

Polyscheme:
A Cognitive Architecture for Integrating Multiple
Representation and Inference Schemes

by
Nicholas L. Cassimatis

Bachelor of Science in Mathematics; Massachusetts Institute of Technology, 1994
Master of Art in Psychology; Stanford University, 1996

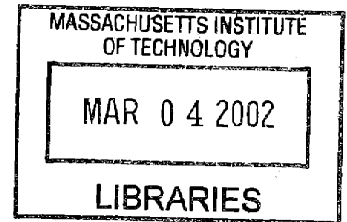
Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

Massachusetts Institute of Technology

February 2002



© Massachusetts Institute of Technology, 2001. All rights reserved.

ARCHIVES

Author:

Program in Media Arts and Sciences
November 9, 2001

Certified by:

Marvin Minsky
Professor of Media Arts and Sciences
Thesis Supervisor

Certified by:

Andrew B. Lippman
Chairman, Department Committee on Graduate Students
Program in Media Arts and Sciences

Polyscheme: A Cognitive Architecture for Integrating Multiple Representation and Inference Schemes

by
Nicholas L. Cassimatis

Submitted to the Program in Media Arts and Sciences on November 9, 2001
in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Abstract

In order to understand and create human-level intelligence I have developed the Polyscheme cognitive architecture to build systems that combine several representation and inference schemes when they think.

Polyscheme is based on three principles. First, different (aspects of) situations that intelligent systems must deal with are best modeled with different schemes for representing knowledge and making inferences. Polyscheme includes several "specialists" such that each models a particular aspect of the world with its own (possibly unique) representation and inference techniques. Second, specialists must communicate with other specialists frequently so that each specialist uses the most complete, accurate and relevant information when it deals with a situation. Specialists in Polyscheme communicate and combine information by simultaneously concentrating on the same focus of attention. Finally, because information about some aspects of a situation is more important than information about others and because the order that specialists focus on those aspects is important, a system of focused specialists must have mechanisms that decide where to focus. Polyscheme's specialists, especially the reflective specialist, guide the focus of attention and thereby implement inference schemes using an "attraction" mechanism to specify their preferred foci.

Polyscheme enables multiple inference techniques to be integrated in dealing with a situation because each inference technique can be implemented with one or more focus schemes. I describe how to implement several important inference techniques (e.g., script matching, backtracking search, reason maintenance, stochastic simulation and counterfactual reasoning) as focus schemes.

I have used Polyscheme to implement the S6 system for common sense physical reasoning. S6 views interactions in a simple physical world through a 2-dimensional projection of that world. S6 keeps track of the identity of objects, infers the character and existence of events it cannot see, predicts the outcome of events, explains events and nonevents and revises its inferences when it receives new information. S6 successfully reasons about many scenarios researchers present to infants and young children in order to study their knowledge of the physical world.

S6 combines specialized representation and inference techniques for identity, time, events, causality, space and paths to successfully deal with a wide range of situations. The knowledge representation schemes S6 uses include scripts, frames, logical propositions, neural networks and constraint graphs. The inference schemes S6 implements include script matching, rule matching, backtracking search, neural network propagation and counterfactual reasoning. I show that these representation and inference schemes form part of a common sense substrate that underlies much of human cognition. The success of S6 therefore demonstrates that Polyscheme is important for understanding and building intelligent systems in any domain of human cognition.

Thesis Advisor: Marvin Minsky

Title: Professor of Media Arts and Sciences

Thesis Committee

Thesis Advisor:

Marvin Minsky
Professor
Media Laboratory, MIT

Thesis Reader:

Hans Moravec
Senior Research Scientist
Robotics Institute, Carnegie Mellon University

Thesis Reader:

Patrick Winston
Professor
Artificial Intelligence Laboratory, MIT

Acknowledgements

There are many people and institutions to acknowledge and I will do so mostly in chronological order.

First, I would like to thank my parents for a long series of computers that began with an IBM PCjr, for having made it possible for me to get a great education and much more.

Raymond Caciale, my French and philosophy teacher at Hubbard High School, introduced me to a world of ideas that has been an important influence on my life and research ever since.

As an undergraduate student at MIT, I worked with several faculty members who each gave me a chance to learn about various branches of artificial intelligence and cognitive psychology. These included Robert Berwick, Susan Carry, Ken Haase, Pattie Maes, Marvin Minsky and Marry Potter. Most of this work occurred within the framework of MIT's important Undergraduate Research Opportunities program which enables hundreds of undergraduates each year work at the frontier of science and technology.

As a graduate student at the Stanford Psychology Department, I learned a great deal about child psychology from Ellen Markman, especially while doing studies in her lab, and from John Flavel. I am grateful to have had the support and freedom to do research at Stanford that did not fit well with what anyone else was doing.

I was also very fortunate to meet Tim Paek at Stanford. Dozens of conversations with Tim impressed upon me the role of common sense social reasoning in conversations. His insights are beginning a revolution in the study of human language and will ultimately transform the interface between humans and computers.

As a graduate student at the MIT Media Lab, I had the unique opportunity to be a student in Marvin Minsky's Society of Mind Group. Without the freedom and support

Marvin has given me for the last five years, the work in this thesis would have taken longer. Ryan Bender, Tim Chklosky, Betty Lou McClanahan and Push Singh made the group an interesting place to work. Betty Lou and Linda Peterson also occasionally helped my progress through the program with their considerable influence.

My thesis committee – Marvin Minsky, Patrick Winston and Hans Moravec – have provided interesting feedback on my research and on this thesis. Patrick Winston urged me to clearly explain and motivate each feature of Polyscheme and S6 as I introduced it and that has made the thesis much more readable.

I have had good roommates throughout graduate school. Although Joost Bonsen, Rehan Khan and Paul Whitmore are all very different people, each provided a stimulating outlook that was a great contrast to modern academic orthodoxy.

Finally, though neither Han Huang nor Patrycja Missiuro had much to do with this thesis, they are each in their own way a force of nature that provided a welcomed background of intensity at various intervals during my time at MIT.

Table of Contents

Chapter 1. Integrated Theory of Intelligence.....	10
1.1 Uniformity and integration as research strategies.....	10
1.2 Common sense systems need multiple inference schemes to use knowledge ...	12
1.3 Inference schemes must be flexible.....	14
1.3.1 Sharing progress and exploiting opportunities in path search and property comparison.....	15
1.3.2 Rules that are sensitive to the situation	18
1.3.3 Flexible case-based reasoning.....	20
1.3.4 Assembling new inference schemes for a problem.....	21
1.3.5 Inference schemes must interrupt each other gracefully.....	21
1.4 Summary.....	22
Chapter 2. Common Sense Physics and the Substrate.....	25
2.1 Reasoning about physical events requires most of artificial intelligence	25
2.1.1 Deduction, falsification, default reasoning and explanation.....	27
2.1.2 Temporal reasoning.....	29
2.1.3 Belief revision and reason-maintenance	31
2.1.4 Searching through problem spaces	31
2.1.5 Probabilistic inference	33
2.1.6 Common sense physics and “high-level” reasoning	34
2.2 Linguistics and Psychology.....	34
2.2.1 Language Semantics.....	34
2.3 The common sense substrate.....	37
Chapter 3. Specialists.....	43
3.1 Why specialists.....	43
3.2 The specialists in S6.....	45
3.2.1 Perception Specialist.....	45
3.2.2 Attribute Specialist.....	45
3.2.3 Physics Specialist.....	46

3.2.4	Causal Specialist	47
3.2.5	Temporal Specialist	47
3.2.6	Identity Hypothesis Specialist	47
3.2.7	Identity Specialist	48
3.2.8	Place Specialist	48
3.2.9	Path Specialist	49
3.3	Summary of the representation and inference schemes S6 uses.....	50
Chapter 4. Focusing Specialists		51
4.1	Specialists must focus on the same object, attribute, relation or event	51
4.2	How specialists focus in S6.....	54
4.2.1	Propositions	54
4.2.2	Focus.....	58
4.2.3	Focus in Polyscheme is more general than visual attention.....	58
4.3	How the specialists relate to the focus.....	59
4.4	Psychological evidence for integrative focus of attention.....	59
4.4.1	Feature integration theory	60
4.4.2	Stroop effect	60
4.4.3	Human and rat navigation.....	60
Chapter 5. Managing Focus		62
5.1	Guiding focus	62
5.2	Polyscheme's mechanisms for guiding focus	63
5.2.1	Attractions.....	64
5.2.2	Reflective specialist	65
5.3	Focus schemes realize inference schemes.....	66
5.3.1	Some inference schemes require several specialists	67
5.3.2	Some inference schemes in S6.....	68
5.4	Polyscheme integrates the important frameworks of artificial intelligence research.....	74
5.5	Appendix. Reasoning about focus	75
5.5.1	Knowing that you do not know	76
5.5.2	Knowing you can.....	76
5.5.3	Dangerous thoughts.....	77

Chapter 6. Polyscheme and Physical Reasoning.....	78
6.1 The program.....	78
6.2 The bridge example.....	78
6.3 More examples	85
6.3.1 Gravity and imagination.....	85
6.3.2 Belief Revision	88
6.3.3 Object permanence and identity.....	88
6.3.4 Identity blocked by a failure to find a path	88
6.3.5 Postponing identity	91
Chapter 7. Conclusions	92
Appendix.....	94
Bibliography	99

Chapter 1

An Integrated Theory of Intelligence

This thesis describes a new framework for understanding and creating human-level intelligence by integrating multiple representation and inference schemes. In this first chapter, I will describe why the problem of combining several schemes is so important and why I have not tried to find one common computational scheme to explain all of intelligence.

1.1 Uniformity and integration as research strategies

So far in the history of artificial intelligence, researchers have discovered several major schemes for representing knowledge and making inferences. These include, among others, logical deduction, heuristic search, rule-based systems, connectionist networks, and Bayesian networks. Each of these frameworks has advanced our understanding of intelligence and our ability to build intelligent machines.

Each framework let researchers build systems that could solve formerly intractable problems. Logical deduction systems discovered and justified knowledge based on easily expressed axioms and potentially long chains of reasoning. Heuristic search systems solved problems by evaluating the impact of particular actions and their interactions. With rule-based systems, knowledge engineers could easily build thousands of pieces of knowledge into systems that often matched or surpassed human experts in many applications. Connectionist networks were able to learn new classes of patterns without any explicit programming. Bayesian networks enabled systems to integrate a wide range of uncertain information and make reasonable inferences¹.

¹ See Minsky's (1992) article on causal diversity.

However, as research within each framework continued, investigators encountered limits. Logical deduction systems could not make inferences that humans made easily. Heuristic search systems found that for any but the most restricted problems, there were too many possible solutions to search through. Rule-based systems were often too brittle to adapt to situations that were only somewhat different from those for which they were programmed. Neural networks could not represent certain classes of patterns that involved "global" constraints between their parts. Bayesian networks were often slow to deal with problems that unfolded over time or whose components were not all known in advance.

One reaction to these limits was to try to extend each framework to work in more situations. This broadened the power and scope of the framework and has deepened our understanding of just how and why it is so powerful. Still, most limits remain. Logical reasoners have not become good at recognizing alphanumeric characters. Bayesian networks still cannot play a good game of chess.

Some proponents of each framework persist and continue to address the limits and occasionally produce valuable insights. Fanaticism, however, is not the only productive research strategy. Another approach is to accept that no particular framework will be enough to build broadly intelligent systems and to try to build systems that integrate components from each framework. When one of these "integrated" systems encounters a new situation, it will have a wide variety of resources to understand the situation and decide on an appropriate action. When one resource (based, say, on neural networks) stops making progress, another resource (based on logic, or rules, etc.) can take over and advance the system's progress. That is, instead of basing a system on just one framework, build systems with several components based on a variety of resources and find a way to deploy the appropriate resources.

In this thesis, I describe an architecture for building intelligent systems based on this "integrative" approach. It is impossible conclusively argue that trying to build intelligent systems on any particular framework will fail. I will merely note that there are thousands and thousands of researchers who are trying to extend their favorite

framework to cover all of intelligence without serious success while very few researchers have tried to build integrated systems.

There is another way to motivate an integrative research strategy. Most research in AI has followed one of two modes: (1) extend or better understand the techniques of a computational framework or (2) find a way to solve a particular problem or build a particular kind of system within a framework. Both approaches, consciously or not, aim to advance a framework. Achieving human-level machine intelligence or understanding the human mind is only an implicit goal based on the tacit premise that our ultimate understanding of intelligence will be based on one particular framework. However, framework-driven research can keep you from appreciating the need to integrate frameworks. Each mode keeps you focused on what your framework can do and does not pressure you to consider cases where a framework fails or wonder how to integrate the framework with others.

