ANALOGICAL CONSTRUCTIVISM: THE EMERGENCE OF REASONING THROUGH ANALOGY AND ACTION SCHEMAS

By

John Licato

A Dissertation Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: COMPUTER SCIENCE

Examining Committee:

Selmer Bringsjord, Dissertation Adviser

Ron Sun, Member

Sergei Nirenburg, Member

John E. Hummel, Member

Paul Bello, Member

Rensselaer Polytechnic Institute
Troy, New York

April 2015
(For Graduation May 2015)

CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENT

My first acknowledgments should go to my family and friends back home in Hawai‘i. My parents, Michael and Remedios Licato, were willing to put up with my lack of direction in my “entrepreneurial” years, my suspensions from both the University of Hawai‘i at Mānoa and Leeward Community College because I never turned in homework, and my long path towards finally accepting that maybe I actually need to work to get what I want.

At RPI, I was lucky enough to have formed friendships with people who got me through graduate school, along with connections to interesting people and researchers who I'll be seeking to continue working with for the rest of my professional career. Many of these people are on my Ph.D. committee, and so to my committee members I thank you for the encouragement and the freedom you've given me to explore what must have seemed (and probably still seem) like unusual ideas. Believe me, they'll only get weirder from here on out.

In particular, I am eternally grateful to the guidance of Selmer Bringsjord, to whom I will always be indebted for whatever success I might later find in my career.

Thanks to the programs which funded my graduate research: The Air Force Office of Scientific Research (including the guidance of James Lawton, Jun Zhang, and Jay Myung), and the Office of Naval Research (under the guidance of Paul Bello).

And most importantly, if events in my life hadn't transpired as they have, I might not have met my fiancée Kaitlin Gallup. I simply could not have completed this work without her, and I like her so much I'm taking her with me. Here's to our future life together.

ABSTRACT

The ability to reason analogically is a central marker of human-level cognition. Analogy involves mapping, reorganizing, and creating *structural knowledge*, a particular type of cognitive construct commonly understood as residing purely within the domain of declarative knowledge. Yet existing computational models of analogy struggle to show human-level performance on any data sets not manually constructed for the purposes of demonstration, a problem referred to as the *tailorability concern*. Solving the tailorability concern may require more investigation into the nature of cognitive structures, defined as those elements in mental representation which are referred to whenever contemporary literature on analogy discusses "structured" knowledge.

I propose to develop the theory of Analogical Constructivism. This theory builds on Piaget's constructivist epistemology, first refining its concepts by clarifying the modifications Piaget himself made in his later, less-discussed works. I reconcile Piaget's assertion that meaning is, first and foremost, rooted in the action schemas that the agent is both born with and develops throughout life, with an account of cognitive structure, concluding that cognitive structure is inseparable from action-centered/procedural knowledge.

After a defense of the claim that cognitive structure cannot exist apart from actions (a claim which I refer to as "No-semantically-empty-structure"), I introduce PAGI World, a simulation environment rich enough in possible actions to foster the growth of artificial agents capable of producing their own cognitive structures. I conclude with a brief demonstration of an agent in PAGI World, and discuss future work.

CHAPTER 1
Late Piaget, Analogy, and Summary

1.1 Introduction

In this dissertation, I will introduce, defend, and demonstrate what I call **Analogical Constructivism**[1], which is both a new theory of cognitive development and an approach to Artificial Intelligence. Analogical Constructivism (referred to hereafter as $\mathcal{AC}$) is essentially the unification of analogy and Late Piagetian Theory (**LPT**). LPT is a term I use to distinguish Piaget's later works (ranging from around 1965 to his death in 1980). This period can be characterized by the maturation of Piaget's constructivist ideas, and the elaboration of what he called **abstractions**. Though Piaget's earlier work on developmental stages and schema mechanisms (assimilation and accommodation) are very well known in developmental psychology and childhood education, contemporary discussion on his later thought is comparatively minor.

There are several tasks this dissertation achieves:

- Introduce $\mathcal{AC}$;
- Refine the understanding of structured knowledge used in computational cognitive models and computer science in general;
- Describe a new simulation environment for developing such structured knowledge; and
- Present a simulation created in this environment which uses $\mathcal{AC}$.

The layout of this dissertation is as follows: This chapter will summarize some of the most important concepts characteristic of LPT. This will be followed by an

[1] In this dissertation, for clarity, I will try to adopt the following convention. Foreign terms and text to be emphasized will be in *italics*. New terms, when first introduced, will be in **bold**. Properties or conceptual structures, particularly those expressed using English words, will be in `teletype font`. Formulae and corollaries will be either in greek letters or, for hypotheses, the lowercase italicized h. Domains, sets, lists, and groups will be in bold, capital letters (e.g. $\mathbf{S}$). Finally, objects, and other symbols, will be italicized single letters.

overview of the relevant literature on analogy, the second major component of $\mathcal{AC}$. Chapter 2 will develop $\mathcal{AC}$'s understanding of structure, and close by stating the central positions that $\mathcal{AC}$ takes. With those preliminaries out of the way, I then summarize work I have done up to this point in ADR and Analogy (Chapter 3). Chapter 4 introduces PAGI World, a simulation environment capable of developing an artificial agent based on the principles of $\mathcal{AC}$. Finally, Chapter 5 will present a short simulation in PAGI World, serving as preliminary work to launch a new research program.

1.2 Late Piagetian Theory (LPT)

LPT refers to a highly productive period of Jean Piaget's life which produced work that, compared to his earlier work, is not afforded as much attention or discussion in the psychological literature. However, Piaget arguably wrote some of his most important work in this period, as he performed several important tasks: he addressed some of the contemporary criticism of his earlier work by more precisely defining and providing evidence for central concepts like equilibration; he introduced and developed the concepts of reflective and empirical abstraction (a concept which is central to this dissertation); and he even completely changed his position in some areas, by espousing viewpoints that directly contradicted previously-held viewpoints for which he is still criticized today.

So why did LPT not enjoy the acclaim given to Piaget's earlier work? We might briefly identify several possible reasons here. The first is that as early as 1970, with the publication of Juan Pascual-Leone's work (Pascual-Leone, 1970), we saw the emergence of the so-called "neo-piagetians," whose criticisms of Piagetian theory (as they understood it) shifted focus away from Piaget himself (Case, 1992; Morra, Gobbo, Marini, & Sheese, 2008). Another reason is the late translation of his work from the LPT years. Piaget never was fully comfortable writing in English (he says this several times in Piattelli-Palmarini (1980)), and much of his work, to this day, remains only available in his native French. Consider, for example, *Studies in Reflecting Abstraction*, which was only translated into English in 2001 (Piaget, Montangero, & Billeter, 2001). By comparison, the last major effort to implement

Piagetian thought into a computational simulation was probably Drescher (1991). The few translations of LPT work into English that do exist contain confusions that result from Piaget's use of terms that did not have direct equivalents in English. The results of this can be seen in the inconsistent English translations, between different authors, of the different types of reflective abstractions, which I describe shortly.

Finally, Piaget tried to reach out and connect his work to the emerging school of French Structuralism (e.g. the work of Claude Lévi-Strauss and Jacques Lacan) which was gaining popularity at the time, being applied to fields as diverse as literary analysis, linguistics, and sociology (Piaget, 1970; Turner, 1973; Solo, 1975). It is entirely possible that the failure of the English-speaking world to adopt French Structuralism (due in no small part to the work of "post-structuralists" like Jacques Derrida) led to a throwing out of the baby with the bathwater. Fortunately, a new wave of investigations into structure (e.g. the limits of logic-based AI, which intimately relies on structured representations (Bringsjord, 2008b)), the notion of structure used by literature in analogical reasoning (Hummel & Holyoak, 1997), and the nature of structure as it exists in the world, perhaps best exemplified by Luciano Floridi's Philosophy of Information project[2] may provide a renewed focus on the nature of structures themselves, both as they exist in the world and in how cognitive systems represent them. As a result, it may be time to re-assess the insights LPT gives us into the nature of cognitive structures, and $\mathcal{AC}$ proposes to do just that. We will return to the topic of structures in Chapter 2.

In any case, LPT introduces and develops several important concepts which we will now summarize: abstractions, actions as meaning, and a revision of the logicism with which Piaget was previously associated.

1.2.1 Abstractions

Although Piaget's better-known works already offer the contrasting mechanisms of accommodation and assimilation to explain the driving forces behind

[2] Floridi's work proposes a construct*ion*ist rather than a construct*iv*ist approach (Floridi, 2011b). The difference is that whereas Piaget's constructivism tends to focus on describing the emergence of cognitive structures, constructionism shifts the focus to a prescriptive one which tells us what sorts of cognitive structures *should* be developed. (Floridi, 2011a)

schema development in the child, LPT is distinguished from his earlier work by an increased focus on specific types of what Piaget called **abstractions**, a focus that would become the basis of his theory in his final years (Lourenço & Machado, 1996). The abstraction processes, which allow complex knowledge structures to form from simpler ones, come in at least three types:

- **Empirical (or Simple) Abstraction**: Abstraction which comes from perceived objects. Consists of deriving the common characteristics from a class of objects. Because in constructivism all perception is done through the schemas the subject currently has available (as opposed to direct perception of properties), Dubinsky (1991) interprets empirical abstraction as referring to experiences which "appear to the subject to be external. The knowledge of these properties is, however, internal and is the result of constructions made internally by the subject" (Dubinsky, 1991, p. 97).

- **Reflective (or Reflexive[3]) Abstraction**: Abstraction which starts from actions and operations. Consists in "deriving from a system of actions or operations at a lower level, certain characteristics whose reflection (in the quasi-physical sense of the term) upon actions or operations of a higher level it guarantees; for it is only possible to be conscious of the processes of an earlier construction through a reconstruction on a new plane [...] In short, reflective abstraction proceeds by reconstructions which transcend, whilst integrating, previous constructions" (Beth & Piaget, 1966, pp. 188-189). This is the process which creates new (as in, new for a reasoner) logico-mathematical constructions (Beth & Piaget, 1966, p. 205)(Dubinsky, 1991). Even the sensorimotor structures are derived "from more elementary structures by a process analogous to reflective abstraction" (Beth & Piaget, 1966, p. 204), which raises the question of where the most fundamental structures come from in the first place (this is addressed in the next section on action schemas). Piaget would later say that reflective abstraction was another way of describing equilibration (Montangero & Maurice-Naville, 1997, p. 63).

[3] Both 'reflective' and 'reflexive' are in use due to an inconsistency in English translations (Gallagher & Reid, 2002).

- **Pseudo-empirical Abstraction**: Can be considered a hybrid mix between empirical and reflective abstractions.

Abstraction dominates LPT, and is perhaps one of the most misunderstood Piagetian concepts (partially responsible, no doubt, for the fact that many of Piaget's writings still remain untranslated). Yet these studies of abstractions offer us insight into the particulars of conceptual construction—a useful feature indeed for the researcher interested in modeling such constructive processes computationally.

The primary difference between the three types of abstractions listed above seem to be the sources of their inputs. Empirical abstraction operates on the products of direct perception, whereas reflective abstraction works with structures in memory, ultimately producing new actions, coordinations, and objects onto "higher levels" (Dubinsky, 1991) (to make use of language, common in Piagetian writing, which assumes a sort of hierarchy with simple, empirical structures on the bottom, and highly-abstracted, logico-mathematical constructs on the top). The types of abstractions are supposed to interact in a way that allows for the products of one to feed into another.

Let us take the example of a child who sees rocks of different sizes and decides to place them in order according to size. Empirical abstraction is the process of perceiving the rocks, recognizing them as rocks, and recognizing that they have the property `size` is a perceptual process which draws on conceptual structures `rock` and `size`, themselves products of reflective abstraction. When the child actually interacts with the rocks and observes that her actions transformed what she is seeing is pseudoempirical abstraction. Finally, if the child should recall this experience later and decide that the entire event was similar to a time she did the same thing with differently-sized sticks she found, she would have performed a reflective abstraction.

Reflective abstraction can further be divided into two components:

- **Projection (*Réfléchissement*)** - The projection of structures from one level onto a higher level (Gallagher & Reid, 2002); the process of an activity or mental operation (which is not a static combination of sensory elements) which was developed on one level being abstracted and applied on a higher level

(Glasersfeld, 1991), allowing the projected coordination to be understood explicitly (Campbell, 2009).

- **Reflection (*Réflexion*)**[4] - The cognitive reorganization or reconstruction of what was abstracted or transferred from lower levels (Gallagher & Reid, 2002; Glasersfeld, 1991); the integration of the projected coordination with other structures at the higher level (Campbell, 2009).

In some sources there are other types as well: reflect**ed** abstraction (*abstraction réfléchie*) (reflecting abstraction of the second order) and metareflection/reflective thinking (*métaréflexion/pensée réflexive*) (Piaget et al., 2001; Campbell, 2009). These are higher-order products of reflective abstraction, and we will leave them out of this discussion.

Our first takeaway from LPT, then, is this list of abstractions, which together elaborate, coordinate, and make use of cognitive structures, and a high-level view of how they coordinate to do so. Each type of abstraction seems to build off of the products of the others, which raises the question: What are the initial cognitive structures? Piaget offered the following answer:

> [T]here are certain givens from which the construction of logical structures takes off, but these "data" are not primordial in any absolute sense, being merely the starting point for our analysis, nor do they "contain" what is, in the course of construction, "derived" from and "based" on them. We called these initial structures behind which we cannot go "general coordinations of actions," meaning to refer to the connections that are common to all sensori-motor coordinations (Piaget, 1970, p. 63).

The connection between "general coordinations of actions" and knowledge structures, however, is a complex one, and another central topic of LPT. The resulting implications for the nature of cognitive structures will therefore be important to understand for the purposes of this dissertation.

[4] As if to add to the confusion in translations: Gallagher (2002) refers to *réfléchissement* as 'projection,' von Glasersfeld (1991) as 'reflection.' However, Gallagher refers to *réflexion* as 'reflection,' while von Glasersfeld calls it 'reflexion!' We will refer to these terms in their original French, in italics, to avoid confusion.

1.2.2 Actions as Meaning

Piaget's final book, *Toward a Logic of Meanings*, attempted to create a 'logic of actions,' an intensional logic which would describe operations beginning at the sensorimotor level and ultimately lead to the creation of logico-mathematical structures (Piaget, Garcia, & Davidson, 1991). What resulted, however, was a topic of great interest to the present discussion. It was argued, in no unclear terms, that the nature of meaning is rooted in action schemas and the action implications between them. The meaning of an object amounts to "what can be done" with it, and therefore to perceive an object is to assimilate that object to an action scheme. As for actions themselves, their meanings are defined by "what they lead to," or what transformation(s) they produce in whatever object or situations they are applied to. In other words, the meanings of actions are what roles they play in action schemas.

Such a view of semantics seems to parallel Wittgenstein's meaning-as-use semantics and Floridi's action-based semantics (Floridi, 2011b), the latter which I will discuss in Section 2.2.1. LPT also has an interesting connection to Whitehead's event-based ontology (Whitehead, 1919), in that they view objects as stable constructions rather than epistemological primitives—for Piaget, objects are constructed from the roles in actions, and for Whitehead they are created with events.

Piaget grounds all meaning in the action schemas that are constructed by a combination of reflective, empirical, and pseudo-empirical abstractions. In (Piaget et al., 1991) he argues that there must be *action implications* between actions, and discovering a logic that describes their behavior is a primary goal of the book. One of the major discoveries here is that the action implications seem to behave in a way that captures all 16 of the binary connectives of propositional calculus. This is the case at the earliest stages of sensorimotor development, and is present even in the innate action schemas:

> [A]ny observable is always linked to an interpretation which necessarily involves not only meanings, but also inferential links between these meanings or to previous meanings. [...] Thus, even the most elementary scheme, the preformed sucking reflex, already involves implications (between displacements and successes or failures, i.e., when the newborn

> must change its position in order to adjust its ill-positioned mouth to the nipple) (Piaget et al., 1991, p. 8)

I will not be so bold as to claim that all of meaning can be explained by a reduction to actions (we will return to this topic later in the discussion of Floridi's Action-based Semantics). But the idea that actions, or action schemas, can be a foundation for both a sensorimotor-level and a higher-level computational model of cognition is intriguing. Turning our attention briefly to computational implementations, it is of no surprise that perhaps the field of robotics—which deals directly with actions in the world—should have what is perhaps the best-developed literature on implementations of action schemas. For just a few of these approaches, see (Platt, Grupen, & Fagg, 2006; Cohen, Chang, & Morrison, 2007; Aksoy et al., 2013). However, a survey of the field makes it obvious that research projects in robotics typically stop at action-centered representations, rarely using processes like reflective abstraction to modify and build upon their action schemas, ultimately producing the sort of structures capable of elementary logico-mathematical reasoning. Furthermore, the rich description of action implications described by Piaget and Garcia (1991) is not nearly satisfied; consider, e.g., how Piaget and Garcia claim action implications should be rich enough to reflect the propositional-calculus connectives. Without a full account of action implications, it is difficult to see how something like contradiction between action implications can be represented, and without contradictions, schema development is severely handicapped. In Piaget's own words: "[The] progressive overtaking of contradictions, which constitutes the formative process of differentiations as well as coordinations, is fundamental when it comes to relations between operations and causality. To raise contradictions is, in effect, to construct a new operational structure" (Piaget & Garcia, 1974).

To their credit, some of the work from the robotics field in implementing action schemas have built on Ed Dubinsky's APOS theory (Dubinsky & Mcdonald, 2002). Dubinsky is one of the few experts on LPT who takes Piaget's views on meaning seriously: he builds primarily on Piaget and Garcia (1991) to create a model of how actions become processes, which in turn become objects, which then combine with groups of other objects and actions to form schemas (hence the initialism APOS).

APOS theory has primarily been applied to education (Dubinsky & Mcdonald, 2002; Arnon et al., 2013).

The astute reader may recognize the applicability of the distinction between procedural (or action-centered) thought and declarative (or non-action-centered) thought. LPT's recommendations, then, might be interpreted as saying the *structures* used to represent declarative knowledge arise from a complex interaction between existing structures, some of which may be considered declarative, others procedural. Ultimately, however, the primary structures from which all others are built are implicit and action-centered, to use the language of Sun (2002) . One might even go a step further, and declare that to successfully model the development Piaget and colleagues observed, procedural and declarative knowledge must be so tightly integrated that aside from the innate structures, no structure of either knowledge type arises without a complex interaction with knowledge of the other type.

This point — that cognitive structures and action schemas increase their complexity by building off of each other — will be restated later as one of the three central positions of Analogical Constructivism (albeit in a slightly weakened form).

If the reader is at this point concerned that I have made a perfunctory and uncritical restatement of LPT, it should help to clarify my purpose for selecting the concepts described thus far: My intent is to search LPT for concepts, terminology, ideas, and empirical observations, all of which LPT has in abundance, and to explore whether they can form the basis of a new approach to AI. Although LPT distinguishes itself from pre-LPT thought on more points than I mention here (see, e.g. (Lourenço & Machado, 1996; Montangero & Maurice-Naville, 1997)), the points I discussed are directly relevant to the simulation which this dissertation will present. The next step is to see how LPT can hold up, and perhaps be refined by, more recent literature.

1.2.3 Action Implications

Piaget observed that young children (9-10 months old) who were given boxes of varying sizes so that some could be nested in the others, behaved in an interesting way. When they placed one box inside of a larger one, they would first put the smaller box in their mouth. Piaget suspected that this was evidence of the children starting with the established scheme of placing-object-in-mouth, and through reflective abstraction constructing a new container-content scheme (Piaget et al., 1991). Furthermore, the ability to place an object in a container was accompanied with the ability to take the object out again, constituting a pair of reversible actions. More generally, Piaget referred to the relationship between such paired actions as a type of **action implication**, e.g. "action x implies the possibility of the reverse action" (Piaget et al., 1991, p.5).

Action implications (and the closely related idea of meaning implications, of which action implications are a subset) are among the most interesting ideas introduced by LPT, primarily because they represent an abrupt change in Piagetian thought. Piaget and collaborators found that relations between actions, or the possible ways that children combined pairs of actions, formed in a way isomorphic to the sixteen binary operations of propositional calculus, despite previously believing that the operations of propositional logic did not appear until age 11-12 (Piaget et al., 1991, p.6).

I will define action implications following (Piaget et al., 1991, p.vii-viii). Given two actions a_1, a_2 of an agent, Piaget defines the meanings of those actions as "what they lead to" in the mind of the agent (Piaget et al., 1991, p.119). An action implication, then, is a relationship between the *meanings* of a_1 and a_2 such that the first meaning "leads to" the second (Piaget et al., 2001, p.96). For example, reaching an object implies getting closer to it. Getting closer to an object might imply either grabbing it and pulling it closer, or walking towards it. These are examples of action inferences between actions firmly dependent on physical objects. But action inferences can apply to operations over purely mental constructs as well, e.g. the action of multiplying a number by two can be "reversed" by dividing by two.[5]

Action implications, and their parent class meaning implications, are a fundamental building block of LPT's take on constructivism. Along with reflective abstraction, Piaget and colleagues believed, meaning implications give rise to the

[5] I do not discuss here the concept of reversible actions, which Piaget spent much time trying to understand in his work prior to LPT (Inhelder & Piaget, 1958).

formal operations, and therefore it is plausible to assume that simulating the human ability to reason logically might benefit from careful attention to modeling the processes from which the ability emerges.

1.3 Analogy

A quote from Piaget on reflective abstraction suggests a connection to what we today might recognize as analogical transfer:

> [Reflective abstraction] can be observed at all stages: from the sensory-motor levels on, the infant is able, in order to solve a new problem, to borrow certain coordinations from already constructed structures and to reorganize them in function of new givens (Piaget et al., 2001, p. 6)

Compare this to the modern understanding of analogy, where analogy is typically used to solve problems that are new for the reasoner, that it is a process that involves the borrowing of already-obtained knowledge structures, and that it can often involve the reorganization of the retrieved knowledge structures in order to improve the match quality (Gentner & Forbus, 2011).

Analogy is the cognitive process which operates over structured representations in a *source* and *target* domain, making analogical reasoning possible (Gentner, 1983). Analogy *simpliciter* may be better described as consisting of four different processes, in a division roughly following (Gentner & Forbus, 2011):

- Analogical Retrieval - Given a target domain **T** and an optional set of pragmatic constraints, retrieval searches a knowledge-base for relevant source cases which it may retrieve as a whole or construct from semantically related pieces (depending on the model used). It is widely believed (Gentner & Forbus, 2011) that whereas analogical matching relies on structural knowledge, the retrieval step relies on "surface similarity," which is not considered to be structural (this, however, is a point of contention to be addressed later) (Gentner & Toupin, 1986; Holyoak & Koh, 1987; Thagard, Holyoak, Nelson, & Gochfeld, 1993).

- Analogical Mapping - Given a source domain **S** and a target domain **T**, a mapping M is found between the elements of **S** and **T**. The mapping may be subject to constraints such as systematicity, the one-to-one constraint (Gentner, 1983), pragmatic constraints, or any of the others identified by multiconstraint theory (Holyoak & Thagard, 1989).

- Analogical Inference and Generalization - This might be considered the ultimate goal of analogical reasoning. Given **S**, **T**, and M from the mapping step, new conceptual structures are created and hypothesized to be a part of the target domain **T** (analogical inference), or to serve as a generalization of both **S** and **T** (analogical generalization).

- Re-representation - Perhaps the most poorly understood of the four analogical processes, re-representation involves a reorganization of the conceptual structures in the source and/or target domains to better satisfy one of the constraints in the mapping step, or to produce better results in the inference/generalization step.

The analogical mapping step involves, at the very least, making comparisons between *structured* representations, but the definition of structured representations used by computational and cognitive models of analogy is one that, as shall be argued in Chapter 2, is unsatisfactory. In particular, the weak understanding of what it means for a representaton to be structured may be to blame for the lack of models that can perform the re-representation step (at least to a level of proficiency matching the abilities manifested even by small children).

