland, Amsterdam, The Netherlands.

Ebbinghaus, H. D., Flum, J. & Thomas, W. (1984), *Mathematical Logic*, Springer-Verlag, New York, NY.

Ebbinghaus, H. D., Flum, J. & Thomas, W. (1994), *Mathematical Logic (second edition)*, Springer-Verlag, New York, NY.

Eisenstadt, S. & Simon, H. (1997), 'Logic and thought', **7**(3), 365–385.

Ericsson, K. A. & Simon, H. (1984), *Protocol Analysis: Verbal Reports as Data*, MIT Press, Cambridge, MA.

Fagin, R., Halpern, J., Moses, Y. & Vardi, M. (2004), *Reasoning About Knowledge*, MIT Press, Cambridge, MA.

Fitch, F. (1952), *Symbolic Logic: An Introduction*, Ronald Press, New York, NY.

Friedland, N., Allen, P., Matthews, G., Witbrock, M., Baxter, D., Curtis, J., Shepard, B., Miraglia, P., Angele, J., Staab, S., Moench, E., Oppermann, H., Wenke, D., Israel, D., Chaudhri, V., Porter, B., Barker, K., Fan, J., Chaw, S. Y., Yeh, P., Tecuci, D. & Clark, P. (2004), 'Project halo: Towards a digital aristotle', *AI Magazine* pp. 29–47.

Fuchs, N. E., Schwertel, U. & Schwitter, R. (1999), Attempto Controlled English (ACE) Language Manual, Version 3.0, Technical Report 99.03, Department of Computer Science, University of Zurich, Zurich, Switzerland.

Gabbay, D., ed. (1994), *What is a Logical System?*, Clarendon Press, Oxford, UK.

Genesereth, M. & Nilsson, N. (1987), *Logical Foundations of Artificial Intelligence*, Morgan Kaufmann, Los Altos, CA.

Gentzen, G. (1935), 'Untersuchungen über das logische Schlieben I', *Mathematische Zeitschrift* **39**, 176–210.

Glymour, C. (1992), *Thinking Things Through*, MIT Press, Cambridge, MA.

Goble, L., ed. (2001*a*), *The Blackwell Guide to Philosophical Logic*, Blackwell Publishers, Oxford, UK.

Goble, L., ed. (2001*b*), *The Blackwell Guide to Philosophical Logic*, Blackwell Publishing, Oxford, UK.

Goldstein, E. B. (2005), *Cognitive Psychology: Connecting Mind, Research, and Everyday Experience*, Wadsworth, Belmont, CA.

Halpern, J., Harper, R., Immerman, N., Kolaitis, P., Vardi, M. & Vianu, V. (2001), 'On the unusual effectiveness of logic in computer science', *The Bulletin of Symbolic Logic* **7**(2), 213–236.

Hamkins, J. D. & Lewis, A. (2000), 'Infinite time Turing machines', *Journal of Symbolic Logic* **65**(2), 567–604.

Hamming, R. (1980), 'The unreasonable effectiveness of mathematics', *The American Mathematical Monthly* **87**, 81–90.

Hintikka, J. (1962), *Knowledge and Belief: An Introduction to the Logic of the Two Notions*, Cornell University Press, Ithaca, NY.

Inhelder, B. & Piaget, J. (1958), *The Growth of Logical Thinking from Childhood to Adolescence*, Basic Books, New York, NY.

Jaskowski, S. (1934), 'On the rules of suppositions in formal logic', *Studia Logica* **1**.

Johnson-Laird, P. (1997), 'Rules and illusions: A criticial study of Rips's *The Psychology of Proof*', *Minds and Machines* **7**(3), 387–407.

Johnson-Laird, P. N. (1983), *Mental Models*, Harvard University Press, Cambridge, MA.

Johnson-Laird, P. N., Legrenzi, P., Girotto, V. & Legrenzi, M. S. (2000), 'Illusions in reasoning about consistency', *Science* **288**, 531–532.

Johnson-Laird, P. & Savary, F. (1995), How to make the impossible seem probable, *in* 'Proceedings of the 17th Annual Conference of the Cognitive Science Society', Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 381–384.

Kahneman, D. & Tversky, A., eds (2000), *Choices, Values, and Frames*, Cambridge University Press, Cambridge, UK.