In my own research, I chose a third mode. I first decided to understand common sense physical reasoning and only then searched for computational techniques to use. In this mode, I confronted the problems of integration early and often.

I will explain why I chose common sense physics in the next chapter. But first, I will show how you really do need a surprising number of computational techniques and that the problem of integrating them together is surprisingly difficult.

1.2 Common sense systems need multiple inference schemes to use knowledge

An intelligent system must know many things about the world and be able to use that knowledge to make inferences. Methods, routines or procedures for using knowledge, which I will call **inference schemes**, include techniques such as case-based reasoning and search as well as more mundane imperatives such as “simulate the results of an action before executing it.”

The appendix lists some inference schemes. Their breadth and number along with some more detailed examples will illustrate how important inference schemes are to common sense reasoning.

As a first example, consider one of the most basic facts about solid physical objects: they travel along continuous paths in space and time. Objects never move from one place to another without passing through several adjacent, intermediate places. A system that reasons about physical objects must know this in order to have common sense. But that knowledge is useless unless the system knows when to check for continuous paths and how to do so. A system that reasons about physical objects thus needs *path-search* imperatives such as:

- When you see an object that you think is the same as an object you saw in the past, look for a continuous path between the locations of each sighting.
- To look for a continuous path between locations, check first to see if they are adjacent, if not ...

We also know that objects, physical or otherwise, have certain attributes that persist or change slowly over time. Mt. Washington, e.g., will be in New Hampshire during the next decade as well as this one. This leads to the following *persistent-property comparison* imperative:

- When you glimpse an object at one time and then glimpse another object at another time and suspect that the two glimpses may be of the same object, compare the persistent attributes in each glimpse. If the attributes are not the same, then conclude that the two objects are not identical.

Some properties of objects change, but only because of some external event. If the color of a car changes between two sightings of it, it must have been painted, soiled or dramatically rusted. This suggests an *explain change* inference pattern:

- When a property of an object changes, consider actions that could have led to that change.
- If there is more than one action that could have led to the change, assume that one happened, see if you can find evidence for it and make sure it is consistent with everything else you know. If you conclude nothing definite about it, assume one of the other actions happened

These are just a few of the inference schemes that one finds everywhere throughout common sense reasoning. The appendix is only a rough sample of the possibilities.

1.3 Inference schemes must be flexible

Perhaps the most obvious approach to building a system with several inference schemes is to make modules for each inference pattern or collection thereof. The system would have an executive system that decides when to use which module dispatch control of the system's resources to it. I will call this a **modular** system.

This subsection will show that modular systems have serious problems and that you need a more sophisticated architecture to build a system with several representation and inference schemes. To an extent impossible in modular systems, inference schemes must share progress, exploit unforeseen opportunities, interrupt each other without hazard and be responsive to world knowledge.

A few examples will show what this means and why it is so important.

1.3.1 Sharing progress and exploiting opportunities in path search and property comparison

The first example involves the path search and persistent-property comparison inference schemes mentioned in section 1.1. Imagine a river that is several meters wide and just as deep. A bridge spans the river and has a large and deep puddle of water on it. (See Figure 1.) You, the hypothetical common sense system in the figure, are on one side of the river and see a man on the other side of the river. He is lanky, six-foot tall and has red hair. Call him "Red".

For a few minutes you ignore Red and then turn around to see your friend Joe coming from the direction of the bridge. Joe is lanky, six-foot tall and has red hair. You therefore wonder if Joe is the same person as Red.

Now consider a modular system that dispatches between algorithms depending on the situation. After hypothesizing that Joe and Red is the same person, it runs a path search algorithm, which finds a path Red could have taken across the river. Next, the persistent-property comparison algorithm compares the properties it knew of Red (lanky, six-foot tall, red hair) to see if Joe has them also. He does. A modular system would thus permit the identity hypothesis.

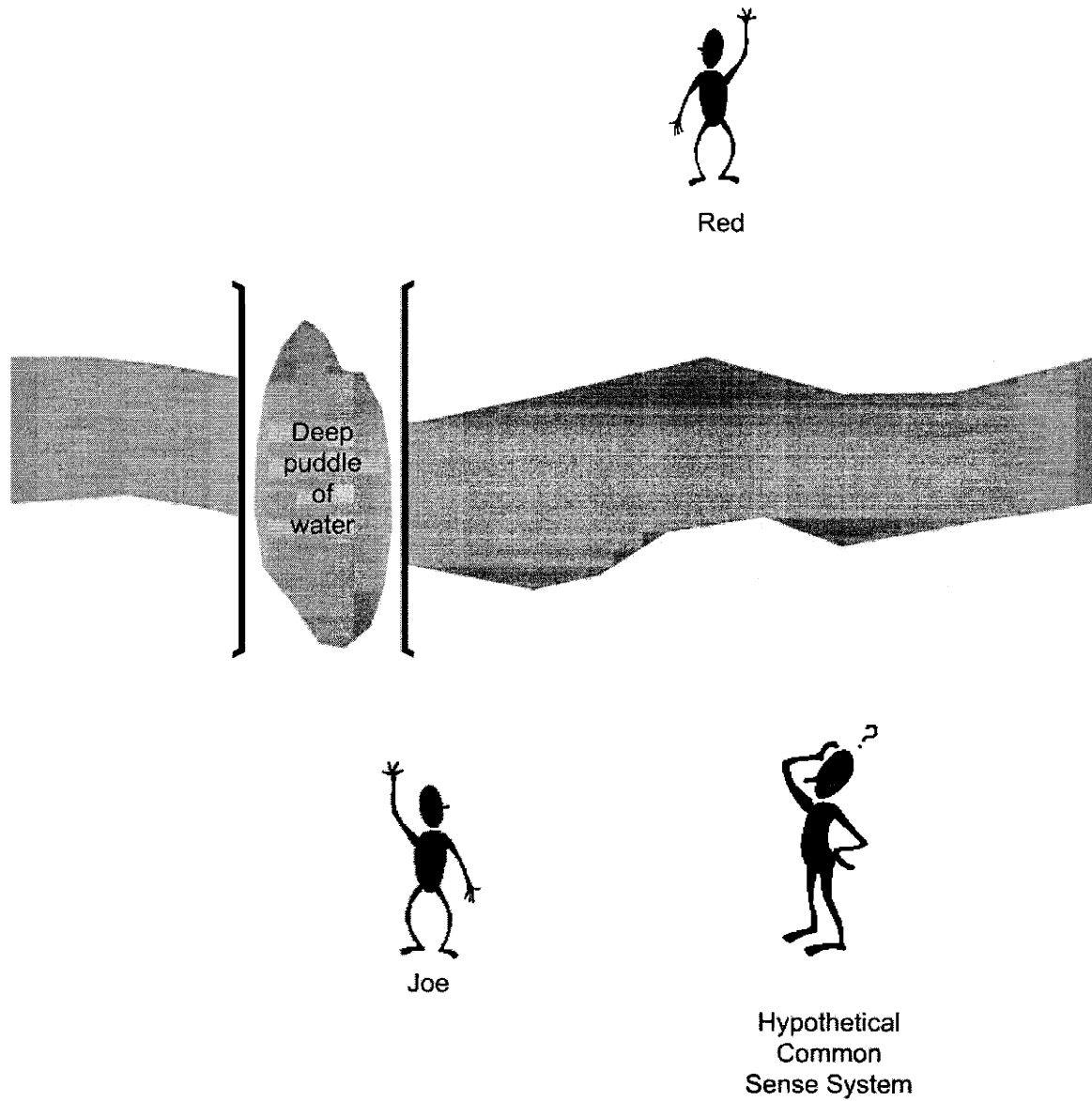


Figure 1. The hypothetical system sees Red and then later sees Joe. It then tries to decide if they are the same person by comparing their properties and by searching for a continuous path connecting the places where it saw them.

Suppose, however, that Joe is perfectly dry when you see him. Obviously, Joe and Red could not be the same person because nobody could have crossed that bridge without walking through the deep puddle and gotten at least somewhat wet. Let us see how the modular system performs. The path search algorithm would allow identity because it is not concerned with dryness and the property comparison algorithm also allows identity because Red was too far away for the system to see if his legs were wet. Thus, both algorithms implemented in a modular system incorrectly allow identity to hold. A modular system would therefore fail to make this simple common sense inference even though it has the appropriate knowledge (path continuity, property persistence) and inference schemes (path search and property comparison) about the objects in this scenario.

Consider the precise nature of the problem. The property comparison algorithm needed information (that if Red/Joe crossed the river, he must have walked through the puddle and wet his feet) that it could only have gotten from the path-search algorithm. However, the path search algorithm does not care about wetness or dryness; it just wants to find a path. Likewise, the property comparison algorithm does not care about the bridge or the river; it only considers the properties of Red and Joe. You could modify these algorithms to consider these other facts, but for human-level common sense, there would be no end to those modifications as anything could be potentially relevant to anything else².

What we need is this: the property-comparison algorithm must recognize and exploit an *opportunity* created during the execution of the path search algorithm. The path search algorithm must be able to *share progress* it makes with the property-comparison algorithm.

² For example, the price of eggs in China could kill someone in Chicago if a commodity futures trader is driving while preoccupied by the price of eggs in China and accidentally hits a pedestrian.. Notice that the price of eggs in China is only separated from the killing by a two-step inference any normal human could make.

As subsequent examples will show, this is not an isolated problem but instead an example of two important principles:

- Inference schemes must recognize and exploit opportunities created by other inference schemes.
- Inference schemes must share progress with each other

Procedures, routines or algorithms in a conventional modular system are not able to share progress because in general they represent the state of their data and the state of their control of the algorithm differently for each algorithm. For example, a typical path search algorithm uses a stack and a tree as its main data and control structures whereas a property-comparison algorithm might use something more like a hash table.

Algorithms in modular systems cannot seize opportunities for the same reason. Even if they are running in parallel and can access the same memory, their incommensurable data and control structures hinder them from monitoring each other's activities for opportunities.

We need a new architecture for realizing fluid inference schemes.

1.3.2 Rules that are sensitive to the situation

The next example, which was suggested by Ken Haase, shows that inference schemes must be sensitive to world knowledge involving the context in which they are used. Part of knowing how to read a standard elevation map, e.g., is know something like:

A bump on a map means a hill.

This works fine in most cases, but not in the desert. Because desert terrain is formed by sand dunes that are shaped by changing winds, a bump on a map means little about the terrain just days after it is created.

We thus have a piece of knowledge that overrides the bump-means-hill generalization for the desert:

In the desert context, a bump on a map does not mean a hill.

This is still not enough and continuing in this vein does not help. Consider the following cases:

- The bump represents a hill that is sacred to the local culture and people who live nearby restore its original shape after the wind changes it.
- The hill is between a ring of buildings that keep the wind from hitting it strongly.
- The hill has been made part of a golf course and is now covered with often-watered grass that prevents erosion.
- The map is five minutes old (constructed, say, from fresh satellite data).

In each of these cases, contrived or not, it is obvious, indeed common sense, that contrary to our rule about the desert context, the bump on the map does mean a hill. This is an example of what it means for a system to *be sensitive to world knowledge pertaining to a situation*. It must adapt its normal inference schemes to the current situation based on world knowledge. This is also an example of the “qualification problem” in common sense research: how do you specify enough of the important qualifications to a piece of knowledge to support human-level common sense but without enumerating all possible cases?

Note finally that that his case is an example of two problems common sense researchers know well: *context* and the *qualification problem*. Many difficulties in building a common sense system can be solved with flexible inference. Subsequent examples will do the same for other problems in common sense research.

1.3.3 Flexible case-based reasoning

When you have a goal, you should try plans that worked for similar goals in the past. This kind of reasoning is often called “case-based reasoning”. The following example will show that case-based reasoning must be flexible by being sensitive to world knowledge.

Suppose you are a short-order cook who normally works the lunch shift. Early on in your job you probably ran out of tomatoes that you needed to achieve your goal of making a sandwich. In the past, you developed the plan of going to the backroom and retrieving a handful of tomatoes from the refrigerator when you need a tomato. Now suppose you fill in for a friend on the night shift and someone asks for a sandwich five minutes before closing time. You reach for a tomato and do not find one. The current goal of having a tomato is similar to the goal of having a tomato during your normal shift and you retrieve your usual plan retrieving a handful of tomatoes.