1.3.1 Analogy as the "Core of Cognition"

When construed as a cognitive process that compares structured representations and creates new ones on the basis of those comparisons, it is easy to see why analogy might be considered a central process to thought (assuming of course, that cognitive thought is fundamentally structured, whatever that means; see Chapter 2). The extreme position, held by Douglas Hofstadter, takes analogy to be "the very blue that fills the whole sky of cognition" (Hofstadter, 2001). According to

Hofstadter's view, all of the higher-level cognitive features typically associated with the neurobiologically normal adult human mind are fundamentally made possible by analogy. To cite a few examples: Categorization is an analogical process because it requires the adaptation, through a mapping process, of an existing mental entity to a set of incoming stimuli (Hofstadter, 2001; Hofstadter & Sander, 2013). This means that recognition of, say, everyday visual inputs ("that looks like a tree") can be described as a form of analogy.

Hofstadter goes further to criticize the source-target paradigm as it is used in many experiments, arguing that the vast body of literature which rely too heavily on it can only produce conclusions of limited generality. In the real world outside of the experiment, we draw on source knowledge "deeply rooted in our experiences over a lifetime, and this knowledge [...] has been generalized over time, allowing it to be carried over fluidly to all sorts of new situations. It is very rare that in real life we rely on an analogy to a situation with which we are barely familiar at all" (Hofstadter & Sander, 2013). The use of "surface features" identified by participants in analogy experiments shows no more than "when people learn something superficially, they wind up making superficial analogies to it" (Hofstadter & Sander, 2013). Instead, Hofstadter seems to suggest that models of analogical retrieval (and perhaps matching) should be influenced by not surface similarity, but rather by two factors: the simulated agent's *familiarity* with the source analog, and the source analog's level of conceptual development (relative to the simulated agent). These two factors may be virtually equivalent in practice, but for an artificial simulation they each suggest different methods of implementation. Familiarity level may be captured by a simple scalar value attached to a source domain or schema, whereas the level of conceptual development may be an emergent property of a cognitive structure falling out of the complexity of the relations (Hofstadter would call these *chunks*) used, the height of the overall structure, the flexibility of the structure to be re-represented, and so on.

The ability to perform *argument by analogy*, which conforms to a particular argument form (e.g. (Bartha, 2010; Bringsjord & Licato, 2015)), is trivially reliant on the four analogical processes listed above. But what about deductive reasoning?

There have been attempts to work in the other direction, by reducing arguments by analogy to a deductive schema, though none of them have gained widespread acceptance (Bartha, 2010). Reduction of deductive reasoning to arguments by analogy, on the other hand, is difficult because analogical arguments do not have the validity that distinguishes deduction; if from a source domain **S** and a target domain **T** we perform an argument by analogy to infer a formula ϕ about the target domain, we cannot say that ϕ is formally valid. At most, ϕ is *prima facie* plausible (Bartha, 2010).

Thus, reducing argument forms of analogy and deduction to one another is a difficult task, and it is not the concern of this dissertation. So we must refine the question to one that is more relevant: Are the four analogical processes sufficient to create a system that can correctly distinguish, carry out, and work with deductive arguments? The answer to this question is a tentative yes, and the demonstration-of-concept is in a paper I previously published (Licato, Sun, & Bringsjord, 2014) which will be discussed in Chapter 3. For now, let it suffice to state this hard-to-deny assertion: Both analogical and deductive reasoning cannot operate without structured representations. Analogy relies on structured representations almost by definition, and deduction, insofar that it involves the manipulation of formulae according to their adherence to certain syntactually defined criteria, simply cannot work without the structure which makes those syntactical forms possible in the first place.

1.3.2 Can Analogy Explain Abstraction? Acting in-accordance-with vs. reasoning-over

Let us assume that analogy is a general cognitive process that works on all levels dealing with structured knowledge, including the meta-levels on which we would assume that LPT's abstraction processes work. What, then, is the relationship between the abstraction processes and analogy? Here, I will put forth a brief argument that the abstraction processes, as described by LPT, can be understood as intimately related to analogy. More precisely, we will call a cognitive process **analogy-structure-sensitive** if the majority of its activity relies on the same structured representations which are used by analogy. (Note that the definition of analogy-structure-sensitive processes may encompass a wide variety of cognitive processes, but this is exactly the point: I want to show that the sort of structure used by analogy is central to human-level thought.)

Recall that there are at least four types of abstractions: empirical abstraction, which applies internal constructs to perceptions which appear to the subject to be external; *réfléchissement*, which creates new cognitive structures at a higher level that are reflections of structures on lower levels without transforming the lower-level structures; *réflexion*, which does roughly the same thing but additionally causes a re-organization or reconstruction of structures; and pseudo-empirical abstraction, which is a hybrid of the others. If it can be plausibly argued that the first three types of abstractions can be explained using the four analogical processes, then the task is complete (at least for the current work, whose goal is to plausibly simulate the abstractions).

Empirical abstraction is essentially the act of categorizing perceptual stimuli as it arrives. Let us assume that a child, her eyes directed towards a set of rocks on the sand of a particularly rocky area on the north side of Makapuʻu Beach Park, receives the low-level uninterpreted visual sensory data S. At some stage there needs to be a recruitment of knowledge structures and categories with which she is familiar, and these structures need to be compared with S to determine if there is an acceptable match. Now S does not have any apparent structure (if we agree with the approach of (Hofstadter & Sander, 2013)), since empirical abstraction first attaches structure to the sensory input, allowing those rocks to be recognized as such. So it seems disingenuous to say that analogy, while comparing structured representations, is directly and solely responsible for the *initial* level of structuring. On the other hand, empirical abstraction simply cannot operate without previous products of reflective abstraction that are themselves necessarily structured entities (Montangero & Maurice-Naville, 1997; Gallagher & Reid, 2002). Furthermore, once the initial layer of structuring is done, any subsequent structuring is then being performed between two structured components—the source domains and the newly structured input. At best, then, we might characterize empirical abstraction as a "pseudo-analogical" process, but it certainly can also be described as analogy-structure-sensitive.

Reflective abstraction, however, is more complicated. I will interpret it using a distinction often used in discussions of Wittgenstein, that of acting **in-accordance-with** versus **reasoning-over**. Given a cognitive structure S stored as an action schema (something that the agent can behave *in-accordance-with*), reflective abstraction allows for the creation of S_R, which is a reflection of S, that can be used as an object or substructure in operations by the agent in the future. S_R is not an exact duplicate of S, nor is it a simple generalization; rather, S_R is the reflecting process's interpretation of S. The reflecting process is finite, and it is itself limited to the tools provided to it at the time of reflection, namely, the existing conceptual constructs (themselves products of previous instances of reflective abstraction). This is perhaps why the reflective process does not produce an exact copy; it is for the same reasons that a man observing another cannot generate an explanation of the observed man's actions in terms of the low-level neurons firing; rather, he must produce an explanation that is high-level and conceptual (e.g., "The man is eating because he is hungry.").

This distinction seems consistent with the example of reflective abstraction given by Robert Campbell (translator of Piaget's *Studies in Reflecting Abstraction* and perhaps the leading expert in English on abstraction in LPT):

> Multiplication looks like repeated addition — yet children find it much harder than addition. According to Piaget's analysis, children have to be able to recognize how much they are adding each time. This is *empirical abstraction*; even the youngest children in [a study done by Piaget and colleagues] easily recognized the number of poker chips that they were adding to the row each time. To multiply successfully, however, Piaget maintains that children must also attend to the number of *times* that they add that amount. Only through *reflecting abstraction* can children understand how many times they added poker chips to one row or how many times the experimenter added chips to the other. The same goes for realizing that adding two and doing that three times has produced

> the same number as adding three and doing that twice. (Campbell, 2009, p. 153-154, emphases in original)

In this dissertation, behaving *in-accordance-with* vs. *reasoning-about* will be connected to using action-centered (procedural) vs. non-action-centered (declarative) knowledge.

A simple example is in order. If a child were to grab a marshmallow every time he sees one, we can conclude he has an action schema, a structure, which he can behave *in-accordance-with*. But if one were to tell him not to eat the marshmallow on the table, this requires a few things. First, he needs to transform his action schema from something he can behave *in-accordance-with* to something that he can *reason-about* (*réfléchissement*), and then he can further transform the result appropriately (*réflexion*). In such a case he would need to perform some sort of negation, in order to know how *not* to perform the action of eating the marshmallow, and then to make this negated action schema an intention of his. Quite a task to demand of a child!

Now to return to the current goal: Can reflective abstraction be understood as structure-sensitive processes? In the marshmallow example, we redescribed the two subtasks which comprise the reflective abstraction step: The *réfléchissement* step (reflect S, an *in-accordance-with* structure, into S_R, a *reasoning-about* structure); and the *réflexion* step (incorporate S_R into the existing *reasoning-about* structure-space). The key to $\mathcal{AC}$'s understanding of the *réfléchissement* step is this: The process of transforming an *in-accordance-with* structure to a *reasoning-about* structure is like an analogical inference where the source domain is a subset of action-centered (procedural) knowledge, and the target domain is a subset of non-action-centered (declarative) knowledge. When the child transforms his implicit action schema into the sort of thing that he can reason about, he is performing a comparison between two very different types of knowledge, resulting in the creation of new cognitive structure, but the knowledge that is compared is *structured* nonetheless.

The *réflexion* step incorporates S_R into the non-action-centered knowledge-base. Some of this incorporation falls quite nicely out of the analogical inference process: Because analogy constructs S_R out of concepts and structural elements that are already extant in the target domain, it already achieves a level of semantic integration. In that aspect, it is simply a side-effect of analogy processes. But clearly there is more to it than that. In the marshmallow example, further processing is done to transform S_R, to ensure that the resulting structure is compliant with the command that was given to the child ("Do not do S_R"). Some pragmatic motivation is seemingly unavoidable here. We must therefore have at least a basic theory of semantics upon which we can rest our conception of structures, and this task will ultimately be performed in Section 2.2.1.

1.4 Lessons to Take Away

This chapter brought up several important points:

- Most (if not all) cognitive structures are created through reflective abstraction.
- The complexity and richness of actions and cognitive structures can bootstrap each other.
- Analogical reasoning, which relies fundamentally on structure, is being increasingly regarded as a core process of cognition.
- The abstraction processes described by LPT, which are responsible for the elaboration of cognitive structures, can be redescribed as rooted in analogical-structure-sensitive processes (with a minimum of additional innate processes).

Stated in this way, an obvious theme emerges: Structure, particularly the aspect of structure analogy is sensitive to, is extremely important to the $\mathcal{AC}$ approach to AI. But *what is* structure, and how should it be modelled? The next chapter will address this topic in more detail.

CHAPTER 2
Structure and Actions

Chapter 1 argued that the two major components of $\mathcal{AC}$, LPT and analogy, are each inseparable from the concept of structure. This is not an inconsequential claim: the ability to recognize, operate on, behave *in-accordance-with*, and *reason-about* structures are virtually taken for granted in the modern computer and cognitive-science literature. The limits, strengths, advantages/disadvantages, and ultimately the nature of structures is rather insufficiently explored, to the detriment of theories and computational models that rely heavily on the concept.

As this is a dissertation for a degree in computer science and not philosophy, I refer only to *cognitive* structures, i.e., the sort of structure is that is used by analogy. This is in contrast to structures in the world (isofar as they exist), as described by varieties of Structural Realism (Floridi, 2011b; S. French, 2014). I will not be attempting to answer any questions about what is "real," or what things exist outside of human knowledge; I will only be talking about features of knowledge representation in human beings. Where it is clear, I will omit the word 'cognitive.' I may alternatively refer to cognitive structures as **constructions** if there is a need to emphasize their non-innateness.

In this chapter, I try to answer the following questions:

1. How should $\mathcal{AC}$ reconcile the conflicting understandings and ultimately define cognitive structure? While the analogy literature and LPT both happen to use the word 'structure,' what reason do we have to suppose that they are referring to the same thing?

2. How can we identify structure in humans and AI agents? It is one thing to posit the existence of mental constructs, but what should we expect of an AI agent with *truly* structured representations?

3. What does $\mathcal{AC}$'s definition mean for models of analogy, which primarily work with structure?

4. What is the relationship between structure and action?

A precise understanding of structure is of great interest to computer science as a whole, as the current extreme popularity of the Deep Learning paradigm (heralded by breakthroughs in (Hinton, Osindero, & Teh, 2006) and (Bengio, Lamblin, Popovici, & Larochelle, 2007)) is paving the way for a literature that refers to structure as the sort of thing neural networks can now learn and represent (e.g. see (Adams, Wallach, & Ghahramani, 2010; Saxe, McClelland, & Ganguli, 2013)). Such an implicit assumption may or may not be right. While it is entirely possible that the definition of structure used by most computer scientists is useful for describing (artificial) neural networks, the fact remains that its overlap or lack of overlap with definitions used in cognitive science and computational cognitive modeling is in dire need of clarification.

This chapter will attempt to face this challenge, as will be necessary to provide practical answers to the list of questions above. This will pave the way for a formal introduction of $\mathcal{AC}$'s central principles, closing out this chapter.

2.1 Structure in Models of Analogy

A natural starting point in the story of modern models of analogy is Gentner's (1983) paper introducing the Structure-Mapping Theory (SMT), whose central claims are the linchpin of almost all computational models of analogy today. Knowledge in SMT is represented using "propositional networks of nodes and predicates," where nodes would be used to represent localist concepts, and predicates represent either attributes (single-argument predicates) or relations (predicates with more than one argument). Second- and higher-order predicates are also possible, where a second-order predicate is one taking a proposition as an argument (e.g.: *CAUSE[COLLIDE(x,y), STRIKE(y,z)]* is a second order predicate *CAUSE* relating the propositions *COLLIDE(x,y)* and *STRIKE(y,z)*).

An important clarification of SMT's knowledge representation is that a given collection of objects and predicates (referred to as a domain or situation) is intended to "reflect the way people construe a situation" (Gentner, 1983, p.157), meaning that a domain might be the result of, for example, a perceptual process, or possibly a perception that has previously been restructured in some way. In a comparison between two domains, the type of predicates used determine whether the comparison is considered an analogy. If a large number of predicates (both attributes and relations) are mapped relative to the number of nonmapped predicates, the comparison is a **literal similarity**. If many relations but few to no attributes are (or in Gentner's words, "can be") mapped, then it is an **analogy**. Finally, if few relations but many attributes are mapped, then the comparison is a **mere-appearance** or **surface** match.

SMT holds that there are three constraints on the mapping process (the following descriptions following (Gentner & Forbus, 2011)). For a source domain $\mathbf{S}$, target domain $\mathbf{T}$, and mapping $M \subseteq \mathbf{S} \times \mathbf{T}$:

- Structural Consistency:
  - 1-to-1 constraint - M does not map any element in $\mathbf{S}$ or $\mathbf{T}$ to more than one element.
  - Parallel connectivity - For $(s, t) \in M$, if s and t are predicates, then their arguments must also be paired in M.
- Systematicity - Mappings are preferred if they put into correspondence larger systems of relations, in particular "those governed by higher order constraining relations" (Gentner & Forbus, 2011).
- Tiered Identicality - Mappings are preferred if they consistently put the same symbols into correspondence. E.g., if $(s_1, t_1) \in M$, and there exist $s_2 \in \mathbf{S}, t_2 \in \mathbf{T}$ where s_1, s_2 have the same symbol and so do t_1, t_2, then a mapping which paired (s_2, t_2) would be preferred.

Structure in SMT, therefore, is conveniently a property of the formal syntax of the representation: If something in a domain is a predicate, then it is part of the structure of that domain (as far as analogy is concerned).

2.1.1 Competing Views of Analogy

SMT, being a theory formalizing analogy to a certain amount, lends itself quite nicely to computational implementation. The first computational model to take advantage of SMT, the Structure-Mapping Engine (SME) (Falkenhainer, Forbus, & Gentner, 1989), has been a central figure in the computational analogy literature ever since its publication. But perhaps due to the restrictions and precise definitions that are at the core of SMT, it also prompted a body of criticism and competing theories.

The Multiconstraint Theory (MT) (Holyoak & Thagard, 1989), proposed three changes:

1. That the constraints identified by SMT be interpreted as mere preferences (rather than hard rules), referred to as the *isomorphism* constraint;

2. treat semantic similarity as a constraint on analogy separate from isomorphism; and

3. treat pragmatic centrality as a constraint on analogy, also separate from isomorphism.

The computational implementation of MT was named 'ACME' (Analogical Constraint Mapping Engine), and although the authors noted that "in many respects ACME can be characterized as an extension of SME" (Holyoak & Thagard, 1989, p.316), there were several key differences between the two systems, reflective of the differences between MT and SMT.

MT uses a definition of structure that is more or less the same as the one used in Gentner (1983). However, a note in Holyoak and Thagard (1989) tells us something rather interesting:

> This sense of "structural" [as used in Falkenhainer et al. (1986)] is to be distinguished from that used by Holyoak (1985) and Holyoak and Koh (1987), who defined "structural" properties as the goal-relevant aspects *within* a single analog. Structural properties in the latter sense will be termed "pragmatically central" or simply "important" properties in the present paper [...] Use of different senses of the term "structural" in the analogy literature has contributed to some theoretical misunderstandings (e.g. Gentner 1989; Holyoak 1985). (Holyoak & Thagard, 1989, Footnote 1)

What we see here is a move toward a definition of structure that is more in line with that used by SMT and SME, away from a definition that incorporates pragmatically central properties, e.g. considerations of the *action* that initiated the analogy in the first place. This move is parallel, but in a direction completely opposite to, the move $\mathcal{AC}$ will make later in this chapter.

Although MT might be considered a slight modification to SMT, Chalmers et al. (1992) and Hofstadter and Mitchell (1995) put forth more biting critiques of both SMT and SME. I will not here recap the entirety of the exchange, but will instead extract a few points relevant to the present discussion.[6] Perhaps the primary critique here is that analogy and high-level perception are either extremely close, or the same thing (Chalmers et al., 1992; Hofstadter & Mitchell, 1995; Hofstadter & Sander, 2013). If this is accepted as fact, then some important consequences follow:

- The structural mapping and perceptual processes are not temporally separable; rather, they interact in so close a manner that we might consider analogy-making to be a perceptual process.
- Categorization, insofar as it is a perceptual process (i.e., to categorize X as an instance of Y is to *see* X as Y), is therefore also temporally inseparable from structural mapping.
- An approach like that of SME is doomed to suffer from tailorability and a lack of flexibility.

According to Chalmers et al. (1992), the (high-level) perceptual processes giving us the highly formalizable representations used in SME are themselves the results of multiple iterations of structural mapping and restructuring. This is so important, that by the time we end up with the representations that SME maps together, the problem of finding an analogy is essentially solved: The answer is encoded into the representations.

Ultimately, Chalmers et al. (1992) warns that "the use of hand-coded, rigid representations will in the long run prove to be a dead end, and that flexible, context-

[6]See (Chalmers, French, & Hofstadter, 1992; Hofstadter & Mitchell, 1995; Forbus, Gentner, Markman, & Ferguson, 1998).

dependent, easily adaptable representations will be recognized as an essential part of any accurate model of cognition" (Chalmers et al., 1992). This worry was taken seriously by what I'll call the 'Copycat Group' of models that came from this line of thought: Copycat (Hofstadter, 1984; Mitchell, 1993; Hofstadter & Mitchell, 1995) Metacat (Marshall, 2006), Tabletop (R. M. French & Hofstadter, 1991), Musicat (Nichols, 2012), and so on. But it is not clear that research with completely fluid structures like in that family of models has not reached the same dead end Chalmers et al. (1992) warned would come from the "rigid representations" of other models.

In their response to criticisms from (Chalmers et al., 1992; Hofstadter & Mitchell, 1995), Forbus et al. (1998) argued that the metaphor between analogy and high-level perception was too vague as a technical proposal. Among other responses, they argue that the Copycat model is extremely inflexible, being domain-specific and lacking generality, and in many ways, SME is actually *more* flexible than Copycat. The debate seems to have settled into the background for now, with no clear winner.

The consensus so far (at least among the research groups thus far described) seems to be that cognitive structure is a feature of human knowledge representation used heavily by analogical processes, and that structure is either identical to, or emerges from, the network of relations between concepts. What can the clash of ideas between the Copycat Group, MT, and SMT further teach us about cognitive structure? Let us examine the claims of Chalmers et al. (1992) from a LPT perspective.

I find at least one claim made by the Copycat Group to be quite convincing: A source and target analog already rich with high-level concepts (represented as localist chunks) virtually obviates the need for a high-level perceptual process. The perceptual processes, then, whatever they are, are such that their result is a structured representation which can then be operated on by analogy. The perceptual and categorization processes, when they are initiated by an observation actively being made by an agent, make use of concepts and *tools for structuring*. This makes the perceptual and categorization processes look very much like empirical abstraction. But in the view of Chalmers et al. (1992), this process is not a one-time executor which takes an input once and spits out a fully formed conceptual structure on par with the tailored examples typically provided to SME. Rather, the process is incremental: Empirical abstraction partially structures the input, analogy comes in and does *something*, and these two steps repeat until a full perceptual structure emerges.

The three approaches taken by the Copycat Group, MT, and SMT might be placed on a sort of continuum. At one end, we have SMT with its emphasis on the structural mapping taking place over "rigid" syntactual constructions, already assumed to exist in more-or-less a final form at the time of mapping. At the other end is the Copycat Group, for whom the structures do not exist in a single form that is used in all cases; rather, *every time a structure is needed* it is re-created from some partially structured or unstructured representations. MT is in the middle only by virtue of its having both structured and unstructured knowledge, and its acknowledgement that there are some nonstructural features of interest to analogy-making.

$\mathcal{AC}$ takes a position in the middle of the two extremes, and in that sense, it is closest to MT. But $\mathcal{AC}$, and this dissertation, are also attempting to argue for Copycat's emphasis on how structures are created both initially (when they are first formed in the mind of the cognizer), and dynamically (at "run-time", or the way that they are created in response to momentary needs by the cognizer), in order to preserve the flexibility of human cognitive structures. $\mathcal{AC}$ differs from the Copycat Group in that it claims the answer lies in a deep connection between cognitive structure and *action*. My next step is to explore a body of literature that tries to understand structure at a neurobiological level.

2.1.2 The Structure of Structures: Cognitive Structure at a Neurobiological Level

Largely based on MT, Hummel and Holyoak's LISA (Learning and Inference with Schemas and Analogies) model (Hummel & Holyoak, 1997, 2003a, 2003b) attempts to show how the analogical processes of retrieval, mapping, and inference can be explained as a natural side-effect of neurobiological activity. Achieving such a task, as it turns out, requires breaking apart the predicate-object dichotomy characterizing SMT, which they achieve by arguing for the requirement of **dynamic, independent binding**.

Systematicity and compositionality (systematicity here is not to be confused with the systematicity constraint introduced by Gentner (1983)), two well-known properties of structural representations (Fodor, 1980; Fodor & Pylyshyn, 1988), are both made possible by the ability to represent a predicate's roles and to bind them to fillers (typically objects, but possibly other propositions) in such a way that is both **dynamic** (bindings of variables to role fillers must be easy to frequently create and destroy) and **independent** (the binding must be independent of the entities it binds, so that its destruction or creation does not impede the ability for the entities to bind with others) (Hummel & Biederman, 1992; Hummel & Holyoak, 1997; Holyoak & Hummel, 2000). This is contrasted with **static binding**, also called **conjunctive coding**, a representation style that fixes the binding with the identity of the unit representing it. For example, consider a representation of the concept RED CAT. In conjunctive coding, there would be a single unit used to represent RED CAT, and an entirely separate unit used for BLUE CAT. In dynamic binding, properties are represented with separate units than the objects they are bound to, so that RED and CAT might be separate units, and the representation for RED CAT consists of the two separate units with a binding between them. Furthermore, the requirement that the binding is independent ensures that if the CAT unit had a binding to RED and another binding to BLUE, the bindings can exist simultaneously without interfering with each other. Static binding also causes a loss of similarity structure (Hummel & Biederman, 1992; Hummel, 2001), making it impractical for any system hoping to capture the sort of structure necessary for analogical reasoning. LISA, however, uses static binding in long-term memory (Hummel & Holyoak, 1997; Holyoak & Hummel, 2000).