Kautz, H. & Selman, B. (1999), Unifying SAT-based and graph-based planning, *in* J. Minker, ed., 'Workshop on Logic-Based Artificial Intelligence, Washington, DC, June 14– 16, 1999', Computer Science Department, University of Maryland, College Park, Maryland.
URL: *citeseer.ist.psu.edu/kautz99unifying.html*

Langley, P., McKusick, K. B., Allen, J. A., Iba, W. & Thompson, K. (1991), 'A design for the icarus architecture', *SIGART Bulletin* **2**(4), 104–109.
URL: *citeseer.ist.psu.edu/langley91design.html*

Marr, D. (1982), *Vision: A Computational Approach*, Freeman and Company, San Francisco, CA.

McKeon, R., ed. (1941), *The Basic Works of Aristotle*, Random House, New York, NY.

Metzler, J. & Shepard, R. (1982), Transformational studies of the internal representations of three-dimensional objects, *in* R. Shepard & L. Cooper, eds, 'Mental Images and Their Transformations', MIT Press, Cambridge, MA, pp. 25–71.

Moravec, H. (1999), *Robot: Mere Machine to Transcendant Mind*, Oxford University Press, Oxford, UK.

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.

Newell, A. (1990), *Unified Theories of Cognition*, Harvard University Press, Cambridge, MA.

Newstead, S. E. & Evans, J. S. T., eds (1995), *Perspectives on Thinking and Reasoning*, Lawrence Erlbaum, Englewood Cliffs, NJ.

Nilsson, N. (1991), 'Logic and Artificial Intelligence', *Artificial Intelligence* **47**, 31–56.

Nilsson, N. (1995), 'Eye on the prize', *AI Magazine* **16**(2), 9–16.

Nilsson, N. (2005), 'Human-level artificial intelligence? Be serious!', *AI Magazine* **26**(4), 68–75.

Nute, D. (1984), Conditional logic, *in* D. Gabay & F. Guenthner, eds, 'Handbook of Philosophical Logic Volume II: Extensions of Classical Logic', D. Reidel, Dordrecht, The Netherlands, pp. 387–439.

Penrose, R. (1997), Physics and the mind, *in* M. Longair, ed., 'The Large, the Small, and the Human Mind', Cambridge University Press, Cambridge, UK, pp. 93–143.

Pollock, J. (1974), *Knowledge and Justification*, Princeton University Press, Princeton, NJ.

Pollock, J. (1989), *How to Build a Person: A Prolegomenon*, MIT Press, Cambridge, MA.

Pollock, J. (1995), *Cognitive Carpentry: A Blueprint for How to Build a Person*, MIT Press, Cambridge, MA.

Pollock, J. (2001), 'Defasible reasoning with variable degrees of justification', *Artificial Intelligence* **133**, 233–282.

Pollock, J. L. (1992), 'How to reason defeasibly', *Artificial Intelligence* **57**(1), 1–42.
URL: *citeseer.ist.psu.edu/pollock92how.html*

Pylyshyn, Z. (1984), *Computation and Cognition*, MIT Press, Cambridge, MA.

Quaife, A. (1988), 'Automated proofs of löb's theorem and gödel's two incompleteness theorems', *Journal of Automated Reasoning* **4**, 219–231.

Reiter, R. (2001), *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*, MIT Press, Cambridge, MA.

Rinella, K., Bringsjord, S. & Yang, Y. (2001), Efficacious logic instruction: People are not irremediably poor deductive reasoners, *in* J. D. Moore & K. Stenning, eds, 'Proceedings of the Twenty-Third Annual Conference of the Cognitive Science Society', Lawrence Erlbaum Associates, Mahwah, NJ, pp. 851–856.

Rips, L. (1994), *The Psychology of Proof*, MIT Press, Cambridge, MA.

Rosenbloom, P., Laird, J. & Newell, A., eds (1993), *The Soar Papers: Research on Integrated Intelligence*, MIT Press, Cambridge, MA.

Russell, S. & Norvig, P. (2002), *Artificial Intelligence: A Modern Approach*, Prentice Hall, Upper Saddle River, NJ.

Shankar, N. (1994), *Metamathematics, Machines, and Gödel's Proof*, Cambridge University Press, Cambridge, UK.