You should not simply use that plan as it is, however. If you bring back a handful of tomatoes five minutes before closing, you will have to return most of them to the refrigerator. You cannot simply reuse the previous case. You must instead detect a flaw in that case applied to the current situation and adapt it to achieve your goal.

Like the previous example, this case illustrates that inference schemes must change according to the details of a situation. Where the last example had more to do with the structure of knowledge, this one involves an important artificial intelligence inference technique.

1.3.4 Assembling new inference schemes for a problem

An extreme case of adapting inference schemes to a particular situation occurs when there is no relevant inference scheme to begin with. We are very good at constructing procedures for novel tasks as they arise. Consider a trip to the bookstore to find Isaac Asimov's *I, Robot*. You go to the science fiction shelf and face one of several possibilities:

- *The books are ordered by author.* You start looking for *I, Robot* from the left because "A" is at the beginning of the alphabet.
- *The books are ordered by title.* You look towards the middle.
- *You forget the exact title and author, but remember that your friend's copy was red.* You scan for red books.
- *You forget the exact title and author, remember it is red, and encounter many red books.* You systematically scan through them left to right to keep track of the books you have already scanned.

When the circumstance varies, you easily craft a scheme of inference that generates a more or less optimal search procedure. Crafting new inference schemes, effortless as it seems, is an important component of common sense that is difficult to implement.

1.3.5 Inference schemes must interrupt each other gracefully

A theme throughout these examples is that inference schemes must be able to interrupt each other without causing any serious damage.

For one inference scheme to exploit opportunities created by other inference schemes or share progress with other inference schemes, it must often take control of the system while another inference scheme is operating. In typical computer systems, parts of a

process create structures that other parts of the process presuppose. If those structures are not exactly right, then the system can often stop working completely. When several processes work on the same problem and interrupt other processes in the middle of their operation, this kind of crash becomes difficult to avoid. One solution is to insulate the data and control structures of one process from another inside an “environment” local to the process. Localizing those structures does not work in common sense systems because it would rule out inference schemes coexisting flexibly. Inference schemes with no access to the data of other inference schemes cannot share progress or exploit opportunities. The architecture which I outline in this thesis solves this problem by proposing a common set of *atomic* computational systems and operations that are *not* interrupted.

1.4 Summary

Most artificial intelligence researchers work within a framework or class of computational techniques. To the extent that their goal is human-level intelligence, they try to reach it by generalizing and extending their framework. Through my experience and my reading of history, I suspect that in order to understand and create human-level intelligence, we will need to learn to integrate multiple frameworks more than we will need to find one grand unifying framework.

The integration problem is very difficult because modules built within each computational framework must share progress, recognize and exploit opportunities created by each other, adapt to situations using world knowledge and interrupt each other without hazard.

The research reported in this thesis addresses these problems. Its main contributions are:

- Three principles for organizing a cognitive architecture and three corresponding architectural choices I have made in developing Polyscheme:

1. First, different (aspects of) situations that intelligent systems must deal with are best modeled with different schemes for representing knowledge and making inferences. Polyscheme includes several “specialists” such that each models a particular aspect of the world with its own (possibly unique) representation and inference techniques. I describe these in chapter 3.
 2. Specialists must communicate with other specialists frequently so that each specialist uses the most complete, accurate and relevant information when it deals with a situation. Specialists in Polyscheme communicate and combine information by simultaneously concentrating on the same focus of attention. I motivate the focus of attention and explain how it works in chapter 4.
 3. Because information about some aspects of a situation is more important than information about other aspects and because the order that specialists focus on those aspects is important, a system of focused specialists must have mechanisms that decide where to focus. Polyscheme’s specialists, especially the reflective specialist, guide the focus of attention and thereby implement inference schemes using an “attraction” mechanism to specify their preferred foci. Chapter 5 describes these mechanisms.
- Polyscheme enables multiple inference techniques to be integrated in dealing with a situation because each inference technique can be implemented with one or more focus schemes. I describe how to implement several important inference techniques (e.g., script matching, backtracking search, reason maintenance, stochastic simulation, and counterfactual reasoning) as focus schemes.
 - I have used Polyscheme to implement the S6 system for common sense physical reasoning. S6 views interactions in a simple physical world through a 2-dimensional projection of that world. S6 keeps track of the identity of objects, infers the character and existence of events it cannot see, predicts the outcome of events, explains events and nonevents, and revises its inferences when it receives

new information. S6 successfully reasons about many scenarios researchers present to infants and young children in order to study their knowledge of the physical world.

S6 combines specialized representation and inference techniques for identity, time, events, causality, space and paths to successfully deal with a wide range of situations. The knowledge representation schemes S6 uses include scripts, frames, logical propositions, neural networks and constraint graphs. The inference schemes S6 implements include script matching, rule matching, backtracking search, neural network propagation and counterfactual reasoning. In chapter 2, I show that these representation and inference schemes form part of a common sense substrate that underlies much of human cognition. The success of S6 therefore demonstrates that Polyscheme is important for understanding and building intelligent systems in any domain of human cognition.

Chapter 2

Common Sense Physics and the Substrate

Before I describe the architecture I have developed to integrate different representation and inference schemes, I will explain why I chose to develop the architecture in the context of common sense physical reasoning. Physical reasoning requires a “common sense substrate” that includes reasoning about time, events, identity, uncertainty, causality, and more. Because almost any nonphysical domain uses these, a robust implementation of this substrate would be an important advance towards understanding intelligence. This chapter therefore shows that the architecture I describe in the rest of the thesis is a not just a specific solution to a specific problem, but a fundamental advance in our understanding of intelligence.

2.1 Reasoning about physical events requires most of artificial intelligence

I originally chose common sense physics because I found that it required enough intelligence to require important innovation, but that it was easy enough to be tractable. I began to see this when I considered particular examples of reasoning that I could not understand. To simplify my thinking, I would imagine a similar example in a domain that was simpler than the one I had started with. The domain that most often seemed easiest to think about, but which was still productive, was common sense physics. I ultimately found that I could find a common sense physics example of almost any problem in artificial intelligence. In this section, I will present some examples of this and in the next I will offer some insights from linguistics and psychology that further relate common sense physics to cognition generally.

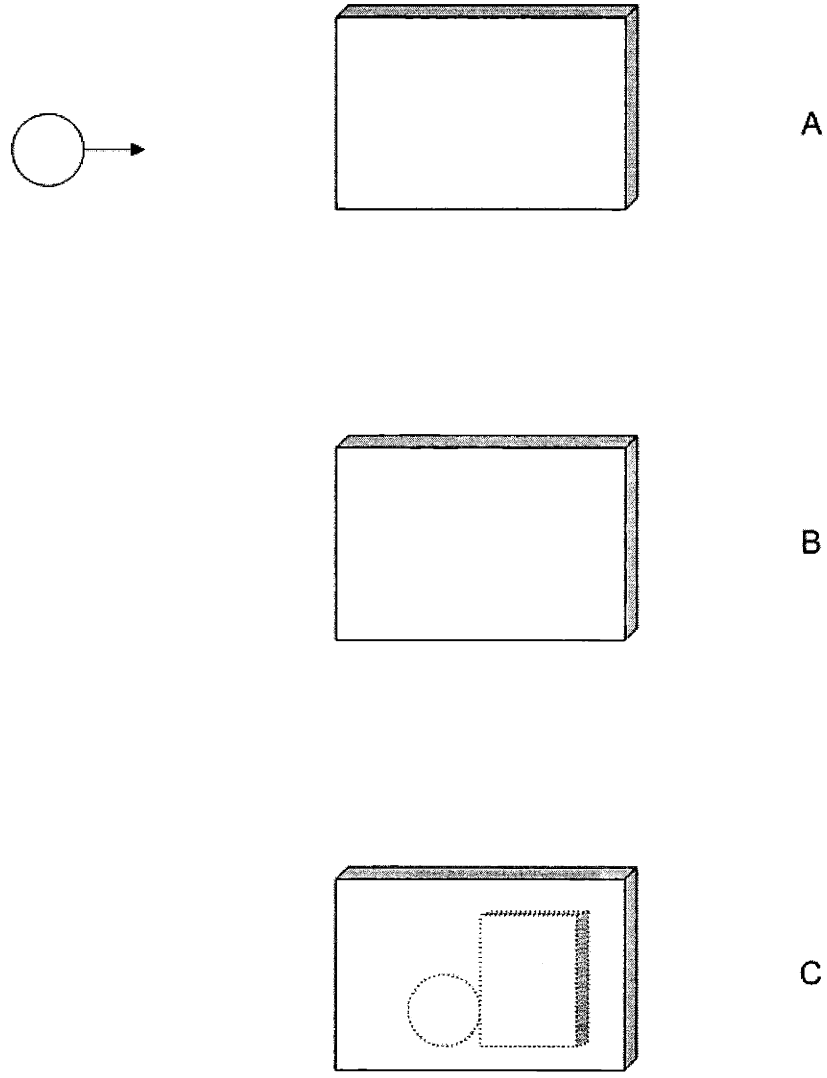


Figure 2. The ball rolls behind the screen (A), but does not roll out (B). There must be an object behind the screen that blocked it (C).

2.1.1 Deduction, falsification, default reasoning and explanation

Researchers using logical approaches to artificial intelligence have encountered difficult issues regarding deduction, falsification, default reasoning and explanation and they have constructed many sophisticated logical theories to deal with them. The following example shows that even the simplest physical interactions involve these issues.

In Figure 2a, you see a ball rolling towards a screen. In Figure 2b, you see the ball roll behind the screen. In screen 2c you see that the ball does not emerge from the screen and you see an object that is posited to have blocked the ball.

You can crudely formalize the inference that the ball should come out of the screen thus:

```
At(ball, left-of-screen, t1) +  
Moving(ball, right, t1) + Empty(behind-screen) →  
At(ball, behind-screen, t2) + Moving(ball, right, t2).
```

```
At(ball, behind-screen, t2) +  
Moving(ball, right, t2) + Empty(behind-screen) →  
At(ball, right-of-screen, t3) + Moving( ball, right, t3 )
```

The inference that the ball comes out of the screen depends on the assumption that:

```
Empty(behind-screen) .
```

When the ball fails to come out from the screen, you infer that the proposition,

```
Empty(behind-screen),
```

is not true and that there must be something behind the screen blocking the ball:

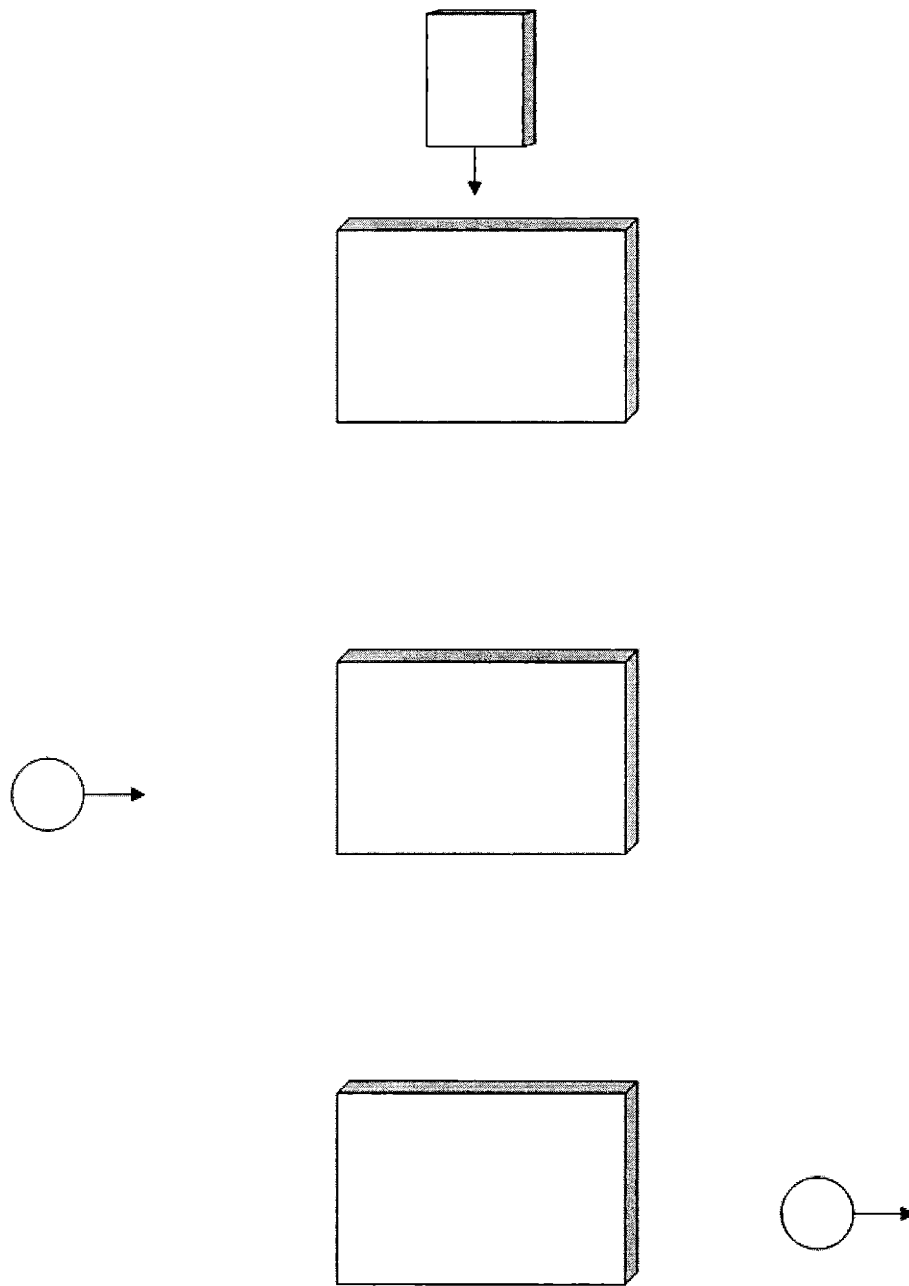


Figure 3. You know that the ball that rolls out to the right is different from the ball that rolls in from the left because the block behind the screen would keep the left ball from rolling out.