LISA, and the closely related object perception model JIM (Hummel & Biederman, 1992; Hummel, 2001), is able to avoid SME's n-ary restriction that predicates can only be mapped to predicates with the same label and the same number of arguments (although the "same label" restriction is somewhat relaxed by their "tiered identicality constraint" (Forbus et al., 1998)), as it would, for example, allow a proposition with two arguments to be mapped onto a subset of another proposition with three (Hummel & Holyoak, 1997). Thus, the principle of independent, dynamic binding gives us a cognitive structure with increased flexibility, but in such a way that the trade-off (in terms of lost formalizability and predictability) is minimal.

In keeping with its MT-inspired roots, LISA has a hirearchical representation that divides representations into two levels. The top level consists of localist units representing objects, roles, propositions, and groups—essentially everything that is needed to create structures satisfying dynamic, independent binding (Hummel & Holyoak, 1997, 2003a; Hummel & Landy, 2009). The bottom level, however, holds distributed representations, in the form of "semantic units," which may also correspond to features or single-argument predicates. This two-level distinction is an important one in understanding the nature of structure, and will be elaborated further in our discussion of CLARION (Section 2.2).

Although LISA explains analogical retrieval, mapping, and inference, it stops short of the re-representation step and, of course, the fundamentally important structural construction step. The DORA (Discovery of Relations by Analogy) model (Doumas, Hummel, & Sandhofer, 2008) extends LISA and shows how roughly the same neurobiological activity allowing LISA to operate can also explain how higher[7] structure can emerge out of the substructural elements LISA and MT already assume exist. It does this by using the role-filler binding representation style of LISA, so that they can reduce "the problem of learning relations to the problems of learning object properties or relational roles (single-place predicates) and linking them together to form multiplace relational structures" (Doumas et al., 2008). In other words, DORA shows that the analogical comparison ability LISA already has can, with slight tweaking, construct relations out of pre-existing relations of lower arity.

Of course, this raises two questions: First, where do the single-place predicates come from? And second, where do the most primitive comparison operators come

[7]The concept of height in knowledge representations is only applicable if knowledge representations are thought of as tree-like structures with object concepts at the bottom, predicate or propositional concepts linking them at a higher level, and higher-order predicate or propositional concepts linking the predicate or propositional concepts at the level below them. The term 'higher-order' is not to be confused with its use in formal logic (where a higher-order logic may be one that is formally at least as expressive as Second-Order Logic). This confusion of terms is unfortunately deeply ingrained in the analogy literature.

from? DORA offers an answer to the first question: The emergence of single-place predicates can be explained through comparison of already-semantically-rich representations of objects. But the second question is more difficult and relevant to the present discussion. In (Doumas et al., 2008), DORA learns the `HIGHER-THAN` relation by first learning representations of specific values of height. Next, it learns representations of `HIGHER` and `LOWER`, but these are tied to specific heights or categorical values (as they note, children start by categorizing some things as high, and some other things as low). Finally, by comparing the `HIGHER` and `LOWER` representations, DORA is able to learn representations (`HIGHER-THAN`) that are object-feature- and context-independent. This falls out of comparing the predicates, roles, objects, and semantic (I will refer to these units as 'low-level' or 'distributed' from now on to avoid confusion with the use of the word 'semantics' that philosophers are more familiar with, a term I will also need here) units associated with the `HIGHER` and `LOWER` structures. Therefore, structural alignment in DORA explains the origin of comparison operations.

But something seems missing in this approach. DORA's approach, at present, requires that representations of specific values of height, possibly encoded along with the objects that were perceived at those heights, comes first (meaning it comes first in the sequence of steps DORA concerns itself with). And that encoding has to be done in such a way that when a `HIGHER` element is compared to a `LOWER` one, the distributed units connected to each unit corresponding to the specific values of height can be compared to each other (so that the difference between them can be treated as the core of the nascent relationship). But why should the units for height be compared to each other rather than, say, color?

One answer might be that given enough training examples, the relevant distributed units would eventually be selected. This may be possible, given a sufficiently large amount of *supervised* training examples, but this brings to mind an analog of Chomsky's "poverty of the stimulus" line of argumentation (Chomsky, 1965; Piattelli-Palmarini, 1980). Considering the large amount of low-level sensory inputs children have (related to e.g. color, sound, texture, height, etc.), are children given sufficiently many examples of objects with differing heights to narrow down the search space as efficiently as they do, and can we explain this efficient narrowing-down by presupposing minimal innate mechanisms?

A slight formalization may be in order to clarify things here: Let us assume that in some child's mind, two object units O_1, O_2 are respectively connected to the sets of distributed units $D_1 = \{d_1^1, ..., d_1^n\}, D_2 = \{d_2^1, ..., d_2^m\}$. Some subset $H_1 \subseteq D_1$ corresponds to the (perceived) height of O_1, and likewise some subset $H_2 \subseteq D_2$ corresponds to the height of O_2. We tell the child that "O_1 is *higher* than O_2" and leave him to figure out what is meant by that. Now if the child were to compare every possible combination of distributed units in $D_1 \times D_2$, he would end up with a large set indeed (we assume that this example is not a toy example, and thus n and m are likely quite large). How can children so reliably narrow down the possible combinations without a prohibitively large amount of example cases?

It seems to me that a more plausible answer is to suggest that there is a pre-existing connection, or association between at least some of the units in H_1 and H_2, and possibly also between units in the same set as well. What would such associations be? Two possibilities come to mind: either low-level associative links (like the ones in CLARION (Sun & Zhang, 2006) or LISA), or there may be some deeper fact about how the units in $H_1 \cup H_2$ were acquired in the first place. The former suggestion is certainly plausible, but the latter suggestion gives us something deeper to explore. And that "something deeper," I believe, is the underlying semantics those distributed units are encapsulations of—the action from which the declarative concept of height (in the present example) emerged in the first place!

Consider the following example. A child, or a robot, moves its eyes upward. This action can easily be explained through a series of innate actions that the child has in his arsenal; certainly it should not be controversial to assert that there may be a group of neurons along which the child can send a signal that ultimately gets translated into a motion of the eyes in one direction or another. So the child moves his eyes upward, and sees the object O_1. As a result, a series of distributed units are created, capturing all of the sensory knowledge the child is experiencing at this point—the colors, the brightness levels, the sounds, and so on. But the child also creates distributed units capturing the action the child took to make the present state of things possible, and in this case that action is the upward movement of the eyes. Likewise with O_2 and the downward movement of the eyes. If the distributed units are therefore tied to action, we have an origin story for the sort of distributed units that make DORA possible. The pre-existing association between H_1 and H_2 might be explained by the fact that they contain distributed units whose corresponding actions were activated by (roughly) the same groups of neuron firings.

So far, so good. I have tried to draw a link between the (unstructured) distributed units and the actions associated with them. However—and this is a central point of the present chapter—it seems that something gets preserved up at the structural level as well. If I understand that the relationship `HIGHER-THAN`(O_1, O_2) holds, and I am presently looking at O_1, and I want to look at O_2, then solely by virtue of the fact that I think the `HIGHER-THAN` relationship exists between the two objects, I should be able to infer that I can move my eyes downward and expect to find O_2. This is not a deductive inference in the sense that it was an explicit *modus ponens* performed by a reasoner equipped with an axiom governing the use of the predicate `HIGHER-THAN`. Rather, the ability to draw this sort of action-inference[8] can reasonably be expected of a child who has *just learned* the relational predicate `HIGHER-THAN`, who cannot also be expected to also have immediately acquired an axiom set containing rules for working with the `HIGHER-THAN` predicate! Stated simply: DORA does not explain how structure becomes grounded in action, such that every instance of structural knowledge has a set of action implications.

The preceding paragraph is a cue to introduce what will likely be the most controversial, but possibly most important, claim of $\mathcal{AC}$:

No-semantically-empty-structure Principle (strong version) (NSES$_s$)

- In declarative memory, a cognitive relation not grounded in action is semantically empty.

[8]These sort of deep connections between structural (predicate) knowledge and action-inferences were one of Piaget's main conclusions in one of the most important (and again, most ignored) books of LPT—*Toward a Logic of Meanings* (1987/1991) (Piaget et al., 1991).

- In a cognitively realistic system, no cognitive relations are created in declarative memory that are semantically empty.

In Section 2.4, I will defend the strong version of NSES, but $\mathcal{AC}$ is also compatible with a weaker version:

No-semantically-empty-structure Principle (weak version) (NSES_w)

- In declarative memory, a cognitive relation not grounded in action is semantically empty.

- In a cognitively realistic system, few cognitive relations are created in declarative memory that are entirely semantically empty.

In this dissertation, if it is not important whether the weak or strong versions of NSES are referred to, then I will simply drop the subscript.

A corollary of NSES_s is that bottom-up structure creation as in DORA cannot be accomplished in such a manner that the newly created relation is semantically empty. And this is equivalent to the claim that the newly created relation is connected to action in some way. Since both forms of NSES place action at the root of structured knowledge, clearly action itself must be more carefully defined; I tackle this problem in Section 2.4.

In short, I believe DORA is a step in the right direction, but I think that it, and models like it, need to go further in the direction of integrating action and structure. $\mathcal{AC}$ offers a guide to pursue this goal, and furthermore it offers a way to do this largely within pre-existing systems such as DORA, LISA, and SME. However, as we shall see in the next section, it will require the ability to represent action-centered knowledge, at present lacking from most models of analogy.

2.2 CLARION and the Case for Action-Centered-Knowledge

Although the distinction between distributed (low-level) and localist (top-level) representations is somewhat captured in LISA and DORA, the two knowledge types are perhaps best captured in the hybrid cognitive architecture CLARION (Sun, 2002). CLARION sets itself apart from other cogntive architectures by embracing two fundamental dichotomies: The first is the distinction between explicit (localist) and implicit (distributed) knowledge, and the second is between action-centered (procedural) and non-action-centered (declarative) knowledge. Unlike other cognitive architectures like ACT-R (Anderson, 1976), CLARION treats the two dichotomies as if they existed in orthogonal directions, leading to four divisions of knowledge: explicit action-centered knowledge (EACK), implicit action-centered knowledge (IACK), explicit non-action-centered knowledge (ENACK), and implicit non-action-centered knowledge (INACK) (these acronyms are not used by Sun, but are introduced here for easy reference later). The psychological data supporting the orthogonal treatment of these dichotomies is extensive; there is evidence, e.g., that all four types of knowledge exist in humans (Sun, 2002, 2012b).

The distinction between procedural (action-centered) and declarative (non-action-centered) knowledge (Anderson, 1976) is rooted in the philosophical distinction between reasoning *in-accordance-with* (knowing how) and *reasoning-about* (knowing what) referred to in Chapter 1 (Winograd, 1975). Recall in Section 1.3.2 it was suggested that the *réfléchissement* step might be understood as a transformation from procedural to declarative knowledge. Such a transformation, then, is one possible way that structure in ENACK might be created.

Because the nature of distributed representations only allows for associative links between non-localist units (Sun, 2002), the requirement of dynamic, independent binding rules out the existence of structure in either IACK or INACK. Do the cognitive correlates of what we know to be structured knowledge exist in EACK? According to NSES, structure is defined as being rooted in action. There is an element of redundancy in asking whether the representations of actions and action schemas (the content of EACK) themselves are structured. However, it is difficult to deny that action schemas do not contain at least some primitive sort of structure, since they typically contain some sort of ordered linking between e.g., a context, an action, and an effect (Drescher, 1991). So we will instead call action schemas 'pseudo-structured' and leave it at that for now.

Structure, as $\mathcal{AC}$ understands it, only exists in ENACK (this dissertation says nothing about other possible types of knowledge, such as might exist in a meta-cognitive or motivational subsystem, as in (Sun, 2002)). The localist, structurable units that lie at the intersection of declarative and explicit knowledge are capable of supporting precisely the properties of structured knowledge catalogued by the present chapter. The second part of NSES, with its reference to declarative memory, points to this fact. But the first part, the claim that the semantics of relations are grounded in action, must next be defended.

2.2.1 Action-based Semantics

NSES states that a relation must be grounded in action, otherwise it is semantically empty. This means that it is relying on a notion of semantics that draws its support from two lines of thought. The first of course, is LPT. The second is Floridi's Action-based Semantics (AbS). I will briefly summarize both views here.

Piaget and Garcia's *Toward a Logic of Meanings* (Piaget et al., 1991) attempted to create a "logic of meanings" by demonstrating the existence of implications between actions and operations in young children. They carried out a set of experiments designed to detect a sort of reasoning not necessarily based on truth-values, but rather on action-implications, i.e., *what can be done* with objects? These experiments yielded some interesting discoveries, including evidence for the existence of connectives between actions "isomorphic to 10 of the 16 future binary operations. [...] The 10 that are used are all connectives between meanings that do not depend on an extensional truth table" (Piaget et al., 1991, p.66). Their conclusions were a series of statements about meaning:

- "an object is a set of conjoined predicates and its meaning amounts to 'what can be done' with it, and is thus an assimilation to an action scheme" (Piaget et al., 1991, p. 119)
- "From a general standpoint, the two meanings of an object are, subjectively, what can be done with it and, objectively, what it is made of or how it is composed. The former cannot be separated from the meaning of actions" (Piaget et al., 1991, p. 57)

Although these statements may be seen as a bold declaration of views Piaget had been arguing for in some form or another for most of his career, he was not able to further justify these claims, as *Toward a Logic of Meanings* was published only 3 years before his death.

A more modern theory of meaning that seems to reflect the sentiments behind LPT's final views on meaning (though it does not trace its ancestry to Piaget's work) is Luciano Floridi's Action-based Semantics (AbS) (Floridi, 2011b). Floridi recalls the well-known Symbol Grounding Problem (SGP) (Harnad, 1990), and argues that a satisfactory solution to the SGP must satisfy three conditions comprising the Zero Semantic Commitment Condition (Z condition):

1. No form of innatism is allowed; no semantic resources should be pre-installed in the artificial agent.

2. No form of externalism is allowed either; no semantic resources should be uploaded from the 'outside'.

3. The artificial agent may have its own capacities and resources (computational, procedual, perceptual, etc.) to ground its symbols with (Floridi, 2011b, p.137).

In a discussion of CLARION, Floridi argues that CLARION's learning algorithms in its two-level approach do not solve SGP, in part because CLARION's solution to the emergence of intentionality is to rely on a *first-order intentionality* to come from the interaction of the implicit layers with the world. The explicit layers can then construct conceptual representations from the first-order intentionality provided by the lower levels. According to Floridi, the answer of how the first-level intentionality emerged in the first place is unanswered, and thus "[u]nless a logically valid and empirically plausible answer is provided, the SGP has simply been shifted" (Floridi, 2011b, p. 147).[9]

Floridi proposes AbS as a theory of meaning to explain how new meanings can be generated and symbols attached to those meanings. Such a process of meaning generation begins with proto-meanings derived directly from (or exactly equal)

[9] Floridi, however, may be referring to an outdated understanding of what CLARION is capable of. See (Sun, 2012a) for more recent work in CLARION on symbol grounding, and (Bringsjord, 2015) for a criticism of Floridi's answer to SGP.

the agent's internal states, and those internal states are "directly correlated to the actions performed by the same [agent]" (Floridi, 2011b, p. 164), where the actions used may be *teleologically free* (not necessarily connected to any end goal or purpose). In other words, the part of the agent's internal state corresponding to the action it is currently taking, as opposed to its current command being executed or end goal, is what is used to create the proto-meanings eventually attached to symbols.

AbS is then implemented by a type of artificial agent Floridi calls a "two-machine artificial agent", or AM^2. The agent has an architecture consisting of an object level (OL) and a meta-level (ML). Here we see more overlap with LPT. The OL interacts with the external environment and allows for the coordination of actions, whereas the ML takes the actions at the OL as its data. Furthermore, the "relevant metaprograms are the *reflection processes*, where these function as *upward reflection* [...] The utility of reflection shows that the whole system [OL + ML] not only interacts with itself but is also properly affected by the results of such interactions" (Floridi, 2011b, p. 167). Floridi here draws his reflection-related terminology from the so-called reflective architectures of Brazier and Treur (1999) and Barklund et al. (2000); these reflective architectures center around a notion of reflection more-or-less synonymous with introspection, self-referencing, and metareasoning (interaction between an OL and ML) (Brazier & Treur, 1999; Barklund, Constantini, & Lanzarone, 2000). The reflective architectures cited by Floridi seem to trace their ancestry to early work in meta-levels (see the survey in (Harrison, 1995)), but the concept of a "reflection principle" comes from Feferman (1962), who defines it as:

> [A] description of a procedure for adding to any set of axioms A certain new axioms whose validity follow from the validity of the axioms A and which formally express within the language of A evident consequences of the assumption that all the theorems of A are valid (Feferman, 1962).

So although the terminology of reflective architectures appears close to that of LPT, they refer to quite different conceptions of reflection. It is interesting to note that none of the literature on reflective archictectures refer to Piaget at all, though Piaget's work precedes most of them by a number of years (likely an example of the sort of lack of communication between two fields that this dissertation attempts to bridge).

I will not dwell further on the similarities or differences between LPT and Floridi's AbS. Rather, I only include them here to show that the approach $\mathcal{AC}$ takes is supported both empirically (through the studies done by LPT) and philosophically, and the cursory pointers to the relevant literature thus far will have to suffice for the interested reader.[10] We are finally in position to introduce the central tenets of Analogical Constructivism.

2.3 Introducing Analogical Constructivism ($\mathcal{AC}$)

First, a clarification. $\mathcal{AC}$ attempts to be an approach to modeling analogy and cognition that is largely supplemental; $\mathcal{AC}$ should *not* be considered a competing theory to SMT, MT, or that of the Copycat Group. It should, rather, be properly considered as an approach that can theoretically be implemented with SME, ACME, Copycat, LISA, DORA, CLARION, ARCADIA (Bridewell & Bello, 2015), or the large majority of any of the other currently existing models of analogy or cognition, so long as those models are not based on assumptions significantly incompatible with the tenets of $\mathcal{AC}$, summarized as follows:

Analogical Constructivism ($\mathcal{AC}$)

$\mathcal{AC}$.1 The nature of cognitive structures and their complex relationship with action-centered knowledge should be central foci of models of analogy and cognition.

$\mathcal{AC}$.2 Cognitive structures and action schemas can bootstrap each other in complexity, through a process of reflective abstraction.

[10] For a recent counter argument to Taddeo and Floridi's proposed solution to the Symbol Grounding Problem (first presented in (Taddeo & Floridi, 2005, 2007)), see (Bringsjord, 2015). Bringsjord's analysis is language-focused. Indeed, the language in question is at the level of adult humans.

$\mathcal{AC}$.3 Many human-level thought processes are manifestations of analogical comparison between structured representations.

$\mathcal{AC}$.4 NSES_w:

- In declarative memory, a cognitive relation not grounded in action is semantically empty.
- In a cognitively realistic system, few cognitive relations are created in declarative memory that are entirely semantically empty.

$\mathcal{AC}$.1 is warranted by the discussions in Chapter 1 and the first half of the current chapter, according to which the concept of structure is extremely important to understanding how human knowledge (or at least the part of knowledge relevant to analogy and higher-level reasoning in general) works, behaves, and should be modeled. One important way structured knowledge and actions interact is stated by $\mathcal{AC}$.2 and $\mathcal{AC}$.3, which again were discussed, and their motivations explained, in Chapter 1. $\mathcal{AC}$.4, however, requires two more clarifications. NSES is strictly about cognitive structures: It does not say anything about structures "in the world." NSES also says little about systems where human-like abilities are not a priority; this leaves open the possibility that an approach not obeying NSES may create a convincingly human-like AI (but it does express doubt that such a task is possible without NSES).

2.4 A Defense and Clarification of NSES

Clearly, both NSES_s and NSES_w require a considerable amount of elaboration, offered in this section.

Let us differentiate between constructions that are rooted in action and those that are not. We will define **action-rooted constructions** as mental constructs which both (1) originated with representations of actions or other action-rooted constructions as significant components, and (2) retain actions or other action-rooted constructions as a part of their constructions (though the actions retained by the constructions may not always be the same as those the construction originated from). Now it would be pointless to say that every construction is action-rooted if it is done with the help of an action (because the act of constructing itself is trivially an action, according to the definition I will introduce), so I use the language "originated with representations of actions as significant components". So the act of relation construction in declarative memory requires representations of actions (in IACK or EACK) to be bundled up with the package, though the actions bundled in the package may change over time.

An **action**, in the sense of the word used in this dissertation, is any event or chain of events initiated by the mind of an agent. In its simplest form, an action can be completely automatic and unintentional. For example, consider two ENACK chunks c_1, c_2 having some associative link to each other. If c_1 is activated for some reason, activation may spread to c_2. This spread of activation is, under the definition I use, an action—an extremely simple action, perhaps somewhat far removed from the commonsense definition of action, but an action nonetheless. If the agent has some ability to become consciously aware of which of its ENACK chunks are active, it might at some point become aware that the activation of one chunk is followed by the activation of the other, but the representation of such an awareness must be distinguished from the action itself.

Under such a definition of action, assuming one believes that computers or artificial agents can have minds, we must accept that artificial agents have actions. But this definition of action does not commit to accepting that artificial agents have *intentional* actions, or that they can or must be aware of such actions. Similar remarks can be made for non-human animals. However, my definition of action does depend on the highly controversial assertion that action requires a mind. I will not define 'mind' in this dissertation, being aware of the difficulties of such an endeavor, and will leave the reference to 'mind' *talis qualis*.

The definition of action used here is to be contrasted with Piaget's definition. Piaget distinguished between actions and operations (internalized actions). Although it may be helpful to separate actions that operate on purely mental constructs from actions that result in actualizations in the physical world, $\mathcal{AC}$ treats both as subsets of actions, reflecting its implicit assumption that both types of actions come from the same mechanisms.

What, then, is a non-action-rooted construction? I have been refering to these as **pure observables**; these are constructions existing in declarative memory without being associated with something that the agent "knows how to do", i.e. they either are not constructed from actions, or their recognition does not imply the possibility of actions. They are constructions that do not contain references to actions in their encoding. Such pure observables are few and far between in human beings. As a rule of thumb, if an agent refers to an action as part of the recognition or verification process of some construct, or if recognition of that construct causes the agent to automatically infer the possibility of some action, then the construct is not a pure observable.

For example, consider the featural invariants of relations. These are used in the construction of new relational constructs in the DORA model, and outputted by models of visual perception such as JIM (Hummel & Biederman, 1992). Yet the origins of such featural invariants have to date not been accounted for (Doumas et al., 2008), at least not completely.[11] Purely visual featural descriptors, such as might be reliably recognized by a suitably trained artificial neural network, can be considered pure observables in that they can conceivably be learned without the help of a system suitably rich in action-centered knowledge (although even visual featural invariants might have action implications, e.g. see Section 2.4.2.1). But it is difficult to see how such a network could then use its learned symbols in a way that matches the semantic richness of a system that takes $\mathcal{AC}$ seriously. Such a system could learn that object A is above object B, but would it automatically know that removal of object A would be necessary to reach object B? Would it be able to predict that movement of object B would cause movement of object A?

NSES is therefore the centerpiece in $\mathcal{AC}$'s argument that although the focus on the construction of semantically empty declarative structures have no doubt been productive, they will not be sufficient to create intelligent agents with semantically rich, *truly* structured representations.

[11] According to Doumas et al. (2008), the DORA model does "not provide a complete answer to the question of where relational invariants come from," and focuses instead on being able to "isolate them from other features and represent them as explicit structures that can take arguments" (Doumas et al., 2008, p.2-3).

2.4.1 Comparison with Affordance Theory

Around the time that the major works of LPT were starting to see their first translations into English, J.J. Gibson began writing about what he called **Affordance Theory**. According to Affordance Theory, when we look at objects we do not perceive their qualities, but rather their affordances. An **affordance** of the environment is what the environment "*offers* the animal, what it *provides* or *furnishes*, either for good or ill" (Gibson, 1977). Alternatively, affordances might better be described as not properties at all, but rather relationships between aspects of animals and aspects of situations (Chemero, 2003).

$\mathcal{AC}$, however, is more extreme in its position on the meaning of perceptions. As Floridi explained in his description of the differences between Wittgenstein's meaning-as-use semantics and his own action-based semantics, in linguistic games, meaning is the way a symbol is to be used within that particular game. But in action-based semantics, the meaning of a symbol is the internal (action-rooted) state of the agent associated with that symbol, and is *not* defined by the external action, its externally-defined use, or the external action's results.