Shapiro, S. (1992), *Common Lisp: An Interactive Approach*, W. H. Freeman, New York, NY.

Shapiro, S. & Rapaport, W. (1987), SNePS considered as a fully intensional propositional semantic network, *in* N. Cercone & G. McCalla, eds, 'The Knowledge Frontier: Essays in the Representation of Knowledge', Springer-Verlag, New York, NY, pp. 262–315.

Sieg, W. & Field, C. (2005), 'Automated search for gödel's proofs', *Annals of Pure and Applied Logic* **133**, 319–338.

Siegelmann, H. (1995), 'Computation beyond the Turing limit', *Science* **268**, 545–548.

Siegelmann, H. & Sontag, E. (1994), 'Analog computation via neural nets', *Theoretical Computer Science* **131**, 331–360.

Siegelmann, H. T. (1999), *Neural Networks and Analog Computation: Beyond the Turing Limit*, Birkhäuser, Boston, MA.

Skyrms, B. (1999), *Choice and Chance : An Introduction to Inductive Logic*, Wadsworth.

Sloman, S. (1998), 'Category inference is not a tree: The myth of inheritance hierarchies', *Cognitive Psychology* **35**, 1–33.

Stanovich, K. E. & West, R. F. (2000), 'Individual differences in reasoning: Implications for the rationality debate', *Behavioral and Brain Sciences* **23**(5), 645–665.

Steele, G. (1984), *Common LISP, Second Edition: The Language*, Digital Press, Woburn, MA.

Stillings, N., Weisler, S., Chase, C., Feinstein, M., Garfield, J. & Rissland, E. (1995), *Cognitive Science*, MIT Press, Cambridge, MA.

Sun, R. (1995), 'Robust reasoning: Integrating rule-based and similarity-based reasoning', *Artificial Intelligence* **75**, 241–295.

Sun, R. (2001), *Duality of the Mind*, Lawrence Erlbaum Associates, Mahwah, NJ.

Sun, R. & Zhang, X. (2006), 'Accounting for a variety of reasoning data within a cognitive architecture', *Journal of Experimental and Theoretical Artificial Intelligence* **18**(2), 157–168.

Turing, A. (1950), 'Computing machinery and intelligence', *Mind* **LIX (59)**(236), 433–460.

Voronkov, A. (1995), 'The anatomy of vampire: Implementing bottom-up procedures with code trees', *Journal of Automated Reasoning* **15**(2).

Wason, P. (1966), Reasoning, *in* 'New Horizons in Psychology', Penguin, Hammondsworth, UK.

Wigner, E. (1960), The unreasonable effectiveness of mathematics in the natural sciences, *in* 'Communications in Pure and Applied Mathematics', Vol. 13, John Wiley and Sons, New York, NY, pp. 1–14.

Wolfram, S. (2002), *A New Kind of Science*, Wolfram Media.

Wos, L. (1996), *The Automation of Reasoning: An Experimenter's Notebook with* OTTER *Tutorial*, Academic Press, San Diego, CA.

Wos, L., Overbeek, R., e. Lusk & Boyle, J. (1992), *Automated Reasoning: Introduction and Applications*, McGraw Hill, New York, NY.

Yang, Y., Braine, M. & O'Brien, D. (1998), Some empirical justification of one predicate-logic model, *in* M. Braine & D. O'Brien, eds, 'Mental Logic', Lawrence Erlbaum Associates, Mahwah, NJ, pp. 333–365.

Yang, Y. & Bringsjord, S. (2001), Mental metalogic: A new paradigm for psychology of reasoning, *in* 'Proceedings of the Third International Conference on Cognitive Science (ICCS 2001)', Press of the University of Science and Technology of China, Hefei, China, pp. 199–204.

Yang, Y. & Bringsjord, S. (2003), 'Newell's program, like Hilbert's, is dead; let's move on', *Behavioral and Brain Sciences* **26**(5), 627.

Yang, Y. & Bringsjord, S. (2006), 'The mental possible worlds mechanism and the lobster problem: an analysis of a complex gre logical reasoning task', *Journal of Experimental and Theoretical Artificial Intelligence* **18**(2), 157–168.

Yang, Y. & Bringsjord, S. (forthcoming), *Mental Metalogic: A New, Unifying Theory of Human and Machine Reasoning*, Erlbaum, Mahway, NJ.