At (something, behind-screen, t2) +
something != ball.

There are many issues involved here: what can you assume and why; what does it take to falsify an assumption; when there is more than one explanation for an event; which do you chose; etc. These are the usual issues surrounding explanation and default reasoning and they occur whenever you try to build a good reasoner for even the simplest common sense physics interactions.

2.1.2 Temporal reasoning

Physical interactions are events that happen in time and the outcome of most events depends on how they relate temporally – before, after, during – other events. Figure 3 shows a simple example.

In Figure 3, a ball rolls behind a screen, a ball that looks the same rolls out. Then someone places a brick behind the screen. Because the brick was placed behind the screen *after* the ball-rolling event, you can assume that the space behind the screen was empty *during* the ball rolling event and that the ball that emerged from the screen is the same as the ball that moved behind the screen.

Figure 4 presents the same scenario, except that the ball rolls behind the screen *after* someone puts the brick behind the screen. In this case the ball that emerged from the screen cannot be the same as the ball the rolled behind the screen because balls do not move through bricks. The first ball probably transferred momentum to the second ball by hitting the brick, causing the second ball to move out from behind the screen.

This is just one example of a very simple physical inference that depends on more than one temporal relation.

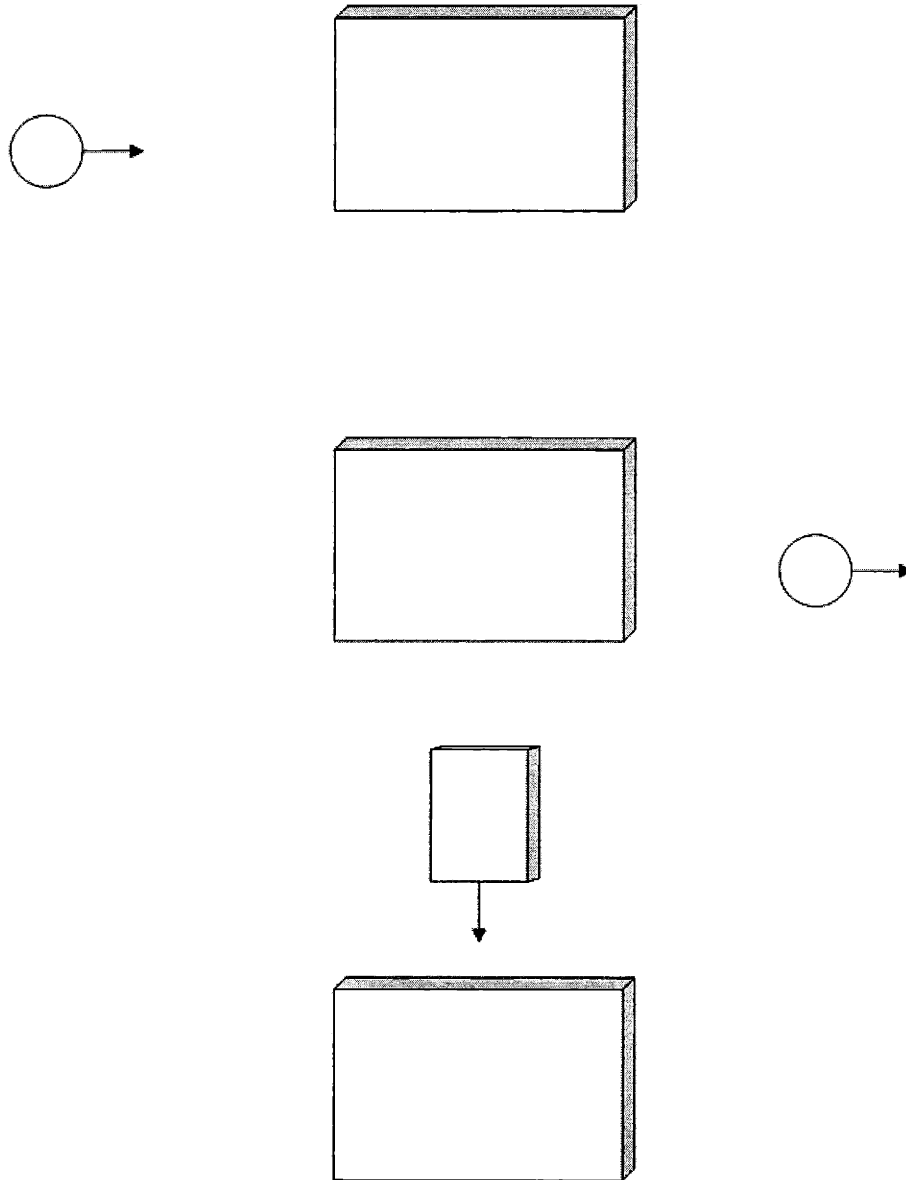


Figure 4. The ball that comes out from the right of the screen can be the same ball that went behind the screen from the left because there was nothing behind the screen to block its motion. The block did not go behind the screen until *after* the ball did.

2.1.3 Belief revision and reason-maintenance

Any system that reasons in almost any nontrivial domain must often infer or assume facts that it must later revise. Because the system could have inferred more facts based on the originally assumed fact, revising its belief about the original fact is much more complicated than simply retracting it (Doyle 1979, 1992). The system must retract all beliefs it inferred using the original fact that are not otherwise justified. Building systems that can revise their beliefs correctly has been a challenge for artificial intelligence researchers, even for those trying to build good models of common sense physical interactions.

Consider an example. Figure 5a shows a scene where a screen occludes a table. A block is dropped above the table, it falls behind the screen and you infer that it comes to rest on the table. Then, when you are told that there is not just one table, but that there are two separated tables, as in 5b, you must revise your belief about where the ball went when it fell behind the screen. In this case, you assume it fell on the floor.

2.1.4 Searching through problem spaces

Events often have more than one possible outcome and systems can usually execute more than one action at any given time. The sequence of possible actions and/or inferences about event outcomes creates a huge “problem space” of possible world states and an intelligent system must choose a sequence of actions and/or inferences to achieve an adequate state. Many innovations in artificial intelligence have involved finding ways to deal with the problem of huge search spaces. Backtracking search algorithms eliminate whole regions of the problem space that involve an intermediate state that is somehow bad or harmful. Heuristic search algorithms improved on backtracking search by creating domain-dependant evaluation schemes that could chose the most promising actions without evaluating all their consequences. Frame and script systems created larger-scale structures that a system could use to construct partial solutions fast without exploring a huge search space.

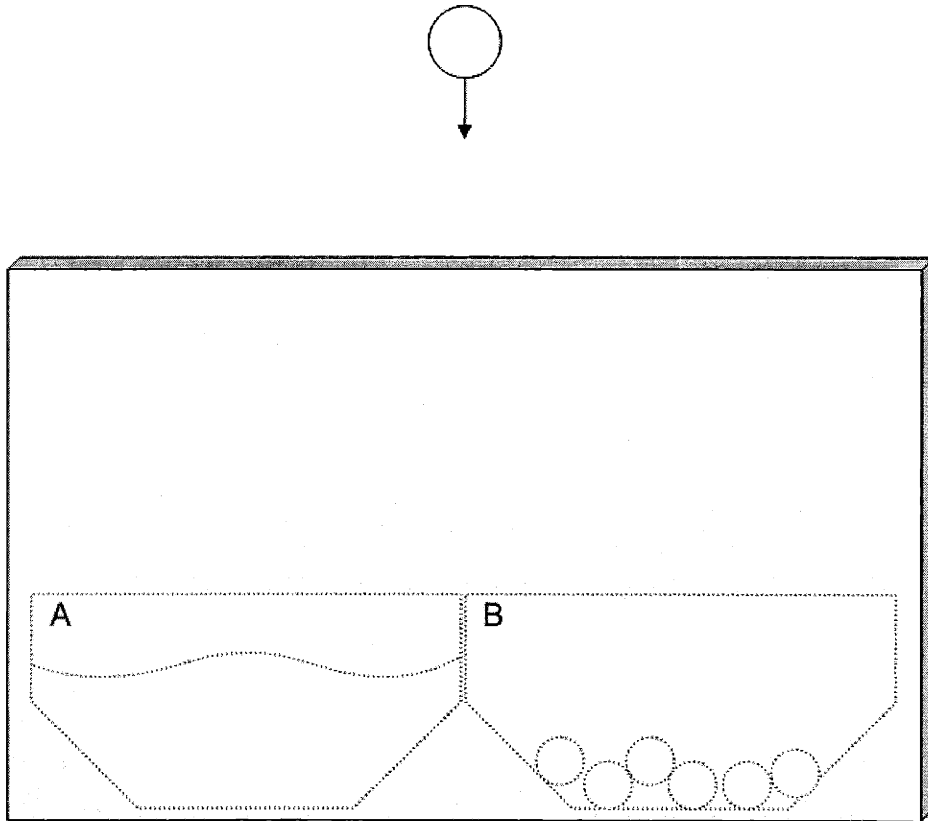


Figure 6. Bucket A is filled with water and bucket B is filled with hot coals. The ball falls into one of the two buckets.

In this section, I present an example that shows that even simple physical interactions involve problem spaces where many indeterminate actions and facts need to be resolved.

Figure 6 illustrates a simple physical interaction that requires backtracking search. Behind the screen in Figure 6 are two buckets. On the left, bucket A is filled with water and on the right, bucket B is full of hot coals. Figure 6 also shows a ball falling behind the screen. The ball is white and shaped roughly like a ping-pong ball, though it may be a regular rubber ball. You see the ball fall behind the screen, though you neither see nor hear any other sights or sounds.

If your task is to figure out if the ball fell into bucket A or B, you might imagine that the ball fell into bucket B and infer the consequences. To infer the consequence of landing in bucket B, you need to know if the ball is rubber or if it is plastic. You can imagine that it is rubber, infer that you would smell burning rubber, remember that you do not and therefore conclude that the ball is not rubber if it fell in B. Likewise, you can infer that the ball is not plastic if it fell into B because when you imagine a rubber ball lying in burning coals, you imagine a certain smell that you do not perceive. So if the ball fell behind B, it is neither rubber nor plastic. But you know it was one of these, so you know that the ball did not fall into B, but instead fell into A.

2.1.5 Probabilistic inference

In many domains where events have more than one possible outcome, some are more likely than others. When a scenario involves several possible series of outcomes, it is often difficult to conclude decide which of the possible scenarios are most likely.

Attempts to make these decisions are often called “uncertain reasoning” or “probabilistic inference”.