I do agree with Affordance Theory that perception is affected by actions of the agent. Witt's theory of Action-Specific Perception, derived from Affordance Theory, argues that we perceive and remember objects in the environment based on our abilities to act on them (Witt, 2011). For example, softball players who performed better in games later reported that the game balls were larger in size than the players who performed poorer (Witt & Proffitt, 2005). Although Witt and Proffitt acknowledge the relationship between perceived size and player performance is not yet proven to be a causal one in either direction, they reference other experiments reporting similar effects.

$\mathcal{AC}$, then, has some significant overlap with Affordance Theory, though $\mathcal{AC}$ is more directly descended from constructivist epistemology and tries to understand mental constructs *as constructs*; that is, how knowledge structures are created, built up, combined, destroyed, and change. $\mathcal{AC}$ is more interested in the general nature of structured knowledge and how it behaves not only in instances of perception, but in instances of thinking, acting, and learning as well.

But the most significant difference between $\mathcal{AC}$ and Affordance Theory is captured in a common phrase used by some authors to summarize Affordance Theory: "perception drives action" (Friedenberg & Silverman, 2012; Kosgeroglu, Acat, Ayranci, Ozbaci, & Erkal, 2009; Azevedo, Reis, & Cornacchione Jr., 2013). In my view, this is wrong. $\mathcal{AC}$ says, rather, actions drive perception. Or perhaps more precisely, action-rooted constructions, created through reflective abstraction, are the building blocks used by empirical abstraction (though this phrasing is less catchy).

2.4.2 Example Relational Concepts

Let us now consider some examples. In each of the cases that follow, I will present a natural-language relation and try to identify typical action inferences that might have been used during the construction of the concepts corresponding to that natural language relation.

Let us start by pseudo-formalizing a corollary of NSES$_s$ called $\mathscr{N}$: If some agent has $R(b,c)$ in declarative memory, then there is a nonempty set of construction-relative actions (called the action inferences of R) the agent believes are possible involving b and c.

$\mathscr{N}$ takes the action inferences of R to be *construction-relative*, meaning that the action inferences of R may vary for different agents depending on how R was constructed for those agents. It might be argued that having two different constructions for some relation R means that they are not quite the "same thing," so we must be careful when we answer questions of the form "What are the action inferences for the relational concept `C`?", where the concept `C` is mapped to the linguistic signifier 'C'. Such a question will be interpreted as "What are *typical* action inferences of the relational concepts `C` that agents map to the signifier 'C'?"

Another important component of $\mathscr{N}$ is the relation between the declarative concept R and its modal relation to its action inferences. In $\mathscr{N}$'s phrasing, it is one of belief, so that if $a(b,c)$ is an action inference of R for some agent α, then $\mathbf{B}^\alpha(R(b,c)) \rightarrow \mathbf{B}^\alpha(\Diamond^\alpha a(b,c))$ where $\Diamond^\alpha$ denotes some sense of possibility relative to the actions of agent α. It is perhaps misleading to use such declarative notation to express the relation between a and R, because the belief the agent has in the possibility of action a is not necessarily a structure in declarative knowledge. Rather, it is a construction in procedural knowledge (hence the α superscript to distinguish it from the more well-known epistemic modal operator **B**).

2.4.2.1 Actions as expectation primers

Under the definition of action used by $\mathcal{AC}$, something as simple as a spreading of activation can be considered an action, albeit an unintentional one. If psychological priming can be explained by the activation of concepts in memory, then priming may also be considered an action. Such an action might be an action inference of a predicate *IsSquare(object)*, manifested when either a square is partially recognized, or when one is told verbally that an object should be perceived as a square.

2.4.2.2 Cause-effect and Sequences

Action inferences might also include expectations for observables in the future, or expectations for "what should happen next." For example, if a child were trained through repetition that whenever hearing a certain sound a certain image will appear momentarily in a certain location l, then the child may create a perceptual symbol s associated with the sound, and attach s to action inferences involving the motion of his eyes towards l. In this way, sequences of purely external events, linked temporally through repetition, can be encoded in the child's mind.

It is possible that much of what is understood as the epistemological origins of causality can be understood in terms of the mechanisms previously described. Certaintly there is much more to it, though much of this is outside the scope of this dissertation. Piaget, particularly in the LPT period, had a nuanced view of how the understanding of causality developed, and his view may offer a set of directions worth exploring (Piaget & Garcia, 1974). Instead, consider the example where an agent observes a boulder falling on to the ground and emitting a loud crashing noise. How would the agent encode the knowledge of what he just saw? His auditory perceptual processes might create a pure observable symbol CRASH, corresponding to the crashing noise. Empirical abstraction recruits the previously-created symbols for BOULDER and GROUND, and some low-level, sub-structural process may associate the two. An association between BOULDER and GROUND may also form with the newly created symbol CRASH. But the resulting representation is *not* structured, according to NSES, unless they are (at a minimum) linked together under a symbol rooted in action. In this case, an appropriate set of action-rooted symbols might be CAUSE-EFFECT and TOUCHES, so that the resulting representation becomes CAUSE-EFFECT(TOUCHES(BOULDER, GROUND), CRASH).

2.4.2.3 Similarity, Difference, and Identity

Identity is often considered to be the prototypical example of an irreducible relation, i.e. a relation between two elements that cannot be translated into more basic relations between those two elements (Jackson, 1977). Similarity and difference may also be as irreducible as identity, and are therefore likely candidates for innate relations. But the ability to represent the relational predicates for similarity, difference, or identity should be distinguished from any actions of producing inferences on the basis of the existence of those relational predicates, or whatever physical chain of events, whether neurobiological as in LISA or electrical as in computers, causes the determination of similarity, difference, or identity to arise in the first place. An agent may be able to recognize and perform actions in response to observing similarities or differences, but it does not follow that the same agent can represent the observed similarity or difference *as a relation of similarity or difference.*

2.4.2.4 Reasoning and Inference

Since we have extended action to apply to comparisons between, and creation of, mental representations, presumably reasoning and inference are also subsumed. The logical operations themselves, according to LPT, are internalized actions created through reflective abstraction. We might even be tempted to say that the most basic inference mechanism that allows for the creation of new declarative structures, utilized by both analogy and deduction, is an action. Given that inference is such a basic and seemingly irreducible operation, we may be forced to accept the innateness of such an action (but I will not elaborate on this further in this dissertation).

However, as will be shown in Chapter 3, both deduction and analogy can be modeled by a process sensitive to just the sort of cognitive structure the present chapter is trying to define. Reasoning, and the autonomous development of reasoning ability, is a profoundly structural process, and studying it is one of the ultimate goals of Analogical Constructivism.

2.5 Conclusion

Although the list in Section 2.4.2 cannot be expected to capture all possible relations humans deal with, it should at least offer preliminary support for the validity of NSES_s, and its weaker form NSES_w. Although a proper and full-length defense of NSES and the other tenets of $\mathcal{AC}$ is warranted, such a defense would take up an entire dissertation, and the defense given thus far will have to do for now.

This chapter analyzed the conception of structure, an idea central to all models of analogy and cognitive architectures in general. I concluded that structure must be rooted in action, and this line of thought resulted in the No-semantically-empty-structure Principle (NSES). NSES takes its understanding of semantics from LPT and AbS. I then introduced the tenets of Analogical Constructivism.

This chapter concludes the bulk of the theoretical discussion behind $\mathcal{AC}$. The following chapters will largely focus on technical details behind a demonstration of NSES in action: a simulation environment capable of supporting an agent able to produce structured representations rooted in its actions, and then a demonstration of this agent taking first steps toward doing so.

CHAPTER 3
Representing Knowledge

In this chapter I will discuss the knowledge representation used by the cognitive simulation in this dissertation, and how the phenomena described in prior chapters (such as *réfléchissement*, *réflexion*, structured knowledge rooted in action, and so on) can be faithfully modeled using the style of knowledge representation I have chosen.

3.1 Prior Work

$\mathcal{AC}$ attempts to be an approach to cognitive modeling that is compatible with many of the already-existing paradigms (most of which arguably have more in common than their creators might be willing to admit). The style of cognitive modeling used by this dissertation is a natural continuation of research whose goal was to model **Analogico-Deductive Reasoning** (**ADR**), which is best described as the intersection of hypothetico-deductive and analogical reasoning (Licato, Bringsjord, & Hummel, 2012; Bringsjord & Licato, 2012). In a typical example of ADR, a reasoner uses analogy to generate a hypothesis h, where h may have been created to explain some observation or theory about the world. h is then subjected to a deductive reasoning step, where the reasoner attempts to show that either h or $\neg h$ follows deductively from a set of axioms $\mathbf{A}$. Depending on the implementation, $\mathbf{A}$ might derive from the reasoner's beliefs, the axioms of a mathematical theory, or anything else which can be stated in the relevant proof theory's language. If some proof $\mathbf{A} \vdash h$ is found, then h is accepted; otherwise if some proof $\mathbf{A} \vdash \neg h$ is found then the analogy which generated h in the first place is rejected.

ADR is a very generalized and domain-independent reasoning technique which relies on a minimal apparatus of hypothesis generation through analogy, and some

Portions of this chapter previously appeared as: Licato, J., Sun, R., & Bringsjord, S. (2014). Structural Representation and Reasoning in a Hybrid Cognitive Architecture. In *Proceedings of the 2014 International Joint Conference on Neural Networks (IJCNN)*. Beijing, China: IEEE.

deductive method to confirm or refute those hypotheses. Not surprisingly then, ADR can be used to explain the reasoning used by young children working through Piagetian tasks (Licato et al., 2012; Bringsjord & Licato, 2012), all the way up to professional logicians such as Gödel (Licato, Govindarajulu, Bringsjord, Pomeranz, & Gittelson, 2013) and Goodstein (Govindarajulu, Licato, & Bringsjord, 2013).

In some of our explorations into ADR (Licato et al., 2012; Bringsjord & Licato, 2012), Hummel and Holyoak's LISA model (Hummel & Holyoak, 1997, 2003a) was used for the analogical steps. The knowledge in an instantiation of LISA can be translated into a human-readable language referred to as "LISA-ese," and propositions expressed in LISA-ese can easily be imported into a deductive reasoner. LISA's explicit representation of propositions, objects, and roles as units also provided some of the inspiration for the implementation adopted in (Licato et al., 2014). (Licato et al., 2014) introduced a way of capturing declarative "structure" (though according to $\mathcal{AC}$, this was not true structure, but semantically-empty/non-action-based structure) in CLARION's non-action-centered subsystem (NACS). This capturing was achieved mostly through the use of two features already present in the NACS: localist chunks (hereafter referred to alternately as "declarative chunks" or just "chunks" if it is clear from context) and associative rules (ARs).

3.2 Declarative Knowledge in CLARION

In CLARION, non-action-centered knowledge is contained in the non-action-centered subsystem (NACS). The NACS consists of a top level (the **general knowledge store**, or **GKS**, which stores ENACK) and a bottom level (the **associative memory network**, or **AMN**, which stores INACK). The AMN holds implicit associative knowledge encoded as **dimension-value pairs** (**DV pairs**). Each localist unit in the GKS (referred to as **chunks**) connects to a set of DV pairs in the AMN with some adjustable weight. CLARION therefore can define a *directed* similarity measure between two chunks c_1 and c_2, derived from the amount of overlap between the DV pairs connected to the two chunks (Sun, 1995; Tversky, 1977; Sun & Zhang, 2004):

$$S_{c_1 \to c_2} = \frac{\sum_{i \in c_2 \cap c_1} W_i^{c_2} \times A_i}{f(\sum_{i \in c_2} W_i^{c_2} \times A_i)} \tag{3.1}$$

Where $f(x) = x^{1.0001}$. Sun and Zhang (2004) define A_i as the strength of activation of the values of dimension i in chunk c_1, and $W_i^{c_2}$ as the weights of the DV pairs specified with respect to c_2. For simplicity, we will set all A and W values to 1, which reduces Equation 3.1 to a function of the number of dv pairs connected to c_1 and c_2:

$$S_{c_1 \to c_2} = \frac{|c_1 \cap c_2|}{|c_2|^{1.0001}} \tag{3.2}$$

The denominator in Equation 3.2 may possibly be zero, in which case $S_{c_1 \to c_2}$ is set to 1.

An **Associative Rule** (**AR**) consists of a set of condition chunks $c_1, c_2, ...$ and a single conclusion chunk d. For any given AR, each condition chunk i has a weight W_i such that $\sum_i W_i = 1$. ARs can be written as:

$$(c_1, c_2, ..., c_n) \Rightarrow d$$

The chunks in the GKS and DV pairs in the AMN have activation levels which can be set by CLARION's other subsystems. Activations can also spread through the NACS using the chunk-DV pair connections and the top-level ARs. The manner in which this activation spreads can be restricted: other subsystems can temporarily disable Rule-Based Reasoning (activation spreading through ARs) or Similarity-Based Reasoning (activation spreading through chunk similarity), or perform activation propagation as some weighted combination of both of these reasoning types. These abilities are detailed further in Sun & Zhang (2004, 2006), where these mechanisms are shown to be psychologically plausible by using them to closely emulate the results of psychological studies. We use no more than these mechanisms to construct the knowledge structures in this chapter.

The activation levels of the units in the GKS and AMN can be set by the other subsystems of CLARION. Activations can likewise spread through both level of the NACS, and this spreading can be confined: Rule-Based Reasoning (activation spreading through ARs) or Similarity-Based Reasoning (activation spreading

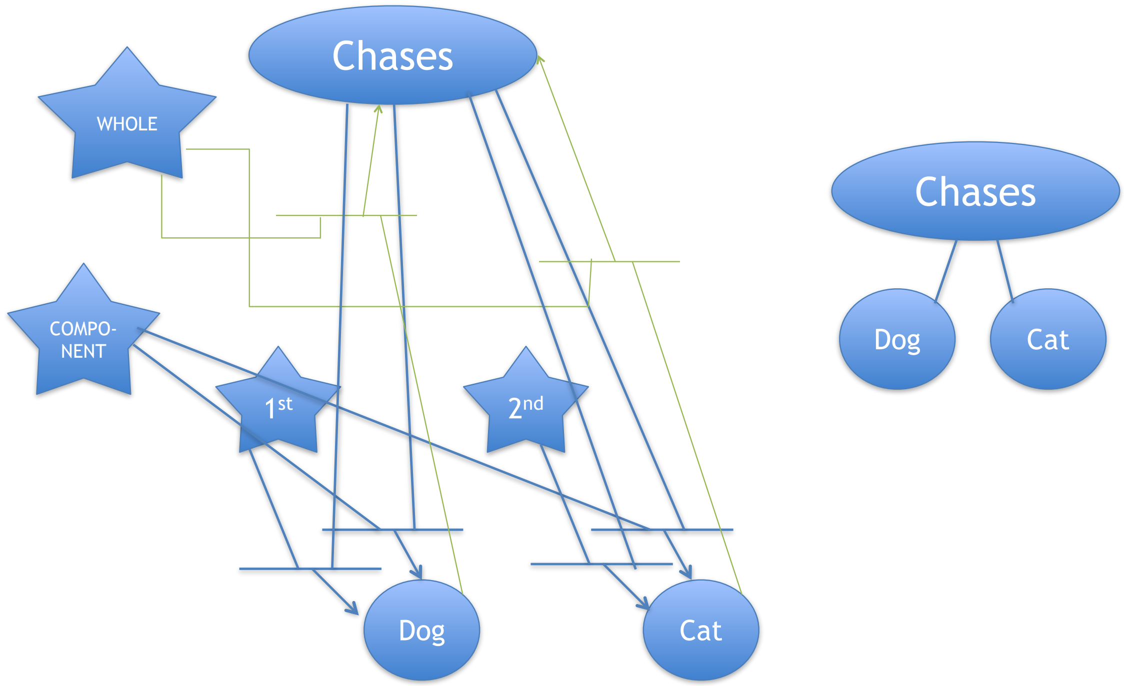

Figure 3.1: A knowledge structure representing the proposition $CHASES(DOG, CAT)$. On the right is the simplified version, which does not picture the CDCs and many of the ARs. This figure previously appeared in Licato, J., Sun, R., & Bringsjord, S. (2014). Structural Representation and Reasoning in a Hybrid Cognitive Architecture. In *Proceedings of the 2014 International Joint Conference on Neural Networks (IJCNN)*. Beijing, China: IEEE.

through chunk similarity) can be temporarily disabled, for example. These capacities are further itemized in Sun & Zhang (2004, 2006), in which these components are used to model psychological data, showing them to be psychologically plausible.

3.2.1 Introducing Chunk Types

It is possible to assign *types* to chunks. The scope of these types, however, holds only within the context of a complete structure, for reasons we will explain shortly.

The first type of chunk is the object chunk. This is often paired with a proposition chunk, which is both a placeholder for the proposition's predicate symbol and a marker of the relationship between the object chunks. Both the proposition and object chunks are pictured in Figure 3.1 as ovals.

Object and proposition chunks need to somehow be linked together, and that role is played by **Cognitively Distinguished Chunks** (**CDC**s). If, in keeping with Licato et al. (2014), we assume that the ability to perform structured reasoning is possessed by all neurobiologically normal adult humans, then we can safely assume that there are cognitive correlates which are either innate or develop very early in life underlying structured knowledge. CDCs are meant to reflect these abilities and as we will see shortly, we can use CDCs to connect to action knowledge.

CDCs are pictured using stars (as in Figure 3.1). ARs connect the CDCs to chunks, e.g., object chunks may be connected to proposition chunks using an AR also connectd to a *WHOLE* CDC. In Figure 3.1 (which depicts the proposition *CHASES(DOG CAT)*), the *WHOLE* CDC is part of two ARs (each AR appears as an arrow with multiple tails and one head):

$$(DOG, WHOLE) \Rightarrow CHASES$$
$$(CAT, WHOLE) \Rightarrow CHASES$$

A *COMPONENT* CDC is also defined to introduce some redundancy into the structure, such that for every rule involving a *WHOLE* CDC, a complementary rule going in the other direction is created with a *COMPONENT* CDC. In Figure 3.1, these ARs are:

$$(CHASES, COMPONENT) \Rightarrow DOG$$
$$(CHASES, COMPONENT) \Rightarrow CAT$$

Whole chunks are always drawn above component chunks whenever possible. *Ordinal CDCs*, which also appear in Figure 3.1 as *1ST, 2ND,* etc., allow the object chunks to fill distinct roles within the overall propositional structure. Ordinal CDCs establish independent, dynamic bindings as laid out by (Hummel & Holyoak, 1997; Holyoak & Hummel, 2000). In the case of Figure 3.1, the existence of ordinal bindings suggests that the reasoner has the knowledge of how to retrieve the object which fills the "first" role in the proposition *CHASES(DOG CAT)*, implying a structural organization which implies an absolute argument order. But the ordinal CDCs are not the only type of CDCs that can specify independent, dynamic bindings. For example, one might also create CDCs $SUBJECT$ and $OBJECT$ and create the following ARs:

$$(CHASES, SUBJECT) \Rightarrow DOG$$
$$(CHASES, OBJECT) \Rightarrow CAT$$

The division of chunk types and the use of ARs described so far is therefore quite flexible in that it can have objects play multiple roles within a proposition simultaneously, and structures can also be nested, so that instead of an object chunk a proposition chunk can have another proposition chunk as a component. Proposition chunks can also have the same object chunks as components multiple times, as in the proposition $P(a, X, a)$ (Figure 3.2). Although there is in principle no limit on the amount of nesting that can be done so that structures can reach arbitrary heights, it may be worthwhile to explore limits in future work.

3.3 Performing Reasoning

It should not be too controversial to suggest that something innate exists that allows basic traversal of knowledge structures, an ability afforded to us by the CDCs we defined in the previous section. But in order to really demonstrate the power of this system to perform higher level reasoning, we need to show that it can match structures based on form, a prerequisite shared by both analogical (Gentner, 1983) and deductive reasoning, as I will now explain.

3.3.1 Templates and Form Matching

Deductive reasoning uses form-based matching when determining whether or not an inference rule applies. To use a standard inference rule as an example, assume that we know all men are mortal. Such a statement can take the following form, with X as a variable ranging over some predefined universe:

$$Man(X) \rightarrow Mortal(X) \tag{3.3}$$

If given the statement $Man(socrates)$, a reasoner would have to first match the form specified in the antecedent of Equation 3.3. If a match is made, there should be enough information available to inform us how to transform the input statements to produce a new statement (the inferred statement) in accordance with the form specified in the consequent of Equation 3.3; that resulting formula is $Mortal(socrates)$.

All of this should be quite familiar to anyone who remembers their first experiences with deductive reasoning. Whenever deductive reasoning takes a general rule and applies it to some specific statement, it performs form matching between the general rule and the statement. But what happens when we instead start with a slightly different statement:

$$Man(plato) \wedge Mortal(plato) \tag{3.4}$$

Given now the statement $Man(socrates)$, it does not follow from deductive reasoning that Socrates is mortal. If it does follow from these statements, it is through *analogical* reasoning—Plato was also a man, therefore by analogy it is plausible that Socrates is also mortal. In a template such as that in Equation 3.3, the antecedent clearly specifies a predicate portion that must be matched exactly (Man), and an object portion that can be anything over which the X variable ranges. In the case of Equation 3.4, the statements $Man(plato)$ and $Man(socrates)$ do not line up exactly—the objects *plato* and *socrates* do share the primary similarity specified by the predicate (they are both men), but an analogical reasoner would likely find similarities between them in other respects: they are both philosophers, they are both from Ancient Greece, etc.

These examples suggest that when matching structured knowledge forms with the end goal of performing deductive or analogical reasoning, at least two things should be available: Firstly, we need to know what constitutes an acceptable match. This may require an exact alignment as in Equation 3.3, or it may allow a relaxed requirement of surface similarity, as in Equation 3.4. Secondly, once the match is made, we need to know what resulting inference, or transformation of the input, can be made, and how to do it. This is specified nicely by the consequent portions of both Equations 3.3 and 3.4.

To achieve these goals, we introduce the **template**, which builds on the method of representing structured knowledge we defined earlier in this paper. Templates are groups of chunks that both specify what constitutes an acceptable form match and how to transform the input when such a match is found. In order to represent templates, two features are created. The first is the Template Chunk, which is a chunk type used by chunks specifically designated to identify complete templates.[12] Template chunks are connected to the template's individual chunks using the template CDC, which is identified in our diagrams by a star encasing the letter 'T'.

For every chunk c in some template identified by the template chunk tc, an associative rule connects these chunks to the template CDC T:

$$(tc, T) \Rightarrow c$$

Since every AR can be weighted as well (not to be confused with the weight of the individual condition chunks within the AR itself), the weights of all ARs outgoing from any particular template chunk adds up to one; this allows us to specify how much matching some particular chunk contributes to the match score of the overall template. Such ARs can have weights of zero, and chunks within templates which have zero weights in their corresponding ARs are pictured using a circle with a double border (Figure 3.2).

Chunks can exist in templates that have zero semantic content. These are called **blank chunks**, and will be used when matching templates to other structures.

Our TF method does not allow chunks to have multiple parents. When a quantified variable appears in multiple locations, it is important to preserve the fact that although separate chunks are created for each instance of the variable, since they correspond to the same variable, any chunks matched to these instances must correspond to the same object (or as we will see, this restriction can be relaxed to allow for objects whose chunks have an extremely high similarity). This restriction is reflected in TF using **identity links**, which are pictured using double lines between chunks (Figure 3.2). Identity links are implemented using the Linker CDC (L, not pictured in Figure 3.2). such that for any two chunks c_1 and c_2 which are linked, the following ARs are created:

[12] We do not in this paper discuss how such templates arise in the first place; this is the subject of future work.

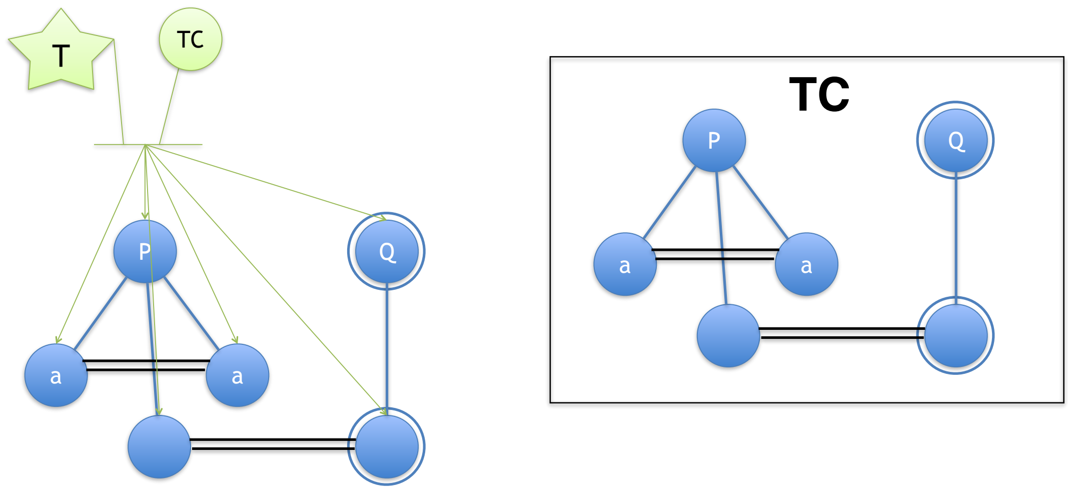

Figure 3.2: A typical template with zero-weighted chunks and blank chunks. A simplified version is on the right, which is equivalent to the left picture. Also note that whenever ARs are pictured with multiple heads like in this figure, each head corresponds to a separate AR which has the same tail connections as the others. This figure previously appeared in Licato, J., Sun, R., & Bringsjord, S. (2014). Structural Representation and Reasoning in a Hybrid Cognitive Architecture. In *Proceedings of the 2014 International Joint Conference on Neural Networks (IJCNN)*. Beijing, China: IEEE.

$$(L, c_1) \Rightarrow c_2$$
$$(L, c_2) \Rightarrow c_1$$

3.3.2 Matching Structures to Templates

Given some template, actually finding a match to that template is a nontrivial algorithmic problem. In order to avoid some issues that have been raised by the Tailorability Concern (Licato, Bringsjord, & Govindarajulu, 2013), an algorithm must be designed that must work with extremely large data sets. With this in mind, the algorithm we chose is designed to be in-place and localized—not in the sense of localist concepts we discussed earlier, but rather as the opposite of global, meaning that after the algorithm is given a set of chunks as input, the algorithm only searches chunks in the vicinity of the given chunks. A globally optimal solution is not needed, or even necessarily desirable, in a project which strives primarily for psychological and neurobiological plausibility.

That being said, it would seem that a neurobiologically plausible algorithm would take advantage of the massive parallelism of the brain. For this reason, we explored the use of an Ant Colony Optimization (**ACO**) algorithm based on (Sammoud, Solnon, & Ghédira, 2005). ACO algorithms are examples of meta-heuristics, which are used in hard computational optimization problems such as these when a "good enough" solution is needed (Dorigo & Blum, 2005).

The following will be a strictly high-level description of this algorithm. The algorithm is given a a set of target chunks TC and a completed template TMP, which consists of a template chunk, a set of chunks placed into knowledge structures, and all the connecting ARs. (This paper does not address how such completed templates arise in the first place and are modified over time.) The form of this input is notably different from most other models of analogy, which often take predefined source and target structures that are already complete. Instead of a source structure we have a template, and instead of a target structure we simply have a collection of target chunks which may or may not already be structured. The target chunks are ideally a reflective sample of which concepts are currently active in the reasoner's mind.

The algorithm consists of four main routines, which are controlled by two other subsystems of CLARION: the **ACS** (Action-Centered Subsystem) and the **MCS** (Meta-Cognitive Subsystem). The first routine recruits chunks to fill out the target. The second organizes the chunks in the target and template. Next, the third routine actually performs the mapping using an ACO algorithm. Finally, some chunks may be transferred on to the target chunks. We will now briefly describe each routine in turn, by using the example illustrated in Figure 3.3.

3.3.2.1 Recruiting of Target Chunks

The target chunks TC are supposed to be representative of the chunks that have the highest activation levels at some given moment in the NACS. This can be interpreted as being the concepts in the *foreground* of the reasoner's mind. (This differs from most models of analogy which come with fully structured target analogs as input.) Needless to say, it is possible that the chunks provided to our algorithm as input are insufficient to draw a proper mapping to the provided template, and so this first subroutine of the algorithm attempts to fill out the target structure by activating the chunks in TC and calling chunks from memory that become activated.

3.3.2.2 Organization of Chunks

Now that we have completed template and target structures, we organize the chunks in TMP and TC into levels, such that all chunks are at the highest levels possible without being on the same level or on a higher level than their parent chunks. The bottom, or lowest, levels are considered to be the 'object levels', and the mapping will be made with the assumption that the two object levels will be mapped to each other, and the same for each level above that. For our example, the template structure would consist of two linked and blank chunks on the object level, and a Man and $Mortal$ chunk on the next level up. For the target structure, there is a *socrates* chunk on the object level, and a Man chunk on the next level up.

3.3.2.3 Mapping

An ACO algorithm is next used to find a mapping between the chunks. We first start by drawing temporary 'eligibility' links between chunks. For each pair of levels starting from the object levels, an eligibility link is drawn between every pair of chunks (c_1, c_2) if c_1 is in the template's object level, c_2 is in the target's object level, and the similarity level between c_1 and c_2 is above some tolerance. Blank chunks automatically have eligibility links drawn to every chunk in the corresponding level of the target. Every ant will start with a copy of this list of eligibility links and, as they decide which of these links to add to their mapping, will remove some of these eligibility links from their own copies. In our example (Figure 3.3), both blank chunks on the object level would have eligibility links to the *socrates* chunk, and the two Man chunks would also have eligibility links.

Each ant starts at the object level and selects pairs of chunks from the eligibility links probabilistically, making choices based on several heuristics that either directly or indirectly increase the total match quality, again following Sammoud et al. (2005):

- (Lookahead criteria) Does the candidate pair have parents which are in the eligibility links?
- (Score contribution criteria) Do they have children that are already paired? Do the ARs connecting them to these children use the same CDCs?
- (Pheromone) Check the pheromone attached to this choice, but only if this is the very first choice being made by this ant.

With every choice that is made, eligibility links on the same and higher levels may no longer be valid (they may, for example, violate structural constraints), and so they are temporarily removed before the next choice is made. At the end of each group of ants, the ant with the best match score (which is a function of the number of pairs in the mapping) is compared to the current best score. If the ant's score is better, then pheromone is deposited on each pair in that ant's mapping.

Each group of ants and a single deposit of pheromone constitutes a single iteration. After a certain number of iterations, the best mapping is returned.

3.3.2.4 Transfer

The best mapping score that is found (s) is then divided by the theoretical maximum score (s_{max}). If this amount is greater than a certain tolerance t (usually 0.8), then a bottom-up search is made for chunks in the template that were not mapped to anything. If that chunk's weight within the template w is such that $\frac{s-w}{s_{max}} \geq t$, then a copy of the chunk is made and can be transferred to the target, and any necessary CDC-related ARs are created. s is set to $s - w$, and the process is repeated. In Figure 3.3, we would have the two Man nodes mapped together, and the leftmost blank chunk would be mapped to the $socrates$ chunk. The remaining two chunks ($Mortal$ and the blank chunk below it) would be transferred to the target as $Mortal(socrates)$.

As can be seen in Figure 3.3, deductive reasoning is performed by having templates with blank chunks for quantified variables, and zero-weighted chunks for the consequent chunks. This way if a sufficient match is found for the antecedent (the non-zero-weighted chunks), then the consequent (the zero-weighted chunks) are automatically created, representing an inference.

Analogical reasoning, as in Figure 3.4, requires an extra step that first collects source chunks using a similar process to the *"Recruiting of Target Chunks"* step described above, and then tries (in parallel) different transformations of the source chunks into templates. The algorithm used in deductive reasoning can then be used. Full details will be described in a full-length journal paper.

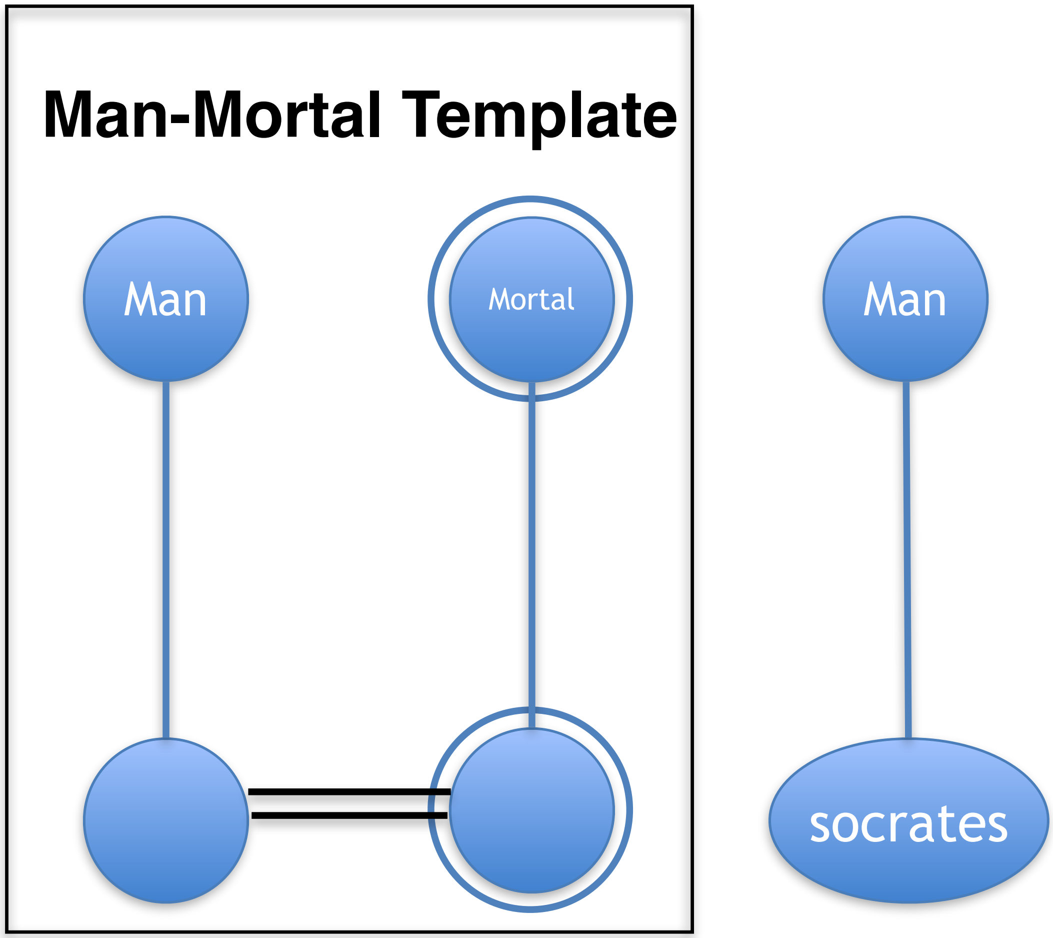

Figure 3.3: Template and target used for the deductive reasoning example in Formula 3.3. This figure previously appeared in Licato, J., Sun, R., & Bringsjord, S. (2014). Structural Representation and Reasoning in a Hybrid Cognitive Architecture. In *Proceedings of the 2014 International Joint Conference on Neural Networks (IJCNN)*. Beijing, China: IEEE.

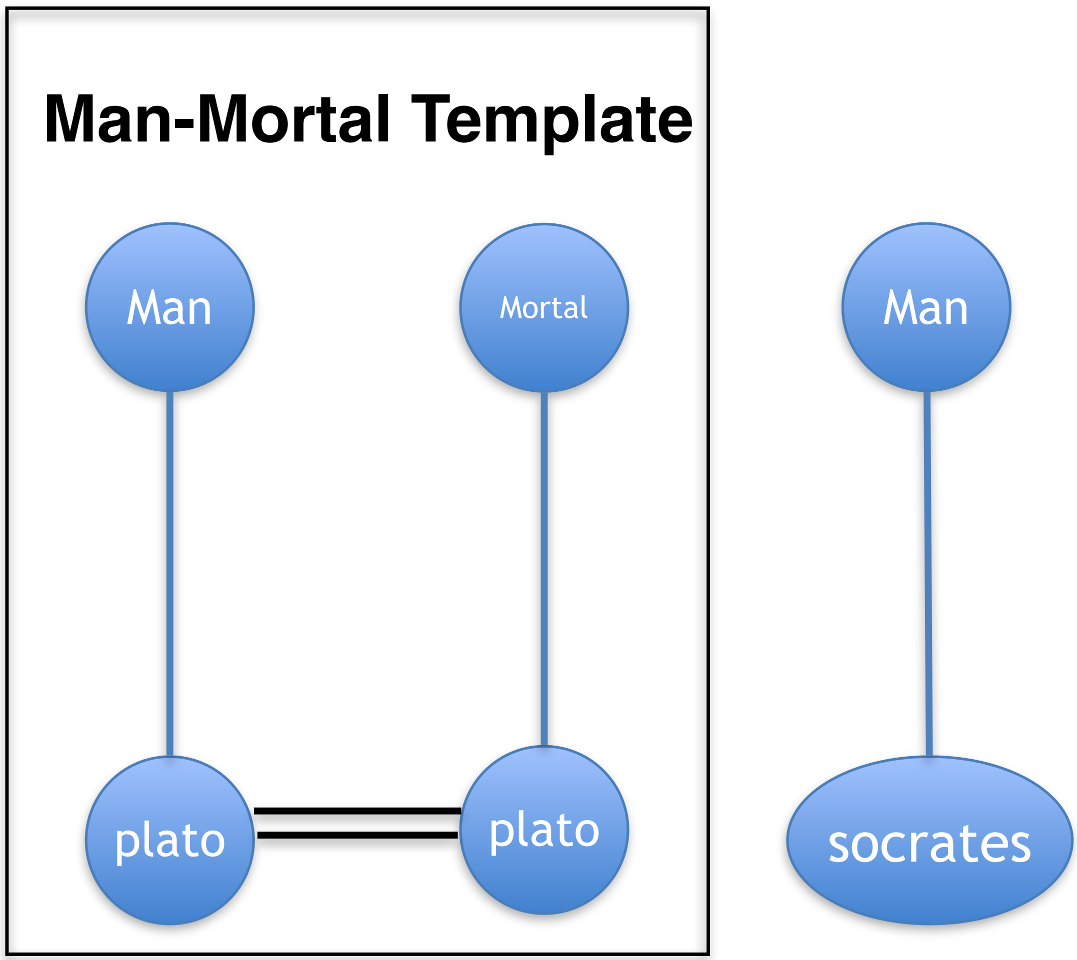

Figure 3.4: Template and target used for the analogical reasoning example in Formula 3.4. This figure previously appeared in Licato, J., Sun, R., & Bringsjord, S. (2014). Structural Representation and Reasoning in a Hybrid Cognitive Architecture. In *Proceedings of the 2014 International Joint Conference on Neural Networks (IJCNN)*. Beijing, China: IEEE.

CHAPTER 4
PAGI World

This chapter describes PAGI World, a simulation environment I developed so that an artificial agent could construct and reason over representations that are semantically rich. PAGI World offers a number of nice features that can be beneficial to the AI and cognitive modeling communities in general, and I summarize these features in this chapter. I will start by outlining the motivations for PAGI World, introduce technical details about its implementation, compare PAGI World to other simulation environments, and conclude by discussing future plans for this project.

4.1 Introduction

Frank Guerin (2011), in his recent survey of the emerging field of Developmental AI, concluded that current systems were lacking in several key areas. Guerin then suggested that a major reason (arguably the most important) why the field has the shortcomings he described, was the absence of a suitable simulation environment. Current simulation environments used by Developmental-AI projects were missing several key features, and Guerin described some conditions that would need to be met by simulation environments in order to cure this problem. Among the most important of these will be referred to as **C1**, **C2**, and **C3**:

C1 It is rich enough to provide knowledge that would bootstrap the understandings of concepts rooted in physical relationships; e.g.: inside vs. outside, large vs. strong, etc.

C2 It can allow for the modeling and acquisition of spatial knowledge, which Guerin notes is widely regarded to be a foundational domain of knowledge acquisition, through interaction with the world.

C3 It can support the creation and maintenance of knowledge which the agent can verify itself (This is Sutton's (2006) **Verification Principle**, to be elaborated shortly).

Though these points are extremely important, there are a few more that should be added to this list that may be worth noting as well:

C4 It is rich enough to provide much of the sensory-level information an agent in the real world would have access to.

C5 It can allow for testing of a virtually unlimited variety of tasks, whether these are tasks testing low-level implicit knowledge, high-level explicit knowledge, or any of the other areas required by Psychometric Artificial General Intelligence (**PAGI**). Ideally, such a system would support the easy creation of new tasks and environments without requiring a massive programming effort.

C6 It provides pragmatic features that enable tasks to be attempted by AI coming from researchers using different types of systems, and different theoretical approaches, thus enabling these different approaches to be directly compared with each other.

This chapter describes a task-centered, physically realistic simulation environment that I have developed to simultaneously address challenges **C1-C6**. It is hoped that this system[13] can remove a sizable roadblock in the fields of developmental AI and cognitive modeling in general. In Section 4.2, I will elaborate on and defend conditions **C1-C6**, and then explain the concepts of PAI and PAGI.

4.2 Guerin's Conditions

A common theme running through conditions **C1-C3** is that what is lacking from current microworlds is a physically realistic environment—one in which the agent can acquire, develop, and test its concepts. But the concerns raised by Guerin (2011) are not only of interest to the field of Developmental AI; in point of fact, AI in general can benefit by addressing them. For example, feature **C1** is extremely important for cognitive models of analogy, which are currently struggling to overcome what has been called the *Tailorability Concern* (TC)(Gentner & Forbus,

[13]In this paper, I will at times use the words 'system' and 'environment' to describe simulators. These words connote slightly different aspects—the former highlights the microworld the simulator creates for AI agents, the latter emphasizes the total software package provided by the simulator.

2011; Licato, Bringsjord, & Govindarajulu, 2013). TC, in essence, is the concern that models of analogy (though this can be applied to all cognitive architectures in general) have far too long dealt almost exclusively with manually constructed knowledge representations, using toy examples often selected solely to display some particular ability. Licato et al. (2013) goes on to argue that overcoming TC is necessary to advance the fields of analogy and cognitive architectures. After making this point, they develop a set of conditions that must be met in order to claim victory over TC:

> $\mathbf{TCA_3}$ A computational system of analogy answers TC if and only if given no more than either
>
> - unstructured textural and/or visual data, or
> - a large, pre-existing database,
>
> and minimal input, it is able to consistently produce *useful* analogies and demonstrate stability through a variety of input forms and domains.

According to $\mathbf{TCA_3}$[14], then, good performance on the part of a cognitive agent on a sufficiently large knowledge base from which source analogs could be drawn is required to answer TC. An agent interacting in the sort of microworld called for by Guerin (2011) might ideally be able to acquire such source analogs by simply interacting with its environment.

C1 and TC together require that the microworld itself is what provides the knowledge drawn upon to construct concepts of basic physical relationships, not manually constructed source analogs or fully explicit logical theories. **C2** expands on **C1** by requiring that this knowledge of physical relationships not be static, but rather should allow for an agent in the world to learn through interaction. The idea that children learn by initiating interactions with the world based on their (often incomplete) conceptions of reality—in a manner that resembles scientific experimentation—was championed by Piaget and later constructivists (Glasersfeld, 1991; Quartz & Sejnowski, 1997; Piaget et al., 2001), and is at the core of contemporary views like Bickhard's Interactivist Model (Bickhard, 2008).

[14] In Licato et al. (2013), $\mathbf{TCA_1}$ and $\mathbf{TCA_2}$ are also formulated and discussed, but ultimately rejected in favor of $\mathbf{TCA_3}$ and $\mathbf{TCA_4}$.

A microworld that satisfies **C2**, then, should recognize that the concepts, schemas, and representations used by the agents will frequently change, and this is difficult to do if the information provided from the microworld to the agents within it are frozen in representation. There are many ways in which information might be frozen in representation: It may make use of labeled concepts which are too high-level and do not change, or it may be too rigid in its form of presentation. Consider, for example, the Event Calculus (Kowalski & Sergot, 1986), which defines the event, a fixed set of predicates and objects, and certain inference steps as primitives. The DORA model, which attempts to explain concept acquisition and construction at the neurobiological level, also (at present) requires primitive conceptual constructs that allow for pairwise comparison of analog values (Doumas et al., 2008; Doumas & Hummel, 2013). **C2** helps to ensure that microworlds make as few assumptions about which of these primitives are required as possible.

Note that this is to say nothing about what mode of representation the agent is better off using. It may benefit the agent to use fully top-down approaches (Bringsjord, 2008a, 2008b), or perhaps hybrid representations (Sun, 2002), depending on the task being solved and the purpose of the demonstration. Some inflexibility may be unavoidable: every form of representation has some set of primitive constructs at its core, whether they are logical operators, undefined microfeatures, or a mostly fixed set of sensory inputs and outputs. PAGI World simply tries to provide a lower granularity for these primitive constructs, with the assumption that a powerful AGI system can eventually come along and construct all of the other higher-level constructs with them. Requiring the information directly given to the agent's sensors to be as low-level as possible is PAGI World's way of ensuring that it does not provide its inhabitants (the agent controlled by AI scripts) with knowledge based on frozen representations, and is the motivation for **C4**. Whatever the AI agent decides to do with that low-level knowledge—whether it extracts higher-level symbols and performs reasoning over them or not—is completely up to the writers of the AI scripts.

In contrast with **C1** and **C2**, **C3** comes closest to placing a requirement on the type of knowledge and cognitive operations on this knowledge that the agents in the microworld have. However, it goes just short of that by not requiring that the agents actually *do* maintain and verify their own knowledge. Rather, it specifies that the world be rich enough to allow such an agent to exist.

C3 requires further elaboration. Guerin (2011) cites Sutton's (2006) **Verification Principle** (VP), which states:

> An AI system can create and maintain knowledge only to the extent that it can verify that knowledge itself (Sutton, 2006).

The VP, Sutton argues, is extremely important in AI systems which aim to acquire and develop knowledge that will eventually become too large or numerous for humans to independently verify. Just like the commonsense reasoner who is reluctant to accept revelations of knowledge that do not 'make sense' to the reasoner, AI and AGI systems would need some way discriminating way to consider ideas. Because PAGI World only provides its agents with low-level information that accurately describes the state of the microworld and avoids providing interpretations of this information, an agent that satisfies VP is entirely possible. However, it would be specific to the tasks to determine how to measure adherence to VP.

There are many ways in which an AI might independently verify knowledge which it creates or is seeking to maintain. One such way is reminiscent of the hypothetico-deductive method, that is, it is reasoned that the knowledge to be verified, if true, would imply some hypothesis h. If h states that some observation must hold or that some sensory data should be expected, then the agent should be able to verify h by looking for itself. **C3**, then, requires that simulation environments be rich enough to allow the agent to do just that, while placing as little restriction as possible on the types of hypotheses h that the agent can check for. For example, the agent may want to check for low-level sensory data, or it might want to perform detailed experiments (as did the children in many Piagetian experiments). PAGI World offers this capability.

4.2.1 PAI and PAGI

Following $\mathbf{TCA_3}$, another formulation of the Tailorability Concern and recommendation for how to surpass it was also presented in Licato et al. (2013):

> $\mathbf{TCA_4}$ A computational system $\mathcal{A}$ for analogy generation answers TC if and only if, given as input no more than either
>
> - unstructured textual and/or visual data, or
> - a vast, pre-existing database not significantly pre-engineered ahead of time by humans for any particular tests of $\mathcal{A}$,
>
> is — in keeping with aforementioned *Psychometric AI* — able to consistently generate analogies that enable $\mathcal{A}$ to perform *provably well* on precisely defined tests of cognitive ability and skill.