Imagine an example like the last one, with the only difference being that you know more about the probabilities of each uncertainty. Bucket A takes twice the area of bucket B

and the odds that the ball is plastic are 5:1. You are certain that you did not hear a splash, but are uncertain whether you smell any new smells. What are the odds the odds that the ball is in A and what are the odds that it is in B? This is therefore an example of a probabilistic inference scenario that shows up in simple physical reasoning.

2.1.6 Common sense physics and “high-level” reasoning

Thinking of these problems in the physical domain is not enough to solve them, but it does show that that the simplest common sense physical inferences involve a surprising amount of artificial intelligence problems that many have previously thought of as a part of “high-level” reasoning. Further, these examples show that any system that is a robust common sense physical reasoner will have to include mechanisms that will help us understand common sense reasoning outside of the physical domain.

2.2 Linguistics and Psychology

As I learned more about linguistics and psychology, I became less surprised that one could use the domain of common sense physics to productively think about so many issues in artificial intelligence. Several strands of linguistics and psychology have demonstrated that much of human cognition has many deep parallels to spatial and physical reasoning.

2.2.1 Language Semantics

Much work in the study of linguistic semantics has shown that many “semantic fields” are analogues to the semantics of physical and spatial fields. Ray Jackendoff’s (1990) synthesis of much of this work and Leonard Talmy’s work on “force dynamics” (1988) provide two compelling examples.

Transfer of Motion. Jackendoff's treatment of verbs that refer to transfers of motion introduces a set of primitives to explain semantic regularities that occur in more than just the physical domain. These primitive notions include *cause*, *go*, *path*, *to*, and *from* and are common in many other frameworks (see Shank, 1972 and Miller and Johnson-Laird, 1976).

For example, you can represent³ the meanings of "John entered the room" and "John left the room" respectively as:

```
GO (John, [path [to: room]])
GO (John, [path [from: room]])
```

With the same primitive notions, you can represent a change of state.

"John became drunk".

```
GOstate (John, [path [to: drunk]])
```

"John sobered."

```
GOstate (John, [path [from: drunk]])
```

Or, temporal extent:

"Class extended to 9pm".

```
GOtime (class, [path [to: 9pm]])
```

"Class began at 9pm."

```
GOtime (class, [path [from: 9pm]])
```

Or, transfer of possession:

"John received \$100."

```
GOpossession ($100, [path [to: John]])
```

³ I am simplifying Jackendoff's notion somewhat to make it more readable to the uninitiated.

“John lost \$100.”

$GO_{\text{possession}} (\$100, [\text{path} [\text{from: John}]])$

Each example uses the same primitives with the same structures. The only difference is the subscript on GO. Jackendoff’s work describes the semantics of a large set of word classes with only a few more primitives, each of which you find in the domain of common sense physics.

Force Dynamics. Leonard Talmy’s work on “force dynamics” in language shows that there is a sophisticated notion of force that occurs in several semantic fields. The semantics of much causal language seems to imply a notion of an agonist and antagonist. The agonist has a propensity to do something or stay a certain way while the antagonist helps, hinders or is neutral to the agonist’s propensity. You can characterize most any causal verb by characterizing its agonist, its antagonist and how the two relate to each other. For example, the verb “help” implies that the antagonist is contributing a force that contributes to the agonist’s tendency. “Hinder” implies that the antagonist is opposing the agonist’s propensity. Here are some more examples:

- Mary *urged* John to leave. (He was reluctant.)
- Mary *persuaded* John to leave. (He relented.)
- Mary *forced* John to leave. (He would not leave.)

This agonist/antagonist structure occurs not only in verbs of physical causation, but in many other semantic fields. The following table shows a few.

Domain	Agonist	Antagonist	Sentence
Physical	Block	Ball	Block keeps ball from leaving the house.
Psychological	John	Fear of enemies	John cannot go out of the house (because he is paranoid).
Legal	Mary	The law	Mary is not allowed out of the house.
Cultural	John	Cultural taboo.	John cannot leave the house (so soon after the funeral).

These examples are only a small sample of the parallels between physical and nonphysical reasoning that Jackendoff, Talmy and others find in their work.

That a handful of primitives that have such obvious physical analogs, if not origins, underlies a large swatch of semantics suggests that understanding common sense physical reasoning will help us to better understand common sense reasoning more generally.

2.3 The common sense substrate

This chapter has shown that many aspects of cognition often have very deep parallels to common sense physical reasoning. Many traditional artificial intelligence problems occur when you try to build a robust reasoner for even the simplest common sense physics problems. Many nonphysical semantic fields are very similar to physical semantic fields. What should we conclude from this? There are at least three possible hypotheses.

1. Humans have a fundamental set of common sense reasoning machinery for dealing with notions of time, place, identity, causality, etc. Common sense physics and nonphysical common sense are similar because they both use the same machinery widely.
2. Humans have a fundamental set of common sense physical reasoning machinery that is good at reasoning about physical events, physical causes, the identity of physical objects, the temporal relations between physical events, etc. Common sense physics and nonphysical common sense are similar because we map many parts of nonphysical reasoning to physical problems and use our physical machinery to solve them.

3. Physical and nonphysical common sense are so similar because they use copies of the same machinery that are modified enough to specialize in their domain, but not modified so extensively that their similarity is not manifest.

I suspect that some combination of each alternative is true. There is probably some basic machinery that is used widely. There are also probably several copies of much machinery specialized for particular domains. There is probably also an extensive analogy system for mapping new domains that evolution could not have accommodated to.

Exactly what reasoning abilities are used widely enough that we should consider them part of a common sense substrate? I have found no way to enumerate these abilities, but I will present here a list that I have developed based on my personal experience. After only a little reflection, it is fairly obvious that most of the notions on this list are involved in several realms of cognition. In the least obvious cases, I use a few examples to illustrate how a particular concept or ability occurs in several cognitive domains.

1. *Objects*. Most every realm of cognition involves objects. Nonphysical objects differ from tangible objects only in not having properties you can perceive through visual, haptic, or some other form perception. There is also a set of questions you asked about an object regardless of what domains they are in: what category does it belong to, which of its properties persist through time and which do not, is it the same as another object you perceived or inferred in the past, what happens when such and such action is taken on it, etc.
2. *Events*. Again, events that occur over a temporal interval are a common aspect of most any domain, and so are many of the questions you would ask about them: what caused the event, what was the result of the event, how long did it take to occur, what were its parts, etc.
3. *Identity*. When we perceive or infer the existence of an object, we often need to ask if the object is the same as any object or event that we contemplated in the past.

- *Physical Object Identity.* When you turn your gaze away from an object and then return your gaze and see an object, you are inferring that both object glimpses were of the same object. When you encounter an object and then encounter a similar object at some point in the future, you need to ask if both encounters involved the same object.
 - *Cross-modal object identity.* When you hear a bird singing and then turn your head to see a bird, you can ask if the bird singing is the same as the bird you see.
 - *Event Identity.* In investigating a crime, you must often ask if a gunshot event from a particular gun (which, say, you detected using forensic analysis) is the same as the gunshot that wounded a victim.
 - *Language.* In understanding language, you are constantly answering identity questions: Is the reference of this pronoun the same as the reference of that noun phrase, is the gift for the party Mary mentioned the same as the kite that Jack said he is thinking about buying, etc.
 - *Nonphysical realms.* Is the cash transfer on such and such date the same as the transfer the suspects were overheard talking about?
4. *Time.* Any realm of thought that involves the occurrence of events will require some notion of time. Even though the scale may be different, the questions you ask and inferences you make in each domain have much in common: how long did such and such even last, if event A occurred during event B and event B preceded C, then A must have preceded C, etc.
5. *Properties, attributes, relations.* Objects in all realms have attributes, properties and relations. Physical objects have sizes and shapes and they relate to other objects spatially; checking accounts have sizes and owners and are related to certain

financial institutions, etc. In all realms, we can ask, e.g., if relations are transitive or associative or what the inverse of a relation might be.

6. *Places.* The last section showed how useful it was to represent many nonphysical aspects of the world in terms of an (often imagined) location in space and change thereof.
7. *Causality.* Events cause other events in most every realm of cognition. For example, when one physical object strikes another, the striking usually *causes* a sound event, a motion in at least one of the objects, and perhaps even a fracture in one or both of the objects. Rising interest rates *cause* companies to make fewer capital investments. The rise of enlightenment ideas *caused* art to be more secular, etc. In any realm, we can ask, what caused an event, how long did it take for the cause to have an affect, etc.
8. *Categories.* Objects and events in all realms belong to categories that predict the properties or attributes an object has, how it relates to other objects and how it behaves under certain circumstances. For example, animate physical objects can move without external causes, inanimate objects to do not and every word in a language belongs to a category that predicts what other words it can appear with and in what order. In all realms, categories can belong to one or more broader categories so that categories form a heterarchy.
9. *Mereonomy.* Objects and events have parts. Trees, e.g., usually have a trunk, branches, leaves, and roots. Football games are events that are divided into quarters, each of which includes a series of play events. Mathematical theories have axioms and theorems and proof of theorems. Relationships can have introductions, courtships, etc.
10. *Number and extent.* Any domain that has individual objects can use numbers. You can ask how many blocks are in a bucket, how many branches there are of the United States Government, and how many equations form the basis of classical

electromagnetism. Quantity and extent are also broadly used. Butter can extend over the surface of a piece of bread to various degrees; the right to speak freely can extend through a population at various degrees, etc.

11. *Sets*. It is often useful to think of several objects or events together in a group. This is as clear in the physical domain (“all the apples from that orchard are rotten”) as it is in less tangible domains (“long-distance relationships do not work”, “growth stocks are not performing well”, etc.)
12. *Agency*. People, and perhaps some animals, have beliefs and goals and take actions they believe will achieve their goals. We can also usefully think of corporations, groups, cultures, mobs, civilizations, etc. as having beliefs and goals and taking purposeful action. Given an action and an agent, we can ask, why did the agent take that action, what can we infer about what it believes about the world to have taken such an action. In understanding language, much of what we try to do is to understand what a speaker tries to accomplish with an utterance.

In the research I report in this thesis, I have built reasoning mechanisms for the first seven concepts on the list.

I call the collection of these aspects of thought the **Common Sense Substrate**. There are four important points to make about the substrate and all that I have discussed in this chapter.

- The substrate deserves a disproportionate amount of attention from common sense researchers. Because the concepts in the substrate are a part of so many cognitive domains and so many reasoning problems, researchers would do well to focus more attention on these items than most others. Building common sense systems about botany, for example would probably give you much less insight into the rest of common sense reasoning than building a robust temporal reasoner.

- The list is relatively small. I suspect that the list I present above encompasses most of the near-ubiquitous common sense reasoning abilities. At worst, I think the size of the list would not be more than doubled or tripled, though there would be more refinements within each element. This is encouraging for those of us who would like to see human-level intelligence in our lifetime.
- Building a system that can reason about these substrate notions at the level of a human would be a major step towards producing human-level intelligence. I suspect that once you have the substrate implemented at this level, the remaining steps towards achieving machine intelligence will be to build systems for making analogies to substrate elements and to find ways of acquiring knowledge about the world that you would describe in terms of the substrate. Large problems, to be sure, but having a good substrate implementation would help greatly.
- The work I present in this thesis implements a significant portion of the substrate in the context of common sense physical reasoning. Because the substrate is so ubiquitous, I suspect that the architectural innovations of this thesis will be broadly useful outside the realm of common sense physical reasoning.

Chapter 3

Specialists

In this and the next two chapters, I will describe the Polyscheme architecture for combining several representation and inference schemes into one system. In these three chapters, I will motivate each major feature of the architecture and describe how I implemented it in the context of S6.

The architecture has three components:

- Specialized processors, called **specialists**, which use different methods of inference and representing knowledge to make inferences.
- An integrative **focus** of attention that the specialists use to communicate with each other and synchronize their beliefs.
- A **reflection specialist** that decides where to shift the focus of the specialists.

In this chapter, I briefly explain why the architecture needs several specialists, I motivate the characteristics all specialists in the architecture must have and then I describe the particular specialists that make up S6.

3.1 Why specialists

The first chapter explained why I decided to tackle the problem of integrating several kinds of computational techniques and representations into one system. I will not recapitulate that discussion here except to repeat that, despite much effort, no one has

found one scheme for representing and making inferences about the world that subsumes the functionality of all the current schemes. I thus set out to develop an architecture for developing intelligent systems that incorporate several different schemes of knowledge representation and inference.

In this architecture, each kind of representation along with its associated method(s) of inference is encapsulated within a specialist. All specialists have the features I describe below and the next two chapters will describe other features that all specialists must implement.