$\mathbf{TCA_4}$ ties TC to Artificial General Intelligence (AGI) by introducing the concept of **Psychometric AI** (PAI) (Bringsjord, 2011; Bringsjord & Schimanski, 2003). PAI sees good performance on well-established tests of intelligence as a solid indicator of progress in AI. Some may note that most intelligence tests fail to capture human-level skills such as creativity and real-time problem solving; therefore, related to PAI is **Psychometric Artificial *General* Intelligence** (PAGI) (Bringsjord & Licato, 2012). For example, one test of PAGI is Bringsjord and Licato's (2012) **Piaget-MacGyver Room**, in which an agent is inside a room with certain items and a task to be performed. The agent must achieve the task using some combination of the items in the room (or using none of them, if possible). Depending on the task, the solutions may require using the items in unusual ways, as viewers of the MacGyver television series may remember.[15]

If PAGI tasks are meant to subsume all tasks solvable by neurobiologically normal human adults, then a simulation environment designed to capture PAGI tasks should also be able to test for two types of knowledge humans make regular use of: explicit and implicit knowledge. The implicit/explicit distinction (Sun, 2002), which roughly parallels the System 1/System 2 distinction of Kahneman (Kahneman, 2011) (but see (Sun, 2014) for a criticism of System 1 vs. 2), encompasses an extremely broad spectrum of explanations for human phenomena (Sun,

[15]Note that although I have adopted "PAGI World" as the name of this simulation environment in order to reflect the fact that it is designed to support many types of PAGI tests (including variants of the Piaget-MacGyver Room, as I describe below), PAI tests are just as easily implementable in PAGI World.

2001, 2002, 2004). If a simulation environment restricts itself to AI controllers that rely on explicit or implicit processes exclusively, then it cannot hope to capture the breadth of tasks required to qualify a Psychometric Artificial *General* Intelligence. For these reasons, **C5** occupies an important place in our list.

C5, however, also contains a warning rooted in pragmatics. Although qualitative breadth is important, the downfall of a simulation environment may also be *quantitative* shortcomings. For example, the tasks developed for such an environment may all be solved, or uninteresting to researchers, thereby motivating the creation of a new simulator. PAGI World's solution to this is to provide a task editor/creator that is extremely easy to use with minimal training; I describe it in Section 4.3.1.2.

4.3 PAGI World

Requirement **C6** is the most practicality-oriented, reflecting both Guerin's (2011) inclination (shared by myself) to believe that an effective way to compare AI and AGI methodologies would be to see how they perform on the same tasks, implemented on the same systems. But few such tasks and systems exist, and therefore before describing PAGI World, it may be helpful to take a step back and look at this project in a broader view.

A wider perspective, after all, may help to understand the need for a system such as PAGI World. Science advances, in no small part, by having an array of increasingly powerful tools available to scientists, and by continually improving the infrastructure available to those scientists for the purposes of testing, analyzing, and comparing their ideas. If the state of lens manufacturing in Europe was not as well developed as it was by the time of Galileo, he would not have been able to develop the telescope and make his discoveries. To use a more contemporary example, it is difficult to see how modern advances in machine learning (and by extension modern search engines) could have progressed so quickly without the parallel development of computers fast enough to run those algorithms. Similarly, AI and AGI can only benefit from having a physically realistic, easy-to-use simulation environment like PAGI World.

Given its potential benefit to the field as a whole, why does such a simulation not currently exist, and do any of the roadblocks currently in the way affect the plausibility of the current project?

4.3.1 Why Isn't Such a System Already Available?

4.3.1.1 Technical Difficulties

One potential roadblock is obvious: programming a realistic physics simulation is *hard*. Some of this difficulty is reduced by working with a 2D, rather than a 3D, environment. Although some software libraries have previously been available for 2D physics simulations, they have often been very language-specific and somewhat difficult to configure.

Secondly, even if one were to stick with a 2D physics library and commit to it, substantial development resources would be needed to enable the resulting simulation to run on more than one major operating system. Furthermore, even if *that* problem is somehow addressed, there is a vast diversity of languages that AI researchers prefer to use: Python, LISP (in various dialects, each with their own passionate proponents), C++, etc. All of these technical issues tend to reduce how willing researchers are to adopt particular simulation environments.

Fortunately, all of the above problems can be solved with a single design choice. Unity, a free game-development engine, has recently released a 2D feature set, which comes with a 2D physics model that is extremely easy to work with. In fact, the blog post making the announcement of the 2D feature set was dated November 12, 2013. Furthermore, Unity allows for simultaneous compilation to all major operating systems, so that developers only have to write one version of the program, and it is trivial to release versions for Mac OS, Windows, and Linux. Because Unity produces self-contained executables, very little to no setup is required by the end users.

Finally, because Unity allows scripting in C#, an interface was designed for AI systems that communicates with PAGI World through TCP/IP sockets. This means that AI scripts can be written in *virtually any* programming language that supports port communication.

4.3.1.2 Theoretical Difficulties

Unity conveniently helps to remove many of the technical roadblocks that have previously blocked the development of simulation environments that can be widely adopted. But there are also theoretical roadblocks; these are problems with the generality vs. work-required tradeoff. For example, if a simulation environment is too specifically tailored to a certain task, then not only can systems eventually be written to achieve that particular task and nothing else, but the simulation environment quickly becomes less useful once the task is solved. On the other hand, if the system is too general (e.g. if a researcher decides to start from scratch with nothing but Unity), then the researcher must devote too much time and energy to developing a new simulation environment for each project, rather than spending time on the AI itself.

PAGI World was designed with this tradeoff in mind. A **task** in PAGI World might be thought of as a Piaget-MacGyver Room with a configuration of objects. Users can, at run-time, open an object menu (Figure 4.1) and select from a variety of pre-defined world objects, such as walls made of different materials (and thus different weights, temperatures, and friction coefficients), smaller objects like food or poisonous items, functional items like buttons, water dispensers, switches, and more. The list of available world objects will frequently be expanding and new world objects will be importable into tasks without having to recreate tasks with each update. Perhaps most importantly, tasks can be saved and loaded, so that as new PAI/PAGI experiments are designed, new tasks can be created by anyone.

4.3.1.3 Problems with Robotics Environments

There have been some attempts to create physically realistic simulation environments for AI researchers. However, as will be argued, they have largely failed to address the concerns in (Guerin, 2011) and (Licato, Bringsjord, & Govindarajulu, 2013). I now briefly discuss other systems that have goals similar to those of PAGI world, in order to emphasize the uniqueness of this project. What follows is not intended to be a comprehensive survey of the field, but I have tried to provide a representative sample of what is currently available.

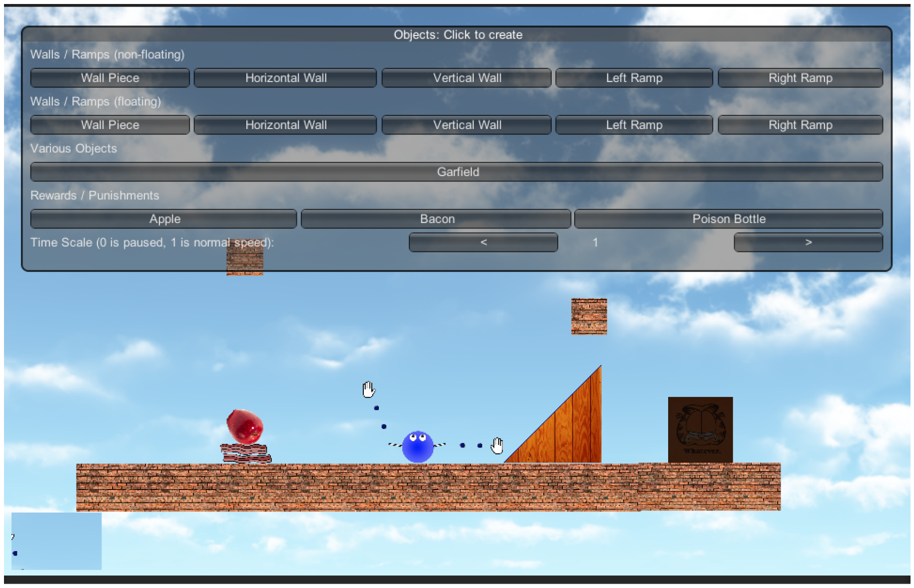

Figure 4.1: PAGI World With the Object Menu Visible

The subfields of developmental robotics (Lungarella, Metta, Pfeifer, & Sandini, 2003) and cognitive developmental robotics (Asada et al., 2009) have long offered an option for researchers interested in demonstrating the effectiveness of a certain AI theory when placed in a real-time test environment. But as researchers know, robotics research can be quite costly, and working with hardware can introduce a steep learning curve that some may want to avoid if possible. Robot software simulators offer a compromise. Thus, for this and other reasons, the number of robotics simulation software options has been increasing quickly over the past few years. Not surprisingly, several surveys and reviews of the software alternatives have emerged to make sense of the growing landscape.

Although it must be emphasized that PAGI World is not in any way intended to be a tool for robotics development, there is a clear overlap in interest with robotics simulation environments, and thus a quick review of the relevant literature is appropriate here. Some robotics researchers have already realized the value of using computer games as testbeds for human-level AI (Laird & Lent, 2001), and more specifically the ability for Unity to simulate worlds for testing robotics, and have created basic frameworks for modeling robot sensors and kinematics (Hernandez-Belmonte, Ayala-Ramirez, & Sanchez-Yanez, 2011; Mattingly et al., 2012). But although the simulators designed for robotics are already popular and enjoy widespread support, they may be too focused on the specifics of robot hardware for our PAI- and PAGI-oriented purposes. For example, of particular concern to robotics researchers are environments that allow them to simulate robot locomotion, grasping, joint dampening (Drumwright, Hsu, Koening, & Shell, 2010), hardware support, robot configuration methods (Kramer & Scheutz, 2007), accuracy of contact resolution, and having the same interface between the simulated and actual robot control systems (Ivaldi, Padois, & Nori, 2014). Furthermore, the sorts of PAI/PAGI tasks PAGI World focuses on are not currently available.

I propose to distinguish PAGI World from the current crop of robotics simulators on several key points:

- As many as possible of the low-level details of hardware implementation are abstracted, so that the AI researcher can focus on cognitive-level problems.

- Although low-level sensory information about the world will be available, information will optionally be available at a slightly higher level of abstraction as well, e.g.: object names, locations, etc.

- An easy-to-use system will be in place for *quickly* creating new PAI/PAGI tasks, so that anyone without programming experience can create them and share them with others. This helps to ensure that the amount of tasks available will continue to increase, and that therefore time spent developing AI systems to work with PAGI World will be re-usable.

- PAGI World's simulator is tied to Unity 2D. This connects PAGI World to an extremely stable physics and graphics engine that enjoys widespread community support and is rapidly being upgraded. This provides a reasonable degree of confidence that any bugs with the physics simulator or other engine-level components will be addressed quickly.

In short, PAGI World is PAI/PAGI task-oriented, and targeted to cognitive-level researchers modeling both high- and low-level tasks, rather than to robotics researchers.

4.3.1.4 Other Simulation Environments

There have been some notable attempts to provide simulation environments for AI systems, particularly those inspired by the Developmental AI approach. For example, Bruce (2010) created a Developmental AI testbed by updating an older version created by Frank Guerin (Bruce, 2010).

Although some of the present paper's authors are sympathetic to the power of Piagetian schemas and the AI systems derived from Piaget's theories, Bruce's (2010) system is tightly coupled with a particular cognitive architecture (presented in the same paper) that uses schema-based AI systems, whereas PAGI World is agnostic about what AI approach is used. It is unclear how easy or difficult it would be to adapt arbitrary cognitive architectures to work with their simulation environment.

They used the JBox2D library for their physics engine, which, according to (Bruce, 2010), was poorly documented and difficult to work with (e.g., implementing a method to detect when the robot hand touched an object took markedly longer than they planned due to a lack of documentation for JBox2D). Although a newer version of JBox2D became available afterward, implementing the new version requires the simulation programmer to manually update the relevant code, whereas updates to the Unity 2D physics engine will automatically be propagated to PAGI World, with minimal to zero code changes. In other words, upgrades to PAGI World's physics engine will be virtually transparent to AI developers.

(Drescher, 1991) proposed an early microworld in which an agent, whose development made use of a primitive form of Piagetian schemas, explored the world and learned about the objects with which it interacted. Although this was a promising start, after this initial start it was not developed further, nor was any significant effort made by other researchers to pick up on Drescher's work, as far as I am aware (merely one small-scale re-implementation of Drescher's work exists, e.g. (Chaput, Kuipers, & Miikkulainen, 2003)). Nevertheless, Drescher's microworld has some very interesting elements that have been taken as a starting point for PAGI World, and those starting points will be described next in detail.

4.3.1.5 Drescher's Simulation

Drescher's (1991) microworld simulation environment was created primarily to test his Piagetian artificial agent, which took low-level information about the environment and constructed increasingly complex schemas about its microworld. The microworld consists of a 2D scene divided into a grid that limits the granularity of all other elements in the microworld. Inside this microworld are objects that take up discrete areas of the grid and contain visual and tactile properties. These properties can be described by vectors which, for the purposes of this simulation, have arbitrarily chosen values.

Most importantly, the microworld contains a single robot-like agent with a single hand that can move in a 3-cell $\times$ 3-cell region relative to the part of the robot's body considered to be its 'eye.' If the hand object is adjacent to an object in the world (including the robot's own body), a four-dimensional vector containing tactile information is returned to the agent. The body has tactile sensors as well, though they do not return tactile information as detailed as that returned by the tactile sensors of the hand.

Visual information is available as well, in the form of a visual field whose position is defined relative to the robot's body. A smaller region within the visual field, called the **foveal region**, represents the area within the visual field where the robot is currently looking. The foveal region returns vectors representing visual information, and the cells in the visual field not in the foveal region also return visual information, but with lower detail.

Perhaps one of the most interesting features of Drescher's microworld is the fact that the robot can only interact directly with the world by sending a set of pre-defined "built-in actions." Although the internal schema mechanism of the robot may learn to represent actions as richer and more complicated, ultimately what is sent to the simulation environment is always extremely low-level. Likewise, the information provided to the robot is always extremely low-level. The task of identifying and naming objects in the world—and even of knowing that objects in the world consistently exist!—is up to the learning mechanism the robot utilizes.

The fact that the learning and control system of the artificial agent can be developed almost completely independently of the features of the world itself, is one of the primary reasons why Drescher's microworld is appealing as a starting point for PAGI World. Drescher's microworld is mostly in line with the theoretical assumptions made here, which will be described further in the next section. But first, the primary areas in which PAGI World departs from, and has innovated beyond, Drescher's microworld are as follows:

- **Agnosticism re. the AI method used**. Whereas Drescher's microworld was created for the sole purpose of testing his Piagetian schema-learning mechanism, I have designed the world, program, and interfaces so that as wide a variety as possible of AI techniques can be productively and easily used.

- **Optional mid-level input**. Related to the previous point, clearly some researchers simply won't want to translate vector input for every piece of tactile or visual information they come across. PAGI World offers the option for the agent to directly receive the name of the object upon touching or viewing it.

- **Granularity**. The granularity of information in PAGI World is dramatically finer; consider the increase in size of the visual field: Drescher's was an area of 7-$\times$-7 cells with one visual sensor per cell. The visual area has been upgraded to span a 450-$\times$-300 unit area, with each visual sensor spaced 15 units from its nearest neighbor (each unit roughly corresponds to a screen pixel).

- **Hands**. The robot is given two hands instead of one, each with a similar range of motion, but with different distances (relative to the body) that each can reach. Although the simulation world is 2D, the hands exist on a separate layer that floats 'above' objects in the world, analogously to a mouse cursor in any major operating system. The hands can grip and move objects they are floating over (just like how one might click and drag an object in Windows or MacOS), provided the objects are not too heavy or otherwise held down.

- **Realistic Physics**. Undoubtedly the most important improvement introduced is the aforementioned realistic physics provided by Unity 2D.

- **Focus on a wide breadth of tasks**. Although Drescher's microworld was a start in the right direction, we feel that it did not make enough of a push to be considered a simulation environment for AGI tasks, nor did it explicitly set out to be a testbed for the sort of tasks prescribed by Psychometric AI.

4.3.2 Reflexes, DFAs, and the Implicit vs. Explicit Distinction

Although communication through TCP/IP ports is relatively quick, and the command system created is designed to be efficient, there are some actions requiring extremely rapid, simple checks and responses. For example, holding an object in the air at a certain position relative to the body for an extended period of time may require many quick corrections. If the object starts to move down, more upward force should be applied. But if it moves too far up, downward force should be applied (or the amount of upward force should be reduced). In order to old the object as still as possible, the amount of force applied would be based on its current and projected velocity and position. However, if the AI script requests this information, does a calculation to determine the amount of correction required, and sends back the command to adjust the amount of force, by the time this command is received by PAGI World and processed it may be inaccurate.

PAGI World fixes this problem by implementing *states* and *reflexes*. Reflexes and states can be set and modified through commands from the AI script, but they are actually checked and executed completely on the PAGI World side, which allows for much faster reaction times. A reflex r consists of a tuple $(\mathbf{C}, \mathbf{A})$, where $\mathbf{C}$ is a list of conditions and $\mathbf{A}$ is a list of actions. Each condition in $\mathbf{C}$ must be satisfied in order for reflex r to activate. These conditions can consist of sensory inequalities, e.g.: whether one of the tactile sensors detects a temperature above a certain amount, or whether the AI agent's body is moving above a certain velocity. If all of the conditions are met, then the actions are executed immediately. Furthermore, sensory inequalities can be specified as simple arithmetic functions of sensory values, so that a reflex can be fired if (to cite an arbitrary example) the horizontal component of the agent's body's velocity is at least twice the value of the vertical component of its velocity.

States can be activated and checked by reflexes. Essentially, this means that multiple deterministic finite automata (DFAs) can be stored and executed completely on the PAGI World side. However, the expressivity of the conditions and actions within each reflex strictly restricts the system so that full Turing machines cannot be implemented on the PAGI world side. This allows developers to implement two important categories of abilities generally regarded to be part of the human experience: explicit, and implicit. Recall that the explicit vs. implicit distinction divides the mind into explicit processes which are generally slow, deliberate, and easy to verbalize, and implicit processes which are mostly quick, automatic, and not easily accessible to the conscious mind (Sun, 2002).

Although PAGI World does not support all imaginable implicit processes (for example, some might believe that a Bayesian probabilistic approach or a deep learning artificial neural network is necessary to implement some implicit processes), the fact that multiple DFAs can be stored and executed in PAGI World's optimized code gives the user a flexibility to capture a wide range of implicit processes.

4.3.3 The Architecture of a PAGI World Setup

Figure 4.2 pictures the architecture of a typical PAGI World + AI controller pairing. As the figure illustrates, it is helpful to think of the processes controlled by the PAGI World application to be the PAGI-side, as opposed to the side which can be completely implemented externally, referred to as the AI-side. The reflex and state machine described in Section 4.3.2 is controlled and managed on the PAGI-side, but both states and reflexes can be dynamically modified through commands sent by the AI-side.

All commands going from the AI-side to the PAGI-side, and all sensory information passing in the other direction, is done through messages communicated through TCP/IP ports. Therefore, the AI-side can be written in any programming language which supports the creation and decoding of strings over TCP/IP. Although this flexibility sets PAGI World apart from many other alternatives, some may prefer an additional level of abstraction on the AI-side, and for this reason PAGI World will provide, and we are continuing development on, a Python library

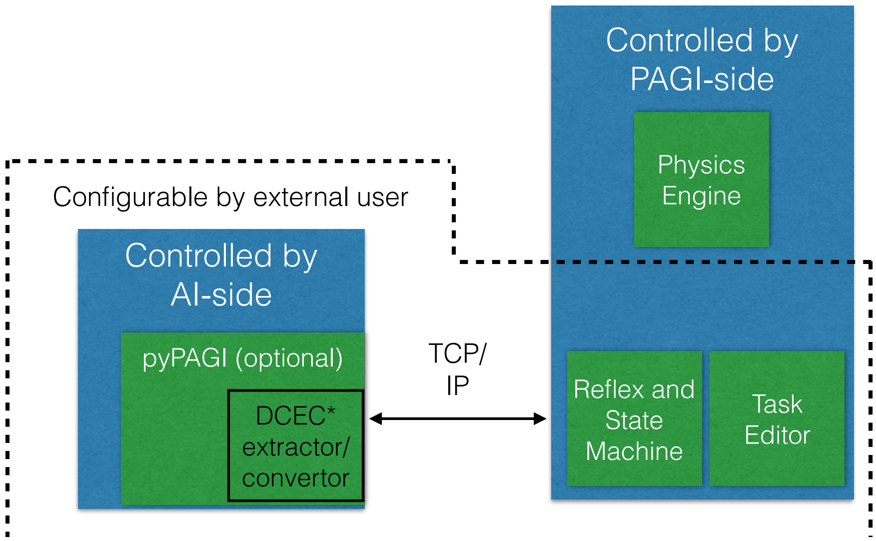

Figure 4.2: The architecture of an instance of PAGI World and an AI controller. Everything on the AI-side can be written by AI researchers, as the interface with the PAGI-side is handled through messages passed over TCP/IP sockets. A Python library, called pyPAGI, is also optionally available to assist researchers with common AI-side functionality, including encoding of PAGI World knowledge in the Deontic Cognitive Event Calculus ($\mathcal{DCEC}^*$). The reflex/state machine and task editors can also be controlled through TCP/IP, though the task editor is additionally available through a WYSIWYG drag-and-drop interface.

called *pyPAGI*, which will be described in the next section.

Tasks can be created, saved, and loaded using the GUI editor at run-time (Figure 4.1), but as suggested by Figure 4.2, they can also be somewhat configured by AI-side commands. This can be useful to modify the layout of the task dynamically in response to actions the AI agent takes (e.g. making an apple appear as a reward, or a bottle of poison as a punishment), or to load new tasks after successful task completion for automated batch processing of tasks.

4.3.4 pyPAGI: An AI-side Library

The **pyPAGI** library, a Python library which will also be made available, provides users with many of the mid- to high-level commands to control the AI agent. This library may be useful to first-time users of PAGI World, to beginners and students of AI programming, or to researchers not concerned with the methods used to convert high-level commands into the low-level sensory commands PAGI World requires. For example, although commands to apply force to the hand and to grab objects are part of the set of TCP/IP commands, actually picking up an object and holding it still in the air for a specified amount of time requires a series of reflexes to be put in place to allow for the tiny corrections that are necessary to accommodate the object's weight, the momentum of the hand, and any other forces which may prevent the object from being held stationary. The pyPAGI library has support for performing this and many other functions.

pyPAGI also performs bottom-up symbol and concept extraction for those who are not concerned with implementing these common functions themselves. It does this by interpreting sensory shapshots of the world at discrete moments in time, and extrapolating information about the motion, relative layout, and other descriptive features of the world. This information is primarily expressed in the language of the Deontic Cognitive Event Calculus ($\mathcal{DCEC}^*$), which is a highly expressive logical framework for representing nested beliefs, knowledge, deontic facts (obligations), and more (Bringsjord, Clark, & Taylor, 2014).

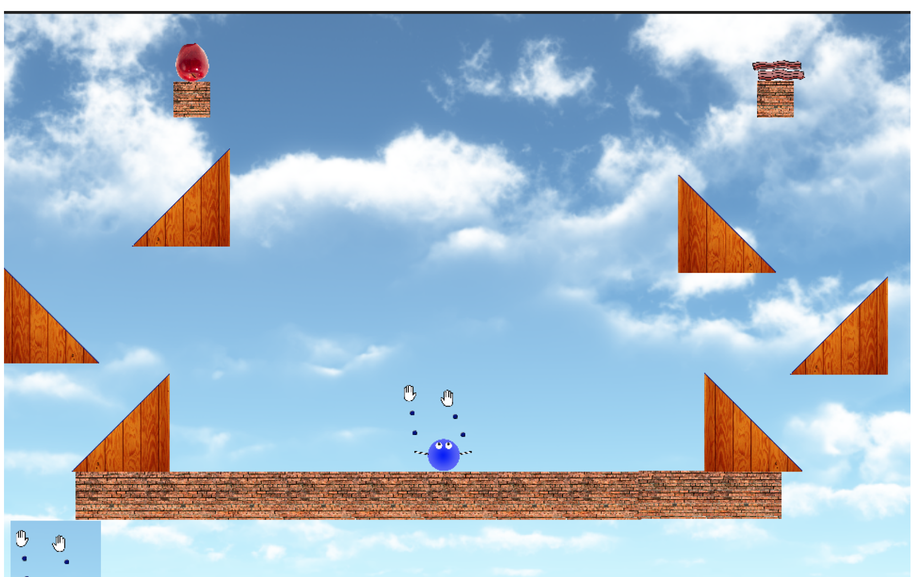

Figure 4.3: A task in which the apple and bacon will fall at the same time, leaving only enough time to save one of them.

PAGI World offers an environment in which the amount of actions that an artificial agent can learn is virtually unlimited. PAGI World's flexibility also allows us to control an agent with the AC notation described in Chapter 3. The stage is therefore set for a demonstration of $\mathcal{AC}$ in action, which will be the focus of the next chapter.

CHAPTER 5
Demonstration and Conclusion

As suggested by this dissertation's title, I want to show how action schemas and analogy can give rise to the ability to reason. In Chapters 1 and 2, I argued that LPT's central concepts of abstraction, cognitive structures rooted in action, and the literature on analogical reasoning suggest an approach for achieving the goal stated in the subtitle, and taking these ideas seriously points to the no-semantically-empty-structure principle (NSES). In Chapter 3, I showed how NSES could be realized in a cognitive architecture. Chapter 4 described a simulation environment which was rich enough to support a semantically rich cognitive agent.

In this final chapter, I will present a short, extremely simple demonstration, that should be considered a starting point for future work. Such future work, as this dissertation has argued, requires an environment capable of supporting the development of knowledge rich in action. PAGI World is such an environment. Therefore, in this demonstration, an agent in PAGI World will use information provided to its sensors by the environment in order to construct representations which take NSES seriously. After describing the demonstration and discussing its strengths and weaknesses, I will conclude the dissertation and discuss future directions of a research program based in $\mathcal{AC}$.

A typical representation of an action instance is a triple (**pre**, *action*, **post**) where **pre** is a set of observables or components of context which hold (or are believed by the agent to hold) prior to *action* taking place, and **post** is a set of observables perceived after the action is taken.

An action is implemented here as a subroutine, or a set of instructions whose execution can be affected by a set of provided parameters. By analogy to programming languages, an action is like a function that can take arguments, but does not return a value. Note the compatibility between this understanding of an action and the definition of action provided in Section 2.4 holds if one accepts that the computer executing the subroutines is a kind of 'mind' (though acceptance of such a claim is not centrally important to our purposes here).

An action by itself is *non-teleological* (does not have a truth value or end goal); any feedback obtained through execution of an action comes in the form of percepts. For example, if an agent performs the action of grabbing an object, the action itself does not return a notification of success. Rather, the agent will receive feedback through its normal senses during the execution of the action (force sensors, touch sensors, etc.) and the agent's judgement of success or failure of the action depends on the match between the agent's observations and the expectations set by whatever goal-setting subsystem initiated the action in the first place.

A **perceptual symbol** is a non-relational mental construction which is constructed out of perceptual inputs, to be identified with the pure observables defined in Section 2.4. The ability to recognize and classify combinations of sensory inputs as perceptual symbols is learned over time, and does not require rooting in action. As a rule of thumb, one might think of a purely perceptual symbol as the sort of thing which can be learned and accurately identified by today's best computer vision systems (e.g. (Socher, Lin, Ng, & Manning, 2011)). A perceptual symbol can be purely constructed out of perceptual inputs, or, in the case of relational perceptual symbols, partially constructed out of actions.

I will define an **observable** as a perceptual symbol, relational or otherwise, paired with an object. An observable may be a *pure* observable (defined in Section 2.4 as those symbols not rooted in action), or it may be a construct partially consisting of action-rooted structures. The observables used in this demonstration are simply given to the agent, so I do not address whether they should be considered pure observables or not.

Observables come in the predicate-object form as described in Chapter 3. Given that observables and objects are so central to the elements in action instances, we are faced with a problem: in PAGI World we are only given sensor values, not direct knowledge about objects! So how do we bridge the gap and generate predicate-object representations from PAGI World's provided sensory data?

This chapter's demonstration will attempt to answer that question, and in doing so demonstrate that PAGI World is a valid system on which an agent completely

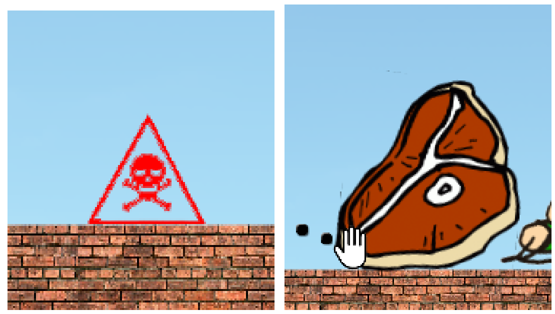

Figure 5.1: A poisoned item (left) and a steak (right), which provide negative and positive endorphins when coming into contact with PAGI guy's body.

Figure 5.2: A set of colored bombs (or dynamite sticks, left) and colored walls (middle). When a bomb comes into contact with a wall piece of the same color, they explode (right) and both disappear.

based in $\mathcal{AC}$ can be hosted. I now turn to the description of the task itself.

5.1 Description of the Task

The agent in PAGI World (referred to as "the agent" for the remainder of this chapter) is placed in an environment (alternately, a specific setup of the PAGI World environment will be referred to as a 'task') and is given a set of perceptual symbols related to its sensors. I will show how how the predicate-object representation described in Chapter 3 can be constructed.

The task is a small area with a series of stationary wall blocks. Randomly scattered throughout the environment are randomized combinations of 8 items:

- Poisoned item (Figure 5.1, left side), which appears as a triangle with a skull and crossbones logo. When this touches PAGI guy's body, a value of -10

endorphins is triggered in the relevant body sensors.

- Steak (Figure 5.1, right side). Like the poisoned item, except it has an endorphin value of +10.
- Green, blue, and red blocks (Figure 5.2, middle). Acts just like the normal brick wall blocks all over the environment, except these can be destroyed by bombs.
- Green, blue, and red bombs (Figure 5.2, left side). Appears (and perhaps better described) as bundled sticks of dynamite. When these come in contact with colored blocks of the same color, they destroy each other, resulting in an explosion (Figure 5.2, right) that is visible to PAGI guy and lasts for approximately one second. Both the dynamite and the colored block it came into contact with disappear when the explosion appears, and remain gone after the explosion dissipates.

PAGI guy starts with no knowledge of how to interact with any of these items. He does, however, come equipped with semantically empty symbols for each of the 8 item types ($\mathbf{O_{starting}}$), so that when an object comes into his visual field, he can recognize the type of the object (the type, however, is a semantically empty label, so that if the types of steak and poison were reversed it would make no practical difference to PAGI guy).

5.1.0.1 From Sensor Data to Objects

On the AI-side (using the terminology introduced in Chapter 4), the AI controller is written in python. The simulation starts with a task consisting of the visual elements previously described. PAGI guy then issues a command to retrieve information from his peripheral sensors, which are visual sensors spaced roughly 200 pixels apart (see Figure 5.3). Each visual sensor v returns an array of data describing extremely low-level data: the color data of the pixels at v, a vector describing visual features of the object at v, etc. For this demonstration, the only information used from the visual sensors is a string naming the object which the visual sensor

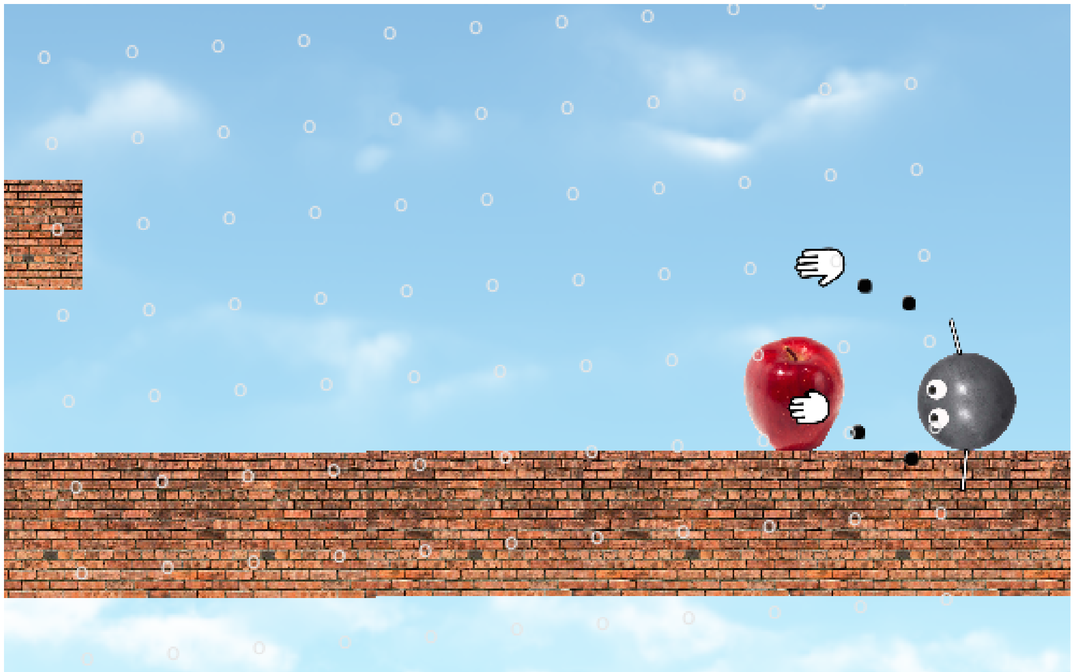

Figure 5.3: PAGI guy with his peripheral vision sensors marked as white 'o's

sees. Again, this string is semantically empty to PAGI guy, and is only meaningful for convenience to the programmer.

For each item type, there is a predicate which can be used to create an observable in predicate-object form. For example, $isApple(o)$ is the ideal output of an observation-generating process which is perceiving an apple. Imagine the situation pictured in Figure 5.3, where the apple object is visible to two or more peripheral vision sensors, which we will call p_1 and p_2. PAGI World provides information that these two peripheral vision sensors have both recognized the symbol 'isApple', and so PAGI guy is given the predicate-sensor pairs $isApple(p_1)$ and $isApple(p_2)$. But under what criteria can PAGI guy determine that these two sensors are referring to the same object? And complicating things further, what if PAGI guy's tactile sensors returns a predicate-sensor pair $isTouchingObject(h_1)$ (where h_1 is one of the tactile sensors on his hands)? What criteria should PAGI guy use to determine whether the object triggering his tactile sensor is the same as the object triggering his peripheral vision sensors?

Questions such as those are central to an $\mathcal{AC}$ research program which makes use of PAGI World. To begin to answer them, I will describe a simple algorithm $\mathscr{A}$ which takes predicate-sensor pairs and outputs predicate-object pairs. $\mathscr{A}$ starts by taking all provided predicate-sensor pairs $P(s)$ and localizing the sensors. Essentially, **localizing** a sensor is assigning a physical location to that sensor such that all predicate-sensor pairs under consideration have a roughly uniform coordinate system. This is akin to a child feeling a sensation on his hand and either looking at the location of the sensation to determine what caused it, or relying on his proprioceptive senses to estimate where he believes his hand, and the sensation, are located. For ease of implementation, I will use the peripheral sensors as a rough coordinate system. If a predicate-sensor pair $P(s)$ is triggered with the tactile sensor s, then $P(s)$ is replaced with $P(s')$, where s' is the peripheral vision sensor which is closest to the location that PAGI guy calculates the event causing $P(s)$ took place. The choice of peripheral vision sensors as a rough coordinate system also have the added benefit that pairs of the form $P(v)$ where v is a peripheral vision sensor do not require any additional conversion.

After the localization step, the algorithm $\mathscr{A}$ employs two rules to combine pairs. Let us assume we are given two localized predicate-sensor pairs $P_1(s_1)$ and $P_2(s_2)$. Combination rule 1 says that if P_1 and P_2 are the same, and s_1 and s_2 are either the same or adjacent, then we combine the two predicate-sensor pairs to produce a predicate-object pair $P_1([s_1, s_2])$. Note that an object here is a combination of vision sensors, and combination rule 1 implicitly assumes that if the exact same visual description holds between two sufficiently close visual sensors, then they are to be considered as the same object.

Combination rule 2 says that if s_1 and s_2 are either the same or adjacent, but P_1 and P_2 are different, then an identity link (see Chapter 3) is created between s_1 and s_2. Combination rules 1 and 2 also generalize to predicate-object pairs as follows: Given two objects $[p_1, ..., p_n]$ and $[q_1, ..., q_m]$, the two objects are considered the same if $p_i = q_j$ for some i, j, and the two objects are considered adjacent if p_i is adjacent to q_j for some i, j.

This algorithm is not perfect, of course. There are many problems with it that are discussed in Section 5.2. But even an algorithm this simple produces results

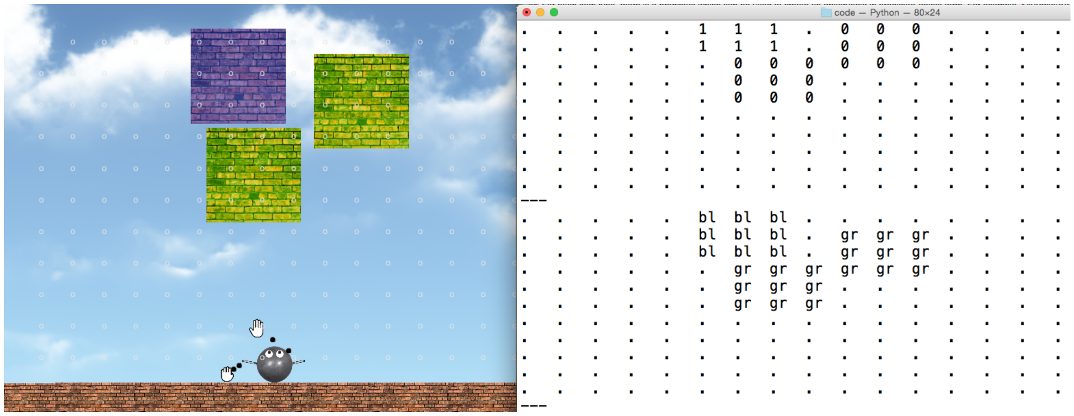

Figure 5.4: PAGI guy observing a configuration of colored wall pieces (left). On the right two grids are displayed. On the first (upper right), each peripheral vision sensor that is detecting an object assigns an index number to each unique object. He labels the two green wall pieces with the same number, indicating that he thinks they are the same object (because his peripheral sensors are spaced too far apart for him to know any better; his detailed vision sensors would correct this misperception). On the second grid (bottom right), each sensor shows the first two letters of the predicate symbol of the observable identified.

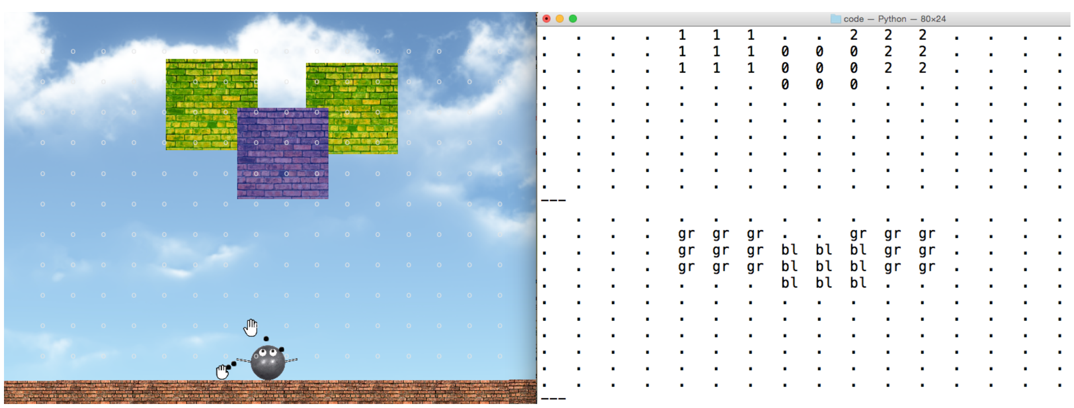

Figure 5.5: PAGI guy observing a configuration of colored wall pieces (left). Note that unlike Figure 5.4, this time the two green wall blocks are spaced far enough that PAGI guy's peripheral vision sensors encode them as two separate objects.

which can be used for real-time interaction with the environment. Take for example, Figures 5.4 and 5.5 which show configurations of objects in PAGI guy's visual field alongside the representations of those objects constructed in real-time (in practice, it took about 0.5 seconds to update this information).

5.2 Discussion

As a demonstration of a theory which takes actions to be central in cognitive constructions, one omission is glaringly obvious: the constructions produced by this demonstration are not at all rooted in action! The significance of this demonstration can only be seen when it is considered as a piece of the larger project of $\mathcal{AC}$. Although action schemas are not made use of here, the ability to turn sensor data into predicate-object representation makes the action schema creation procedure I described in the opening to this chapter much easier. With predicate-object representations, relations can be created using a procedure similar to that described by the DORA model (Doumas et al., 2008), or by an $\mathcal{AC}$-based model that would treat relations as stabilized action instances applied to objects in predicate-object form.

Algorithm $\mathscr{A}$ also needs more work. Much of what makes the algorithm work so well is two simplifying assumptions: First, each visual sensor only produces a single predicate-sensor pair; Second, the perceptual processes which produce predicate-sensor pairs are not self-contradictory, or non-probabilistic (e.g. what happens when we need to consider a predicate-sensor pair which has a confidence of 50%?). More work is necessary to figure out how to make $\mathscr{A}$ more robust to be able to handle weakening of these two assumptions.

5.3 Conclusion and Future Work

PAGI World has already been used in minor demonstrations of cognitive phenomena (Marton, Licato, & Bringsjord, 2015; Atkin, Licato, & Bringsjord, 2015). I have also been introducing PAGI World as a tool for teaching both high- and low-level AI programming techniques, in a class offered during the spring 2015 semester at Rensselaer Polytechnic Institute, called Engineering Human-Level Artificial Cognitive Systems. PAGI World has proven itself a valuable tool both for educational and research goals. The short demonstration presented in this chapter is designed to help convince the reader that PAGI World provides a rich environment with virtually limitless possibilities.

5.3.1 Next Steps for this Demonstration

Building on this chapter's demonstration, I will seek to produce an expanded version of $\mathscr{A}$ which takes into account more robust sensor data, and tries to match its outputs to human behavioral data. This would position PAGI World as a platform upon which much work in cognitive modeling can be built—the representations in predicate-object form can, with some adaptation, be used as input into LISA, DORA, and CLARION, and conceivably other models as well. Successful completion of this work would be followed with a more complete realization of the ideas presented in the first two chapters—namely, the generation of structure rooted in action as required by NSES, and the emergence of reasoning through the analogical-structure-mediated abstractions: *réfléchissement* and *réflexion* (empirical abstraction most closely describes what was modeled in the present chapter's demonstration).

The process of going from purely sensor-level to predicate-object representations is currently nowhere near as sophisticated as contemporary work in computer vision. Next steps, then, would involve using PAGI World and tasks similar to those in this chapter and see how well they perform. A natural direction seems to be the incorporation of the JIM.* models of visual object recognition (Hummel & Biederman, 1992), and to give PAGI guy the ability to recognize 2D-viewpoint-invariant features of objects.

5.3.2 The Future of PAGI World

The release of PAGI World is accompanied by a call to all AGI and human-level-AI researchers to finally examine the strengths and limits of their preferred approaches. PAGI World allows for researchers to very easily create tasks and microworlds in a 2D world with realistic physics, with no knowledge in how to program. PAGI World can interact with AI agents that are written in virtually any programming language, and the simulation can be run on any major operating system. PAGI World has very carefully been designed to have an extremely low technical barrier, so that many researchers can find common ground upon which to compare their different approaches.

The future of PAGI World is bright. Already there are several AI systems in progress solving PAGI World tasks (Marton et al., 2015; Atkin et al., 2015), and as development continues we hope to greatly increase the number of tasks which are available and the sophistication of the agents which solve those tasks. The library of future tasks, we hope, will diversify and reflect the broad spectrum of tasks which require human-like intelligence.

One interesting and possibly fertile source of PAGI World tasks is the area of morality. Figure 4.3 depicts an example task in which two food objects—an apple and a piece of bacon—are falling down a series of ramps where they will eventually fall off the screen and become unreachable, unless the agent chooses exactly one of them (he will not be able to get both in time). Although not many would consider the choice between apples and bacon to be a moral decision, it is easy to see how such a scenario can be adapted to capture miniature moral dilemmas. For example, if the simulation begins with the agent having knowledge that an apple will save the life of person **A**, while the bacon will save person **B** but leave **A** to die, suddenly Figure 4.3 becomes a moral decision which the agent must make in real time. Examples like these illustrate the wide variety of tasks and demonstrations that can be created with PAGI World.

5.3.3 Toward a Research Program in $\mathcal{AC}$

In the long term, I want to pursue a research program in $\mathcal{AC}$. What would an $\mathcal{AC}$ research program look like?

I submit that a research program in $\mathcal{AC}$ is first and foremost *cross-disciplinary*. The research described in this dissertation alone span computer science, AI, cognitive science, and developmental psychology. But I haven't even mentioned yet the implications for education. Presumably a theory of how cognitive structures develop from actions implies that there exists some flexibility in which cognitive structures we try to develop in children. And if such a flexibility exists, then it is in the interest of professional educators to tailor their lessons towards the detection and development of the structures which are most beneficial to the children; this is the goal of approaches such as Papert's "Constructionism" in education (Harel & Papert, 1991).

I have also already applied PAGI World to education at the college level. In the spring semester of 2015 I taught a course at RPI, along with Selmer Bringsjord, which used PAGI World as a platform for a homework assignment and a final project. Students learned how to program and evaluate AI algorithms within the context of PAGI World, and watched their ideas be tested on tasks in real-time. One of the simplest assignments, involving the agent navigating a world containing apples (which were to be seeked out) and poison vials (which were to be avoided), produced a variety of strategies from the students that I could not have anticipated.

A primary measure of success will be whether $\mathcal{AC}$-generated representations can allow for analogical reasoning at the human level. In particular, structural representations should have the flexibility of re-representation that human representations do. Recall that the inability for current models of analogy to successfully replicate such representational flexibility was a motivator for this dissertation's explorations into the nature of structure (Section 1.3). A more robust analogical matcher may allow a connection to the work in Analogico-Deductive Reasoning (ADR). It would be fascinating and also a potentially fruitful area of research to see if an agent like PAGI guy could autonomously develop the ability to perform analogico-deductive inferences in real-world scenarios.

Finally, NSES and its elaboration is an area that will require a lot more work. A research program in $\mathcal{AC}$ would have to explain how all of the things that we believe to have semantic meaning—sounds, events, names, and so on—can ultimately be traced to action-rooted constructions. Armchair theorizing would not be enough. In the spirit of Piaget, NSES needs to be tested empirically: psychological studies on children and adults need to be carried out; neurological data about the parts of the brain associated with action-centered versus non-action-centered knowledge should be contrasted with the predictions of NSES; and the consequences of NSES need to be fleshed out and compared to those of similar models and theories.

Ultimately, the creation of structured representations, in accordance with the views of Analogical Constructivism, may one day achieve the ultimate goal of $\mathcal{AC}$, alluded to in this dissertation's title: the emergence of reasoning through analogy and action schemas.

REFERENCES

Adams, R. P., Wallach, H. M., & Ghahramani, Z. (2010). Learning the Structure of Deep Sparse Graphical Models. In N. Lawrence & M. Reid (Eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS-10).* Sardinia, Italy: JMLR MLOSS.

Aksoy, E. E., Tamosiunaite, M., Vuga, R., Ude, R., Geib, C., Steedman, M., et al. (2013). Structural Bootstrapping at the Sensorimotor Level for the Fast Acquisition of Action Knowledge for Cognitive Robots. In *Proceedings from the Third IEEE International Conference on Development and Learning and on Epigenetic Robotics.* Osaka, Japan: IEEE.