- *Specialists have specialized internal representation(s) and algorithms(s).* Because different aspects of the world are best modeled with different representation and inference schemes, each specialist uses its own special representations and algorithms for performing its function.
- *Specialists must offer opinions on other specialists' assertions about the world.* Because a specialist is an expert in some aspect of the world, other specialists will need to have the other specialists' opinion on some facts before it proceeds to make inferences on those facts.
- *Specialist must be able to communicate what they need to know to other specialists.* Because a specialist's expertise involves only one aspect of the world, it must be able to ask other specialists for information.
- *Specialists must communicate new inferences to other specialists.* When a specialist makes a new inference, other specialists might need that information and might be able to improve upon it.

Most of these features require that specialists have some protocols for transferring opinions and requests for opinions between each other. I will describe those in the next chapter.

3.2 The specialists in S6

The following subsections describe the specialists in S6. Some of the specialists I describe here were actually implemented in S6 as more than one specialist so that the programming project would be more manageable. However, there was no theoretical or computational reason to do this, so I individuate specialists here as I would have implemented them had I not been concerned with writing readable, reusable code. All the functionality I describe, however, has been implemented.

3.2.1 Perception Specialist.

This is the only specialist connected to the “outside world”. The world is a 6x6x6 grid of cells that are either totally occupied or totally unoccupied by objects of certain colors, shapes and substances. At every time step, the perception specialist receives a two-dimensional projection of the world. The projection is always from the same point and is a 6x6 grid. Objects are labeled with color, size, and shape information. If an object is occluded by another object at a certain time step, the perception specialist does not see it at that time step. If the perception specialist sees an object at T1 and then at T2, each object is labeled with a different identification number so that the system is not given identity information, but must infer it. The only two exceptions are when (1) an object remains visible and in the same position over two adjacent time steps and (2) an object remains visible and moves to an adjacent location in the next time step. This models “apparent motion”. Otherwise, the task of deciding which object glimpses are of the same object is left to S6.

3.2.2 Attribute Specialist.

The attribute specialist performs two functions:

- *Detect changes.* Because some attributes of objects can change over time, you need to detect changes in an attribute's value so that other specialists can reason about those changes. When the attribute specialist detects a change, it infers the existence of a change event and lets other specialists better characterize that event. For example:
 - When an attribute changes, it is often because something caused the change. Therefore, other specialists must help explain the change.
 - Changes in the location of an object imply a continuous path between both locations.
 - A change in some attributes might cause other changes. For example, when a light changes from being off to on in a room, people in that room often wake up.
 - The kinds of changes an object undergoes often tell you more about what kind of object it is. For example, if a metal object spends several months exposed to rain and does not rust, then it is probably not made of iron.
- *Enforce uniqueness.* An attribute of an object can only have one value at any time. Thus, if you know that A weights 11lbs at time T1 and B weights 17lbs at time T1, then you know that A and B are not the same object. Uniqueness also in reverse: if the weight of C is 11lbs at T1 and 5kg at T1, then you know that 11lbs and 5kg are the same weight.

3.2.3 Physics Specialist.

Physical objects cannot occupy the same place at the same time; unsupported objects fall because of gravity; supported objects do not fall, etc. The physics specialist uses a combination of rules and trans-frames (Minsky, 1986) to make inferences based on these regularities.

3.2.4 Causal Specialist.

Events often cause other events. When you detect an event you often need to find its cause and predict its effects. For example, when a sugar cube comes to rest in a puddle of water, it (at least partially) dissolves. The causal specialist in S6 makes all the causal inferences that are not made by the physics specialist. The causal specialist uses a representation much like trans-frames. When this specialist detects an event, it finds trans-frames for events that might have caused the detected event and trans-frames that might follow from it.

3.2.5 Temporal Specialist

The temporal specialist maintains constraints among temporal intervals using an approach very similar to Allen's (1983) framework.

This specialist also infers facts using temporal relations with other facts. For example, if a block was green over all of time T and time T includes time T1, then the block was green at time T1. The temporal specialist uses temporal relations between facts such as these to answer questions and to ask the other specialists for facts that might help it answer questions.

3.2.6 Identity Hypothesis Specialist.

Because S6 receives almost no information about the identity of the objects it perceives, it must constantly wonder if the object it is glimpsing at a particular time is the same as such and such object that it glimpsed at some time before that. For example, when a red ball goes behind a screen and a red ball comes out, S6's perceptual specialist merely tells the other specialists that it sees a red ball, not that this ball is the same ball that went behind the screen. One first step to establishing identity is to generate hypotheses about which object glimpses might be of the same objects. The identity hypothesis specialist does this mostly by using a very simple neural network (i.e., no hidden layer and

Boolean input and outputs). Subsequent versions of S6, could easily use more sophisticated techniques here, including decision trees, multi-layer neural networks, etc. One nice feature of this architecture is that you can improve the functionality of a specialist by modifying its internal representation and inference techniques without modifying other parts of the system much or at all.

3.2.7 Identity Specialist.

The identity specialist fulfills two main functions regarding the identity of objects:

- *Constraint Maintenance.* If A and B are the same object and B and C are the same object, then A and C are the same object. If A and B are the same object and B and C are different objects, then A and C are different objects. The identity specialist maintains constraints like these on the identity of objects. It uses a simple constraint graph and constraint propagation algorithm.
- *Identity Fusion.* If A is red at time T and both A and B are the same object, then B is red at time T. The identity specialist fuses properties of object glimpses.

3.2.8 Place Specialist.

The place specialist has several functions:

- It knows which places are adjacent to which other places.
- Given a set of constraints on a hypothetical place, it gives all the possible real locations it might be. For example, if P1 is adjacent to (0,0,0) and to (2,0,0), then P1 must be (1,0,0).
- The place specialist makes inferences based on the premise that an object can be in only one place at one time.

- If two objects are in the same place at the same time, then they must be the same object.
- If two objects are in different places at the same time, then they cannot be the same object.
- If an object is in two places at the same time, then the places must be the same.
- If an object is in one place at a time and not in another, then the two places must be different.

The place specialist uses two hash tables to keep track of which objects are in what locations and which location an object is in. It uses a rule-based system to make the uniqueness constraint inferences, though a more sophisticated version of this specialist (that reasons about places of different shapes and sizes) would use more sophisticated computational techniques.

3.2.9 Path Specialist

Whenever you see an object at one position and then again at another position, you know that there is a continuous path between the two objects. The bridge example in the first chapter showed that finding that continuous path often helps you make inferences about the character of the object. If you fail to find a continuous path between two locations where you have sighted objects, then those two sightings are not of the same object.

When the path specialist learns that an object changes its location from P1 to P2, it posits intermediate points that the object may have traveled. (Other specialists rule in or out those intermediate points.) If all intermediate points are eliminated, then the path

specialist infers that the object at P1 is actually a different object from P2 despite appearances.

To posit the intermediate points, the path specialist uses path scripts to connect two points, P1 and P2. For example, the “left-adjacency” posits a point P’ that is adjacent to P1 and it posits change-of-position events between P1 and P’ and P’ and P2. The script only posits an adjacent point, but does not say which particular point it is. Other specialists suggest and eliminate possible points adjacent to P1 that might be part of the path.

3.3 Summary of the representation and inference schemes S6 uses

The following are the knowledge representation schemes along with their associated inference schemes that S6’s specialists use:

- Scripts and script matching.
- Trans-frames and frame-matching.
- Hash Tables and hash table lookup.
- Constraint graphs and graph constraint propagation.
- Rules and rule matching.
- Neural networks and forward propagation.

Notice that some common algorithms in artificial intelligence research such as backtracking search and belief network propagation are not on this list. That is because these are not implemented as specialists but as schemes of guiding the attention of specialists. I will describe these schemes in chapter 5.

Chapter 4

Focusing Specialists

The previous chapter described the need to have several specialized processors with their own schemes for representing and reasoning about the world. In this chapter, I describe several computational factors that suggest specialists should concentrate their attention on the same object, attribute, relation or event at the same time. Polyscheme has a **focus** of attention that reflects this principle. I explain how the focus is implemented in S6 and present psychological evidence for an integrative focus of attention in humans.

4.1 Specialists must focus on the same object, attribute, relation or event

The following principles suggest that specialists must synchronize and concentrate their activities tightly:

1. *Many specialists can always help to confirm, deny or elaborate on a fact.* If a specialist assumes, infers or wants to know a fact, a surprising proportion of the other specialists might be able to help. For example, assume that a physics specialist wants to know if a particular region of space, P , is empty at a particular time, T , so that it can decide whether to predict that something can fall through P . Several specialists can contribute:
 - If T is the present, then the *perceptual specialist* may be able to see P and indicate whether it is empty or not.
 - The *place specialist* may remember that P was empty at T .

- The *attribute specialist* may remember that P was empty at a time, T1, which was just before T and project the *empty* attribute of P forward to T.
- The *causal specialist* may infer that P was empty at T because something had just fallen through P just before T and that it could not have done so if P was not empty.
- Some specialist might know that P was empty at T2 and the *temporal specialist* might know that T2 includes all of T and infer that P was empty at T.
- Some specialist might know that P1 was empty at T and the *identity specialist* may know that P1 is the same place as P and therefore infer that P was empty at T.
- The *path specialist* may infer that an object must have traveled through P at time T and that T was therefore occupied at that time.

In this simple example, eight of the nine specialists in S6 could quite possibly confirm or deny that P is empty at time T. I have been repeatedly surprised in my research how often so many specialists could be relevant to so many of the most mundane questions.

2. *Specialists should check for the consensus of the other specialists before acting on a belief.* A system suffers two penalties when one of its specialists uses a fact to make several inferences and later has to retract the original fact:

- *Loss of time and opportunity.* If a specialist bases some inferences on a fact that it later finds to be incorrect, it will have to spend time later revising all the facts it inferred from the original incorrect fact. The other specialists will have to do the same for each fact that is revised because of the original mistake. This will take time that could have been spent seizing an opportunity that is no longer available.

- *Painful or costly mistakes.* If a specialist takes an irreversible action based on a fact that is incorrect, the action could lead to unforeseen harm or cost to the system.

For example, if your place specialist does not remember anything being on a staircase you are descending and does not consult your perception specialist, then you risk tripping over something that might recently have been put on the staircase and injuring yourself.

Because most any specialist can be relevant to a particular fact (as I explained in the previous point) and acting on a mistaken belief can have high costs, it is in a system's best interest to check for the consensus of its specialists on a belief before acting on it.

3. *Specialists should have as much information as they can about a situation before acting on it.* Even a system whose specialists make no errors in modeling the world can improve its performance, often crucially, with more information.

For example, suppose your motor and visual specialists coordinate to pick up a cup in your kitchen. If you do not also consult your place specialist to remember that there is hot coffee inside the cup, you may burn your hand trying to pick it up.

As in the last point, a specialist risks causing a system to be harmed or to miss an opportunity if it does not consult with its other specialists to see what information they have to add to a situation.

4. *Specialists must detect unforeseen opportunities offered by other specialists.* When a specialist infers a fact about the world, other specialists may find that fact relevant. For example, if your path specialist is planning a detour around a traffic accident, you may notice that one of the possible detours passes a grocery store. If your spouse asked you to pick up some milk, then that detour, even if it is not the quickest route home, may be the best path to take. Unless your social specialist is

not kept abreast of what your path specialist is inferring and considering, then it will miss that opportunity to achieve a goal.

Taken together, these four principals suggest the following focus discipline:

All specialists should focus on the same proposition at the same time.

By following this discipline, the principles just mentioned are more likely to be followed. If all specialists are concentrating on the same proposition at the same then they will all have some sense of the consensus on that proposition, they will have information from all the other specialists that might help them act on the proposition and they will be able to detect opportunities presented by that proposition. If the specialists do not so focus, they will make wasteful or harmful inferences based on incomplete or inaccurate information and they will miss opportunities created by each other's inferences.

4.2 How specialists focus in S6

In Polyscheme and S6, I have implemented this focus discipline by forcing each specialist to *focus* on the same *proposition* at the same time. In this section, I describe how foci and propositions are implemented.

4.2.1 Propositions

Because all the specialists must concentrate on the same focus of attention and understand each other's opinion on it, Polyscheme has a communications protocol that all the specialists can understand and produce. The focus of attention is represented using that protocol. Such a protocol needs to be general and granular enough to express what each specialists represents internally in its own specialized representation. In Polyscheme, I have chosen a fairly ordinary "propositional" language to do this.

Decades of experience in logic, philosophy, linguistics and artificial intelligence have shown propositions of predicate calculus to be very granular and highly expressive⁴.

In Polyscheme a proposition has four parts: a predicate, zero or more arguments to the predicate and a temporal interval and “world” over which the predicate holds on those arguments.

4.2.1.1 *Entities*

All predicates, objects, events, temporal intervals, worlds, etc. are represented by *entities*. Entities have no internal structure. In S6, I have implemented entities as integers. S6 includes a dictionary that maps entities to names of entities (which are strings) and in the rest of this thesis, I will refer to entities by their string. Predicates, arguments and temporal intervals are familiar elements of artificial intelligence and the thesis has made no innovations in this regard. This thesis does introduce the notion of a *world* that is similar to existing notions in artificial intelligence research, but unique enough to deserve some explanation.

4.2.1.2 *Worlds*

The following three considerations led me to make “worlds” a feature of Polyscheme:

- *Hypotheticals and counterfactuals.* When you try to choose between which of the possible outcomes of an event to infer, or which action to take, or which event to posit in order to explain something, it is often helpful to imagine the hypothetical state of affairs in which each hypothetical choice is realized. Many algorithms in AI have used a “state” mechanism for keeping track of the state of the world that results from each possible choice. It is important to think of each choice separately because

⁴ At first glance, propositions in first-order predicate calculus are not very expressive because, e.g., they cannot express certain “second-order” propositions *about* predicates and propositions. This is not an issue here because higher-order items can and are reified so that they can be referred to as if they were first-order objects.

thinking of them together will lead to impossible outcomes. For example, if you have to decide between pressing on the gas and pressing on the brake when you see a yellow light at an intersection, and you imagine that you pressed on the gas and on the brake simultaneously, then the outcome that you imagine is very different than if you had imagined either separately.

- *Probabilistic and modal inference.* When you are trying to estimate the probability of any particular proposition being true, one class of strategies imagines several states of the worlds, throws out those that are not consistent with perception and finds the probability of the proposition being true by seeing how often it is true in the remaining states. This widely used approach to probabilistic inference is called “stochastic simulation”. Viewed this way, modal logics that formalize notions of possibility and necessity by counting how many possible worlds a proposition is true in are an (extreme) kind of stochastic simulation.
- *Other minds.* When you try to predict or explain the behavior of another person, you need to consider that his beliefs about the state of the world may be different from yours. One strategy is to imagine that you are in the world as he believes it to be, imagine that you had his goals and then imagine what you would do. Some researchers in “situation semantics” (Barwise and Perry, 1983), for example, have reified the points of view of each member of a conversation in order to formalize how the different beliefs of each participant shape the dialog.

Each of these points suggests that an important feature of cognition is to be able to imagine and think about a world that is different from the real world. Polyscheme achieves this with the **world** mechanism.

Worlds are entities. The **real world**, R , is the world that Polyscheme thinks is the actual world. All other worlds are formed by a **basis** of propositions which are assumed to be true or false in that world. The world $R + \text{Red}(\text{sky}, E)$ is the world in which the sky is always red (E is the temporal interval corresponding to eternity). A world is **more** or **less real** than another world if its basis is a subset or superset of that world. Unless a

specialist explicitly asserts otherwise, a proposition is true or false if the same proposition is true or false in a more real world. Thus, if `President(USA, Bush, May-2001, R)` is true, then so is `President(USA, Bush, May-2001, R+Red(sky,E))`, unless a specialist explicitly states otherwise.

4.2.1.3 *Example propositions*

Here are some examples of propositions. Note that they can express a lot – attributes, change events, spatial relations, etc. – but that it is cumbersome to do so. This is why Polyscheme and S6 use propositions almost exclusively as a communication protocol between specialists and almost never as a medium for representing knowledge and making inferences.

“Ball17 is red during time t1.”

```
color( ball17, red, t1, R )
```

“Ball 17 is behind screen12 during time t1.”

```
at( ball17, pBall, t1, R )
at( screen12, pScreen, t1, R )
behind( pBall, pScreen, t1, R )
```

“Ball17 went from being behind the screen12 to being to the right of the screen during time t.”

```
delta-event(e,E,R)
attribute(e, AT, E, R)
from(e, p1, E, R)
to(e, p2, E, R)
occurred(e,t,E,R)
at( screen12, pScreen, t, R )
behind(p1, pScreen, t, R)
right-of(p2, pScreen, t, R)
```

“Balloon3 would be touching the ceiling if it were filled with helium.”

```
touch(balloon3, ceiling, t, R+filled(ballon3, helium, t, R))
```

4.2.2 Focus

All specialists in Polyscheme concentrate on the same proposition at the same time. In particular, they concentrate on the same **focus**. A focus is a proposition together with a **focus mode**.

Foci need focus modes because specialists need to focus on propositions for several reasons:

- A specialist may be announcing a new inference to the rest of the specialists.
- A specialist may want confirmation of its belief about a proposition.
- A specialist may want to ask other specialists whether a proposition is true or not.
- A specialist would like all the other specialists to assume a proposition.
- A specialist would like all the other specialists to elaborate on a.

Each focus thus contains one of the following focus modes:

- PERCEIVE
- ASSERT
- DENY
- WONDER
- FIND
- ELABORATE
- IMAGINE

4.2.3 Focus in Polyscheme is more general than visual attention

An important point implicit in much of this chapter is that Polyscheme can focus on more than visual features that are immediately (in time and space) before it.

Polyscheme can focus on (i.e., imagine) scenes at other times and other places and even on scenes that do not exist in reality. While S6 only focuses on spatial and physical propositions, there is no reason why systems cannot be built from Polyscheme that focus on propositions about nonphysical objects and relations.

4.3 How the specialists relate to the focus

At every time step, all of the specialists in Polyscheme concentrate on a focus. Each specialist takes three actions on the focus:

1. It offers a **stance** on the proposition of the focus indicating whether the specialist believes the proposition is true or false and how curious the specialist is about the proposition's truth or falsity.
2. The specialists' stances on the focal proposition are all combined into a **consensus** that is reported to the specialists.
3. The specialists ask for information that will help them take a more accurate stance. The mechanism for requesting this information is described in the next chapter.

In the next chapter, I will also describe how Polyscheme chooses its focus.

4.4 Psychological evidence for integrative focus of attention

A large amount of psychological evidence suggests that humans have a focus of attention that combines information from various sources. Much of this is described in Baars' (1988) book on consciousness. In this section, I will describe some of the most compelling and suggestive examples. Unfortunately, the nature of psychological experiments has made it much easier to collect information about a focus of attention on visible parts of the world and there has been comparatively little research on a more general focus of attention.

4.4.1 Feature integration theory

Treisman and Gelade (1980) showed evidence that one function of visual attention is to integrate different kinds of features into a representation of an object and that this integration does not happen without attention. When shown scenes of variously colored and shaped objects, people find it easy to search for objects with a particular feature, e.g., green objects. The time it takes them to pick out a target is roughly constant regardless of the number of objects in the scene. However, when asked to search for objects with a combination of features (e.g., green triangles) subjects take a much longer time that grows linearly with the number of objects in the scene.

This result suggests that searching for a combination of features requires people to focus on each particular object and judge if it combines those features. In other words, subjects cannot (easily or quickly, at least) integrate more than one feature into the representation of an object unless they focus on the object.

4.4.2 Stroop effect.

Stroop (1935) found that people have difficulty focusing on a feature in one modality and a contradictory feature in another modality. For example, the time it takes people to read the word "green" is much longer if the word was written in red rather than green ink.

This suggests that when people focus on a feature, they try to bring a lot of other processing to bear on that features. Thus, when people focus their reading specialist on the word "green", their color specialist(s) also focuses on whatever the reading specialist is focussed on.

4.4.3 Human and rat navigation

Hermer and Spelke (1994) found that older children and adult humans could use both spatial and nonspatial information to locate objects while rats and infants could not. In

their experiments, they would hide an object in a room, disorient subjects and then see if the subjects could find the object. Hermer and Spelke varied the landmarks available to subjects to see what kind of information they could use to find objects. They found that older children and adults could use the unique colors or unique shape of a part of a room to remember where to find a hidden object, while young children and infants could use only the shape information.

One (admittedly quite speculative and vague) inference to draw from this experiment is that the ability to combine different kinds of information is an important part of more advanced forms of cognition. Hermer and Spelke's experiments do not study a focus of attention, but they do suggest that the integrative function of the focus of attention in Polyscheme is an important function in cognition generally.

Chapter 5

Managing Focus

Because specialists in Polyscheme concentrate on a single focus of attention, it needs mechanisms for choosing what focus to concentrate on. Without these, a system risks wasting time, missing opportunities and not detecting dangers. This chapter describes Polyscheme's infrastructure for managing attention and outlines how I implemented those in S6.

The most important insight of this chapter and of this thesis is that many (if not most) important inference schemes are implemented as *schemes of guiding the focus of a system of specialists*.

5.1 Guiding focus

There are several reasons Polyscheme needs mechanisms to choose and order foci:

- Some foci are (very likely to be) completely irrelevant (to a system's goals) and some are more relevant than others. A system maximizes its chances of success and survival when it focuses on the most relevant foci. The structure of the regularities and laws that govern the world make some foci more relevant than others. For example, most ordinary physical events are caused by other events that are near them in space and time. This is particularly true of common physical interactions and suggests that in order to explain a physical event, you should focus (in part) on the immediate spatial and temporal location of the event to find its cause and its effects.

- Specialists need information from other specialists to make inferences and take actions (that may be urgent). A social specialist, for example, may need a spatial specialist to determine if someone in a conversation is facing towards or away from a speaker. Because conversations are very sensitive to time (e.g., it is not optimal to spend one hour formulating your next response in a spoken conversation), there need to be a mechanisms that insure that the social specialist gets the information it needs when it needs it.
- Some specialists may have time-sensitive machinery. For example, a physics specialist may have a short-term memory for the location of the last four objects the system focussed on or a sentence-understanding specialist might have a short-term memory for that last ten syllables. Without a mechanism to make sure these specialists focus on the correct information within an adequate interval of time, the system as a whole would not work as well as it could, if at all.
- Specialists' requests for foci might conflict. Two specialists, for example, may want to focus on two different propositions at the same time and therefore need mechanisms to decide which proposition to focus on.
- New, changed or contradictory information and inferences may cause specialists to improve their actions if they are received in time and may cause them to miss opportunities, waste time making inference that need to be revised or take harmful actions they would have avoided with the improved information.

The next section describes Polyscheme's mechanisms for guiding attention.

5.2 Polyscheme's mechanisms for guiding focus

Polyscheme manages the focus of attention with two mechanisms:

1. Specialists use **attractions** to indicate what they think the system should focus on.

2. A **reflective specialist** detects conflicts, uncertainties and other relationships between specialists and issues attractions for foci that address them.

5.2.1 Attractions

Specialists use attractions to request that Polyscheme guide focus to a particular (kind of) focus. Attractions have 4 components:

1. The *focus* that the proposition would like the system to concentrate on.
2. The *parent attraction* that a focus helps satisfy. For example, focussing on a region in space to see if it is empty can satisfy a parent attraction aimed at determining if an object will fall through that region of space.
3. An *inherent charge* that indicates how important (the specialist thinks) it is for the system to concentrate on the focus. The inherent charge indicates how important the focus is independent of whether the parent attraction (if there is one) is important. Many, if not most, attractions, for example, are only important insofar as they help satisfy their parent attraction. In this case, their inherent charge is zero.
4. A *total charge* that is a function of the parent's charge and the inherent charge. This function can include a great deal. Here are some examples:
 - Attractions can decay their charge so that Polyscheme is more likely to focus on something at a certain time (e.g., before it is likely to disappear) and less important later on.
 - If a specialist is very confident that an attraction will satisfy the parent attraction, then the total charge will be a larger function of the parent charge than if the specialist is not so confident.

- An attraction may only be important insofar as a certain state of the world has not been achieved. For example, an attraction to find food may only be important so long as an animal is hungry. The total charge may therefore be a function of whether the animal is hungry.

One way to think of attractions is as “generalized goals”. Many artificial intelligence systems have goals that are (often partially-specified) states of the world and these systems are designed to achieve those goals. Attractions are generalized goals in that they can do more than just indicate that a proposition is desired to be true. They can indicate that Polyscheme is *curious* about a proposition or that it should *elaborate* the consequences of a proposition. In other words, goals are just one focus mode – ACHIEVE – that attractions can have. There are also many similarities. Just as goals have subgoals, attractions have children. Just as the importance and/or urgency of a goal depends on that of its supergoal, so does the charge of an attraction depend on its parent’s charge.