Anderson, J. R. (1976). *Language, Memory and Thought.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Arnon, I., Cottril, J., Dubinsky, E., Oktaç, A., Fuentes, S. R., Trigueros, M., et al. (2013). *APOS Theory : A Framework for Research and Curriculum Development in Mathematics Education.* New York City, NY: Springer.

Asada, M., Hosoda, K., Kuniyoshi, Y., Ishiguro, H., Inui, T., Yoshikawa, Y., et al. (2009). Cognitive Developmental Robotics: A Survey. *IEEE Transactions on Autonomous Mental Development*, *1*(1), 12-34.

Atkin, K., Licato, J., & Bringsjord, S. (2015). Modeling Interoperability Between a Reflex and Reasoning System in a Physical Simulation Environment. In *Proceedings of the 2015 Spring Simulation Multi-Conference (SPRINGSIM-2015.* Alexandria, VA: ACM Press.

Azevedo, R. F., Reis, C. E., & Cornacchione Jr., E. B. (2013, April). The HRI (Human-Robot Interaction) and Human Resource Development (HRD) Encounter. *International Journal of Education and Research*, *1*(4), 1-10.

Barklund, J., Constantini, S., & Lanzarone, G. A. (2000). Reflection Principles in Computational Logic. *Journal of Logic and Computation*, *10*(6), 743-786.

Bartha, P. F. (2010). *By Parallel Reasoning: The Construction and Evaluation of Analogical Arguments.* Oxford, United Kingdom: Oxford University Press.

Bengio, Y., Lamblin, P., Popovici, P., & Larochelle, H. (2007). Greedy Layer-Wise Training of Deep Networks. *Advances in Neural Information Processing Systems (NIPS)*, *19*, 1-8.

Beth, E. W., & Piaget, J. (1966). *Mathematical Epistemology and Psychology.* Amsterdam, Netherlands: D. Reidel Publishing Company / Gordon and Breach Science Publishers, Inc.

Bickhard, M. H. (2008, July). The Interactivist Model. *Synthese*, *166*(3), 547–591.

Brazier, F. M., & Treur, J. (1999). Managing Conflicts in Reflective Agents. In H. J. Müller & R. Dieng (Eds.), *Computational Conflicts: Conflict Modeling for Distributed Intelligent Systems* (p. 63-81). New York City, NY: Springer.

Bridewell, W., & Bello, P. (2015). Incremental Object Perception in an Attention-Driven Cognitive Architecture. In *Proceedings of the Annual Conference of the Cognitive Science Society (CogSci 2015).* Pasadena, CA: Cognitive Science Society.

Bringsjord, S. (2008a). Declarative/Logic-Based Cognitive Modeling. In R. Sun (Ed.), *The Cambridge Handbook of Computational Psychology* (pp. 127–169). New York City, NY: Cambridge Univ Press.

Bringsjord, S. (2008b). The Logicist Manifesto: At Long Last Let Logic-Based AI Become a Field Unto Itself. *Journal of Applied Logic*, *6*(2), 502-525.

Bringsjord, S. (2011). Psychometric Artificial Intelligence. *Journal of Experimental and Theoretical Artificial Intelligence*, *23*(3), 271-277.

Bringsjord, S. (2015). The Symbol Grounding Problem ... Remains Unsolved. *Journal of Experimental and Theoretical AI*, *27*(1), 1-15.

Bringsjord, S., Clark, M., & Taylor, J. (2014). Sophisticated Knowledge Representation and Reasoning Requires Philosophy. In R. Hagengruber & U. Riss (Eds.), *Philosophy, Computing, and Information Science* (p. 99-120). London, United Kingdom: Pickering & Chatto Publishers.

Bringsjord, S., & Licato, J. (2012). Psychometric Artificial General Intelligence: The Piaget-MacGyver Room. In P. Wang & B. Goertzel (Eds.), *Theoretical Foundations of Artificial General Intelligence* (p. 25-48). Amsterdam, Netherlands: Atlantis Press.

Bringsjord, S., & Licato, J. (2015). By Disanalogy, Cyberwarfare is Utterly New. *Philosophy and Technology*, *1*, 1-12.

Bringsjord, S., & Schimanski, B. (2003). What is Artificial Intelligence? Psychometric AI as an Answer. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)* (p. 887-893). San Francisco, CA: Morgan Kaufmann Publishers, Inc.

Bruce, D. (2010). *Cognitive Architecture and Testbed for a Developmental AI System.* Unpublished master's thesis, University of Aberdeen, Aberdeen, United Kingdom.

Campbell, R. L. (2009). Constructive Processes: Abstraction, Generalization, Dialectics. In U. Müller, L. Smith, & J. Carpendale (Eds.), *The Cambridge Companion to Piaget* (p. 150-170). Cambridge, United Kingdom: Cambridge University Press.

Case, R. (1992). Neo-Piagetian Theories of Child Development. In R. J. Sternberg & C. A. Berg (Eds.), *Intellectual Development* (pp. 161–196). New York City, NY: Cambridge Univ Press.

Chalmers, D., French, R. M., & Hofstadter, D. R. (1992). High-level Perception, Representation, and Analogy: A Critique of Artificial Intelligence Methodology. *Journal of Experimental and Theoretical Artificial Intelligence*, *4*(3), 185-211.

Chaput, H. H., Kuipers, B., & Miikkulainen, R. (2003). Constructivist Learning: A Neural Implementation of the Schema Mechanism. In *Proceedings of the Workshop on Self Organizing Maps (WSOM03).* Kitakyushu, Japan: FCFM.

Chemero, A. (2003). An Outline of a Theory of Affordances. *Ecological Psychology*, *15*(2), 181-195.

Chomsky, N. (1965). *Aspects of the Theory of Syntax.* Boston, MA: The MIT Press.

Cohen, P. R., Chang, Y.-H., & Morrison, C. T. (2007). Learning and Transferring Action Schemas. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (p. 720-725). San Francisco, CA: Morgan Kaufmann Publishers, Inc.

Dorigo, M., & Blum, C. (2005). Ant Colony Optimization Theory: A Survey.

Theoretical Computer Science, *344*(2-3), 243-278.

Doumas, L. A., & Hummel, J. E. (2013). Comparison and Mapping Facilitate Relation Discovery and Predication. *PLoS ONE*, *8*(6), 1-8.

Doumas, L. A., Hummel, J. E., & Sandhofer, C. (2008). A Theory of the Discovery and Predication of Relational Concepts. *Psychological Review*, *115*(1), 1-43.

Drescher, G. L. (1991). *Made-Up Minds: A Constructivist Approach to Artificial Intelligence*. Boston, MA: The MIT Press.

Drumwright, E., Hsu, J., Koening, N., & Shell, D. (2010). Extending Open Dynamics Engine for Robotics Simulation. *Lecture Notes in Computer Science*, *6472*, 38-50.

Dubinsky, E. (1991). Reflective Abstraction in Advanced Mathematical Thinking. In D. Tall (Ed.), *Advanced Mathematical Thinking* (p. 95-123). Norwell, MA: Kluwer Academic Publishers.

Dubinsky, E., & Mcdonald, M. A. (2002). APOS: A Constructivist Theory of Learning in Undergraduate Mathematics Education Research. In M. Artigue, U. Kirchgräber, J. Hillel, M. Niss, & A. Schoenfeld (Eds.), *The Teaching and Learning of Mathematics at University Level* (Vol. 7, p. 275-282). Norwell, MA: Kluwer Academic Publishers.

Falkenhainer, B., Forbus, K. D., & Gentner, D. (1989). The Structure-Mapping Engine: Algorithm and Examples. *Artificial Intelligence*, *41*(1), 1-63.

Feferman, S. (1962). Transfinite Recursive Progressions of Axiomatic Theories. *Journal of Symbolic Logic*, *27*(3), 259-316.

Floridi, L. (2011a). A Defence of Constructionism: Philosophy as Conceptual Engineering. *Metaphilosophy*, *42*(3), 282-304.

Floridi, L. (2011b). *The Philosophy of Information*. Oxford, United Kingdom: Oxford University Press.

Fodor, J. A. (1980). *The Language of Thought* (2nd ed.). Cambridge, MA: Harvard University Press.

Fodor, J. A., & Pylyshyn, Z. (1988). Connectionism and Cogntive Architecture: A Critical Analysis. *Cognition*, *28*(1-2), 3-71.

Forbus, K. D., Gentner, D., Markman, A., & Ferguson, R. (1998). Analogy Just

Looks Like High Level Perception: Why a Domain-General Approach to Analogical Mapping is Right. *Journal of Experimental and Theoretical Artificial Intelligence (JETI)*, *10*(2), 231-257.

French, R. M., & Hofstadter, D. R. (1991). Tabletop: A Stochastic, Emergent Model of Analogy-Making. In *Proceedings of the 13th Cognitive Science Society Conference (COGSCI).* Hillsdale, NJ: Conference Science Society.

French, S. (2014). *The Structure of the World: Metaphysics and Representation.* Oxford, United Kingdom: Oxford University Press.

Friedenberg, J., & Silverman, G. (2012). *Cognitive Science: An Introduction to the Study of Mind* (2nd ed.). Thousand Oaks, CA: Sage Publications, Inc.

Gallagher, J. M., & Reid, D. K. (2002). *The Learning Theory of Piaget and Inhelder.* Lincoln, NE: Authors Choice Press.

Gentner, D. (1983). Structure-Mapping: A Theoretical Framework for Analogy. *Cognitive Science*, *7*(2), 155-170.

Gentner, D., & Forbus, K. (2011). Computational Models of Analogy. *Wiley Interdisciplinary Reviews: Cognitive Science*, *2*(3), 266-276.

Gentner, D., & Toupin, C. (1986). Systematicity and Surface Similarity in the Development of Analogy. *Cognitive Science*, *10*(3), 277-300.

Gibson, J. (1977). The Theory of Affordances. In R. Shaw & J. Bransford (Eds.), *Perceiving, Acting, and Knowing: Toward an Ecological Psychology.* Hillsdale, NJ: Erlbaum.

Glasersfeld, E. von. (1991). Abstraction, Re-Presentation, and Reflection: An Interpretation of Experience and of Piaget's Approach. In L. P. Steffe (Ed.), *Epistemological Foundations of Mathematical Experience* (p. 45-67). New York, New York, USA: Springer.

Govindarajulu, N. S., Licato, J., & Bringsjord, S. (2013). Small Steps Toward Hypercomputation via Infinitary Machine Proof Verification and Proof Generation. In G. Mauri, A. Dennunzio, L. Manzoni, & A. E. Porreca (Eds.), *Proceedings of Unconventional Computation and Natural Computation (UCNC 2013)* (p. 102-112). Milan, Italy: Springer Internaltional Publishing.

Guerin, F. (2011). Learning Like a Baby: A Survey of Artificial Intelligence Approaches. *The Knowledge Engineering Review*, *26*(2), 209-236.

Harel, I., & Papert, S. (1991). *Constructionism*. New York City, NY: Ablex Publishing.

Harnad, S. (1990). The Symbol Grounding Problem. *Physica D*, *42*, 335-346.

Harrison, J. (1995). *Metatheory and Reflection in Theorem Proving: A Survey and Critique* (Tech. Rep. No. CRC-053). University of Cambridge, Cambridge, United Kingdom: SRI Cambridge.

Hernandez-Belmonte, U. H., Ayala-Ramirez, V., & Sanchez-Yanez, R. E. (2011). A Mobile Robot Simulator Using a Game Development Engine. In *Proceedings of the 2011 Robotics Summer Meeting (ROSSUM).* Veracruz, Mexico: ROSSUM Committee.

Hinton, G. E., Osindero, S., & Teh, Y. (2006). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, *18*(7), 1527-1554.

Hofstadter, D. R. (1984). *The Copycat Project: An Experiment in Nondeterminism and Creative Analogies* (Tech. Rep. No. AI Memo 755). Boston, MA: Massachusetts Institute of Technology.

Hofstadter, D. R. (2001). Epilogue: Analogy as the Core of Cognition. In D. Gentner, K. J. Holyoak, & B. N. Kokinov (Eds.), *The Analogical Mind: Perspectives from Cognitive Science* (p. 499-538). Boston, MA: The MIT Press.

Hofstadter, D. R., & Mitchell, M. (1995). The Copycat Project: A Model of Mental Fluidity and Analogy-making. In D. R. Hofstadter & the Fluid Analogies Research Group (Eds.), *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought* (p. 205-268). New York City, NY: Basic Books, Inc.

Hofstadter, D. R., & Sander, E. (2013). *Surfaces and Essences: Analogy as the Fuel and Fire of Thinking*. New York City, NY: Basic Books, Inc.

Holyoak, K. J., & Hummel, J. E. (2000). The Proper Treatment of Symbols in a Connectionist Architecture. In E. Deitrich & A. Markman (Eds.), *Cognitive Dynamics: Conceptual Change in Humans and Machines* (p. 229-264). Cambridge, MA: MIT Press.

Holyoak, K. J., & Koh, K. (1987). Surface and Structural Similarity in Analogical

Transfer. *Memory and Cognition*, *15*(4), 332-340.

Holyoak, K. J., & Thagard, P. (1989). Analogical Mapping by Constraint Satisfaction. *Cognitive Science*, *13*(3), 295-355.

Hummel, J. E. (2001). Complementary Solutions to the Binding Problem in Vision: Implications for Shape Perception and Object Recognition. *Visual Cognition*, *8*(3), 489-517.

Hummel, J. E., & Biederman, I. (1992). Dynamic Binding in a Neural Network for Shape Recognition. *Psychological Review*, *99*(3), 480-517.

Hummel, J. E., & Holyoak, K. J. (1997). Distributed Representations of Structure: A Theory of Analogical Access and Mapping. *Psychological Review*, *104*(3), 427-466.

Hummel, J. E., & Holyoak, K. J. (2003a). Relational Reasoning in a Neurally-plausible Cognitive Architecture: An Overview of the LISA Project. *Cognitive Studies: Bulletin of the Japanese Cognitive Science Society*, *10*, 58-75.

Hummel, J. E., & Holyoak, K. J. (2003b). A Symbolic-Connectionist Theory of Relational Inference and Generalization. *Psychological Review*, *110*(2), 220-264.

Hummel, J. E., & Landy, D. H. (2009). From Analogy to Explanation: Relaxing the 1:1 Mapping Constraint...Very Carefully. In B. Kokinov, K. J. Holyoak, & D. Gentner (Eds.), *New Frontiers in Analogy Research: Proceedings of the Second International Conference on Analogy* (p. 1-9). Sofia, Bulgaria: NBU Press.

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

Ivaldi, S., Padois, V., & Nori, F. (2014). *Tools for Dynamics Simulation of Robots: A Survey Based on User Feedback.* Retrieved 3/1/2015, from http://arxiv.org/abs/1402.7050

Jackson, F. (1977). *Perception: A Representative Theory.* Cambridge, United Kingdom: Cambridge University Press.

Kahneman, D. (2011). *Thinking, Fast and Slow.* New York City, NY: Farrar, Straus and Girous.

Kosgeroglu, N., Acat, M. B., Ayranci, U., Ozbaci, N., & Erkal, S. (2009). An Investigation on Nursing, Midwifery and Health Care Students' Learning Motivation in Turkey. *Nurse Education in Practice*, *9*(5), 331-339.

Kowalski, R., & Sergot, M. (1986). A Logic-based Calculus of Events. *New Generation Computing*, *4*(1), 67-94.

Kramer, J., & Scheutz, M. (2007). Development Environments for Autonomous Mobile Robots: A Survey. *Autonomous Robots*, *22*(2), 101-132.

Laird, J., & Lent, M. van. (2001). Human-Level AI's Killer Application: Interactive Computer Games. *AI Magazine*, *22*(2), 15-25.

Licato, J., Bringsjord, S., & Govindarajulu, N. S. (2013). How Models of Creativity and Analogy Need to Answer the Tailorability Concern. In T. R. Besold, K.-u. Kühnberger, M. Schorlemmer, & A. Smaill (Eds.), *Proceedings of the IJCAI 2013 Workshop on Computational Creativity, Concept Invention, and General Intelligence.* Beijing, China: Morgan Kaufmann Publishers, Inc.

Licato, J., Bringsjord, S., & Hummel, J. E. (2012). Exploring the Role of Analogico-Deductive Reasoning in the Balance-Beam Task. In *Rethinking Cognitive Development: Proceedings of the 42nd Annual Meeting of the Jean Piaget Society.* Toronto, Canada: Jean Piaget Society.

Licato, J., Govindarajulu, N. S., Bringsjord, S., Pomeranz, M., & Gittelson, L. (2013). Analogico-Deductive Generation of Gödel's First Incompleteness Theorem from the Liar Paradox. In *Proceedings of the 23rd Annual International Joint Conference on Artificial Intelligence (IJCAI-13)* (p. 1004-1009). San Francisco, CA: Morgan Kaufmann Publishers, Inc.

Licato, J., Sun, R., & Bringsjord, S. (2014). Structural Representation and Reasoning in a Hybrid Cognitive Architecture. In *Proceedings of the 2014 International Joint Conference on Neural Networks (IJCNN).* Beijing, China: IEEE.

Lourenço, O., & Machado, A. (1996). In Defense of Piaget's Theory: A Reply to 10 Common Criticisms. *Psychological Review*, *103*(1), 143-164.

Lungarella, M., Metta, G., Pfeifer, R., & Sandini, G. (2003). Developmental Robotics: A Survey. *Connection Science*, *15*(4), 151-190.

Marshall, J. B. (2006). A Self-Watching Model of Analogy-Making and Perception. *Journal of Experimental and Theoretical Artificial Intelligence*, *18*(3), 267-307.

Marton, N., Licato, J., & Bringsjord, S. (2015). Creating and Reasoning Over Scene Descriptions in a Physically Realistic Simulation. In *Proceedings of the 2015 Spring Simulation Multi-Conference.* Alexandria, VA: ACM Press.

Mattingly, W. A., Chang, D.-J., Paris, R., Smith, N., Blevins, J., & Ouyang, M. (2012). Robot Design Using Unity for Computer Games and Robotic Simulations. In *Proceedings of the 17th International Conference on Computer Games (CGAMES 2012).* Louisville, KY: IEEE.

Mitchell, M. (1993). *Analogy-Making as Perception : A Computer Model.* Cambridge, MA: The MIT Press.

Montangero, J., & Maurice-Naville, D. (1997). *Piaget or the Advance of Knowledge: An Overview and Glossary.* Mahwah, NJ: Lawrence Elrbaum Associates, Inc.

Morra, S., Gobbo, C., Marini, Z., & Sheese, R. (2008). *Cognitive Development : Neo-Piagetian Perspectives.* New York City, NY: Lawrence Erlbaum Associates.

Nichols, E. P. (2012). *Musicat: A Computer Model of Musical Listening and Analogy-Making.* Unpublished doctoral dissertation, Indiana University, Bloomington, IN.

Pascual-Leone, J. (1970). A mathematical model for the transition rule in Piaget's developmental stages. *Acta Psychologica*, *32*(1), 301-345.

Piaget, J. (1970). *Structuralism.* New York City, NY: Basic Books, Inc.

Piaget, J., & Garcia, R. (1974). *Understanding Causality* (D. Miles & M. Miles, Eds.). New York City, NY: W.W. Norton and Company, Inc.

Piaget, J., Garcia, R., & Davidson, P. (1991). *Toward a Logic of Meanings.* Mahwah, NJ: Lawrence Elrbaum Associates, Inc.

Piaget, J., Montangero, J., & Billeter, J. (2001). *Studies in Reflecting Abstraction.* New York City, NY: Psychology Press.

Piattelli-Palmarini, M. (Ed.). (1980). *Language and Learning : The Debate between Jean Piaget and Noam Chomsky.* Cambridge, MA: Harvard University Press.

Platt, R., Grupen, R. A., & Fagg, A. H. (2006). Improving Grasp Skills Using

Schema Structured Learning. In *Proceedings of the Fifth International Conference on Development and Learning.* Bloomington, IN: IEEE.

Quartz, S. R., & Sejnowski, T. J. (1997). The Neural Basis of Cognitive Development: A Constructivist Manifesto. *Behavioral and Brain Sciences*, *20*(4), 537-596.

Sammoud, O., Solnon, C., & Ghédira, K. (2005). An Ant Algorithm for the Graph Matching Problem. In *5th European Conference on Evolutionary Computation in Combinatorial Optimization (EvoCOP 2005).* New York City, NY: Springer.

Saxe, A., McClelland, J. L., & Ganguli, S. (2013). Learning Hierarchical Category Structure in Deep Neural Networks. In M. Knauff, M. Paulen, N. Sebanz, & I. Wachsmuth (Eds.), *Proceedings of the 35th Annual Meeting of the Cognitive Science Society (COGSCI 2013)* (p. 1271-1276). Austin, TX: Cognitive Science Society.

Socher, R., Lin, C. C.-Y., Ng, A. Y., & Manning, C. D. (2011). Parsing Natural Scenes and Natural Language with Recursive Neural Networks. In *Proceedings of the 28th International Conference on Machine Learning.* Bellevue, WA: ICML Society.

Solo, R. (1975). What is Structuralism? Piaget's Genetic Epistemology and the Varieties of Structuralist Thought. *Journal of Economic Issues*, *9*(4), 605-625.

Sun, R. (1995). Robust Reasoning: Integrating Rule-Based and Similarity-Based Reasoning. *Artificial Intelligence*, *75*(2), 241-295.

Sun, R. (2001). From Implicit Skills to Explicit Knowledge: A Bottom-Up Model of Skill Learning. *Cognitive Science*, *25*(2), 203-244.

Sun, R. (2002). *Duality of the Mind: A Bottom Up Approach Toward Cognition.* Mahwah, NJ: Lawrence Erlbaum Associates.

Sun, R. (2004). Desiderata for Cognitive Architectures. *Philosophical Psychology*, *17*(3), 341-373.

Sun, R. (2012a). Autonomous Generation of Symbolic Representations Through Subsymbolic Activities. *Philosophical Psychology*, *26*(6), 888-912.

Sun, R. (2012b). Memory Systems within a Cognitive Architecture. *New Ideas in*

Psychology, *30*(2), 227-240.

Sun, R. (2014). Interpreting Psychological Notions: A Dual-Process Computational Theory. *Journal of Applied Research in Memory and Cognition*, *In Press*.

Sun, R., & Zhang, X. (2004). Accounting for Similarity-Based Reasoning within a Cognitive Architecture. In S. Bringsjord & Y. Yang (Eds.), *Proceedings of the 26th Annual Conference of the Cognitive Science Society* (p. 169-191). Mahwah, NJ: Lawrence Erlbaum Associates.

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), 169-191.

Sutton, R. S. (2006). *Verification, the Key to AI.* Retrieved 3/1/2015, from http://www.cs.ualberta.ca/ sutton/IncIdeas/KeytoAI.html

Taddeo, M., & Floridi, L. (2005). Solving the Symbol Grounding Problem: a Critical Review of Fifteen Years of Research. *Journal of Experimental and Theoretical Artificial Intelligence*, *17*(4), 419-445.

Taddeo, M., & Floridi, L. (2007). A Praxical Solution of the Symbol Grounding Problem. *Minds and Machines*, *17*(4), 369-389.

Thagard, P., Holyoak, K. J., Nelson, G., & Gochfeld, D. (1993). Analog Retrieval by Constraint Satisfaction. *Artificial Intelligence*, *46*(3), 259-310.

Turner, T. (1973). Piaget's Structuralism. *American Anthropologist*, *75*(2), 351-373.

Tversky, A. (1977). Features of Similarity. *Psychological Review*, *84*(4), 327-352.

Whitehead, A. N. (1919). *An Enquiry Concerning the Principles of Natural Knowledge*. Cambridge, United Kingdom: Cambridge University Press.

Winograd, T. (1975). Frame Representations and the Declarative-Procedural Controversy. In D. Bobrow & A. Collins (Eds.), *Representation and Understanding: Studies in Cognitive Science* (p. 185-210). New York City, NY: Academic Press.

Witt, J. K. (2011). Action's Effect on Perception. *Current Directions in Psychological Science*, *20*(3), 201-206.

Witt, J. K., & Proffitt, D. R. (2005). See the Ball, Hit the Ball: Apparent Ball Size is Correlated with Batting Average. *Psychological Science*, *16*(2), 937-938.