5.2.2 Reflective specialist

The reflective specialist detects conflicts, uncertainties and changes in the consensus of the specialist and suggests foci that will address these.

When Polyscheme focuses on a proposition, specialists can have different kinds of consensi regarding the proposition. Each kind of consensus will require different kinds of foci. The reflective specialist issues attractions for those foci. For example:

- *Full consensus*. All the specialists that offer a stance on a proposition agree that a proposition is true or that it is false.
- *Curious Ignorance*. No specialist knows whether a proposition is true or false, but at least some of the specialists are curious about that.

- *Changed truth value.* The specialists come to a consensus that a proposition is different from their previous consensus on the proposition.
- *Conflict.* Some subset of the specialists think a proposition is true and others think it is false.

I will call this kind of summary description of the specialists' consensus a **metaconsensus**. The job of the reflective specialist is to detect a metaconsensus and issue attractions that will cause the system to focus on propositions that will improve the metaconsensus. For example, if there is a conflict among the specialist, the reflective specialist will issue attractions for foci that might help resolve that conflict. The next section, 5.3, on "focus schemes" will describe the how the reflective specialist works in greater detail.

Once all the specialists, including the Reflective Specialist, have issued their attractions, Polyscheme picks the most charged attraction at that time step and focuses all the specialists on the proposition in that focus.

5.3 Focus schemes realize inference schemes

The premise of this thesis is that artificial intelligence research needs an architecture for integrating various kinds of representation and inference schemes into one intelligent system. The third and fourth chapters have partly explained how Polyscheme enables this. Polyscheme has several specialists, each with their own internal method for representing knowledge and making inferences. These specialists share information through a focus of attention using a proposition language. But this is less than half of the solution. A key insight in this research and in this thesis is that most inference patterns are not implemented within specialized processors, but as *schemes of focussing specialists*. The rest of this chapter explains this idea and the next chapter illustrates it in the context of designing S6 to reason about the physical world.

5.3.1 Some inference schemes require several specialists

Some inference schemes require only one representation while others require many. “Counterfactual reasoning”, for example, requires several representations. If you are uncertain about whether climbing on a table to reach a light bulb is wise, you might imagine the world in which you are on the table in order to see if you would be close enough to reach the bulb and to predict any harmful consequences. When you imagine the world in which you are on a table, *you need to use all the representations and (therefore) all the specialists that are relevant* to inferring what that world would be like. For example, your social specialist might conclude that being on the table might offend the table’s owner; a physical-object specialists might conclude that you are too heavy for the table; a spatial specialist might conclude that you will not reach the bulb; etc. In other words, counterfactual reasoning is not an inference scheme that uses one representation and that should be inside one specialist. Instead, it is a *method of focusing all specialists towards useful propositions*. It is a *multiple-specialist inference scheme*. Counterfactual reasoning is a scheme for guiding focus. It is a focus scheme.

Contrast this with inference schemes that do not need to consult outside specialists. For instance, neural network propagation uses a function to map a vector of inputs to a vector of outputs. The function corresponds to weighted connections of the input vector to the output vector, possibly thorough intermediate “hidden layers”. The function from the input to the output does not need to consult or output information to or from the rest of the system. It can therefore stay encapsulated inside a specialist. Rule-matchers are similar. A rule matcher finds rules that can match against certain preconditions and outputs assertions based the preconditions and the rules. The matching process itself does not need input from other specialists. These are all *single-specialist inference schemes*.

The major insight of the research this thesis reports is that many (if not most) important inference schemes are multiple-focus inference schemes and should be implemented as schemes of guiding an integrative focus of attention. Let us now see how Polyscheme and S6 implement several kinds of inference schemes as foci of attention.

5.3.2 Some inference schemes in S6

In this subsection I will present some inference schemes that S6 uses and show how they are implemented using the attraction mechanism.

5.3.2.1 *If you are curious whether P is true or false, try to perceive P*

This is perhaps the most straightforward focus scheme. If you are curious about a proposition P, try to perceive it. S6's reflective specialist implements this by issuing the following attraction:

```
(attraction
  :proposition P
  :focus-mode PERCEIVE
  :inherent-charge ...
  :total-charge ... )
```

5.3.2.2 *If you are curious about the attribute of an object in the near past or near future, try to perceive the current value of the attribute.*

Most attributes of objects either persist through time or change with some regularity. Thus, a good way to determine the value of an attribute in the near future or near past is to perceive it in the present. S6's attribute specialist implements this by issuing the following attraction, where ATTR and object are the attribute and object in question:

```
(attraction
  :proposition ATTR(object, ?v, now, R)
  :focus-mode PERCEIVE
  :inherent-charge ...
  :total-charge ... )
```

Once this attraction is chosen and the system focuses on the current value of the object, then the attribute specialist can project the value forward or backward. Notice that even though this is an attraction for perceiving the proposition, another specialist can infer the current value. In other words, the attribute specialist can *recognize and exploit opportunities* provided by other specialists because they all focus on the same proposition at the same time.

5.3.2.3 *Resolve conflict with perception.*

If two (or more) specialists disagree about the truth value of a proposition, then you should try to perceive its truth value. When the uncertainty specialist detects a conflict metaconsensus, it inserts the following attraction:

```
(attraction
  :proposition P
  :focus-mode PERCEIVE
  :inherent-charge ...
  :total-charge ... )
```

5.3.2.4 *Resolve conflicts by imaging counterfactuals.*

If two (or more) specialists disagree about the truth of a proposition, P, then you can imagine that P is true and see what follows. If that results in a contradiction, then you know that P is false. You can also imagine that P is false and see if that leads to contradictions. When S6 reaches a conflict metaconsensus on a proposition, R(x, y, t, w), it issues the following attractions:

“Imagine that R(x, y, t) is true in world w.”

```
(attraction
  :proposition R( x, y, t, w + P-true)
  :focus-mode IMAGINE
  :inherent-charge ...
  :total-charge ... )
```

“Elaborate on R(x, y, t) in imagined world.”

```
(attraction
  :proposition R( x, y, t, w + P-true)
  :focus-mode ELABORATE
  :inherent-charge ...
  :total-charge ... )
```

“Imagine that $R(x, y, t)$ is false in world w .”

```
(attraction
  :proposition R( x, y, t, w + P-false)
  :focus-mode IMAGINE
  :inherent-charge ...
  :total-charge ... )
```

“Imagine that $R(x, y, t)$ is true in world w .”

```
(attraction
  :proposition R( x, y, t, w + P-false)
  :focus-mode ELABORATE
  :inherent-charge ...
  :total-charge ... )
```

The inherent and total charge functions in these attractions need to depend on time. If you imagine that P is true first, then the charge on those attractions need to decay at some point so that you can choose the attractions for imagining that P is false.

5.3.2.5 *Backtracking search*

When you imagine a world based on one counterfactual action, you may need to imagine another counterfactual action to see what the result of the first would be. This kind of “nested” counterfactual reasoning amounts to backtracking search. In other words, backtracking search is implemented in S6 by a focus scheme that issues attractions to focus on counterfactual worlds based on actual or counterfactual worlds. You can emulate various kinds of backtracking search by setting the charges of the counterfactual propositions appropriately. In depth-first search, for example, the attraction for imagining one action has an enduring and high charge that keeps a competing action from being imagined for a long time. In breadth-first search, the

attraction for imagining an action decays quickly and the other action is therefore considered quickly.

5.3.2.6 *Stochastic Simulation*

In a world in which events have more than one outcome, it can be difficult to determine what the most likely outcome of a series of uncertain events may be. One strategy for finding the most likely outcome is to imagine events that can occur in the frequency that they are likely to occur. After you discount imagined worlds that are not consistent with what you have perceived of reality, you determine the likelihood of a particular proposition being true by taking the ratio of the number of worlds in which it is true to the total number of worlds.

Though I have not implemented stochastic simulation in S6 (mostly because S6 is in its current implementation too slow to perform hundreds of simulations), here is how I would go about doing it. Each time specialists predict more than one possible outcome to an event, the reflective specialist would issue attractions for imagining each outcome. The charges of the attractions would be such as to lead the number of times each outcome was simulated to be roughly in parity with its relative likelihood of occurring. After some of this simulation, a "frequency" specialist would keep track of which resulting worlds led to contradictions and which were not. To determine the probability of a proposition, P , the frequency specialist would take the ratio of the number of possible worlds in which P was true to the number of total possible worlds.

There are many details to be worked out here, but this outline does show that it is possible to implement stochastic simulation in Polyscheme.

5.3.2.7 *Belief revision.*

For reasons mentioned in chapters four and five, when a specialist changes its view of a proposition, it should try to get the rest of the system to focus on the proposition quickly so that other specialists do not miss opportunities or make harmful inferences based on

the incorrect belief. When a specialist in S6 changes its view on a proposition, it inserts the following attraction, where *very-high-charge* is a charge that is high enough to have S6 focus on P quickly:

```
(attraction
  :proposition P
  :focus-mode ASSERT
  :inherent-charge very-high-charge
  :total-charge very-high-charge)
```

Notice that this focus scheme leads to “recursive belief revision” in that when a belief changes, not only are beliefs based on that changed belief revised, but all changed beliefs that lead from the revised beliefs themselves.

5.3.2.8 *Confirm identity by fusing propositions*

Two entities are identical only if the same propositions are true of both. One way to be more confident about an identity hypothesis between two entities is to make sure that the propositions that are true or false on one are also true or false on the other. The more propositions you do this for, the more likely the identity hypothesis is to be true. If you find a proposition true on one entity and false on another, then you have strong evidence that the identity hypothesis is flawed. When one of S6’s specialists asserts that entities *x* and *y* are the same entity, the identity specialist finds propositions that hold on *x* and asserts that they hold on *y* and *visa versa*. In particular, when the identity specialist focuses on $ID(x, y, E, w)$ it issues the following two attractions:

```
(attraction
  :proposition R( ... x ..., ?t, w )
  :focus-mode ASSERT
  :inherent-charge temporary-charge
  :total-charge temporary-charge)
```

```
(attraction
  :proposition R( ... y ..., ?t, w )
  :focus-mode ASSERT
```



```
:inherent-charge temporary-charge
:total-charge temporary-charge)
```

The charge, *temporary-charge*, is a charge that lasts for a fixed period of time that the identity hypothesis sets. The more important it is to confirm the identity hypothesis, the greater the strength of the charge and the period of time it lasts.

5.3.2.9 *If a specialist asserts that a proposition is true during a temporal interval, make sure that the same proposition is not false over an overlapping temporal interval.*

If one or more specialists assert that P holds over T, then P must be true for all intervals included in T and P cannot be false over intervals that include T. When a specialist asserts that a proposition $R(x, y, T, w)$, S6's temporal specialist issues an attraction to focus on propositions at overlapping times, ?t:

```
(attraction
:proposition R( x, y, ?t, w )
:focus-mode WONDER
:inherent-charge ...
:total-charge ... )
```

5.3.2.10 *Script matching*

Scripts are sequences of events that tend to or ought to occur in certain circumstances. They help a reasoning system explain an event or achieve a goal by giving it a place to start among the almost countless possible sequences of actions. The path specialist in S6, for example, uses the "left-adjacency script" to help it find a continuous path between two glimpses of an object. The left-adjacency script represents a decomposition of a path from P1 to P2 into the path from P1 to P and then from P to P2, where P is adjacent to P1. The script has a slot for each place and a slot for the two changes of position. When part of a script matches, the path specialist issues attractions for the rest of the script. For example, when the path specialist sees that object, O, is at (0,0,0) at time T1 and then at (0,4,0) at T2, it issues the following attractions which posit a point P adjacent to (0,0,0) and a movement from (0,0,0) to P and from P to (0,0,0).

```
(attraction
  :proposition EXISTS( P, E, R )
  :focus-mode ASSERT
  :inherent-charge ...
  :total-charge ... )
```

```
(attraction
  :proposition ADJ(0,0,0), P, E, R )
  :focus-mode ASSERT
  :inherent-charge ...
  :total-charge ... )
```

```
(attraction
  :proposition DELTA( AT, O, (0,0,0), P, T-delta1, R )
  :focus-mode ASSERT
  :inherent-charge ...
  :total-charge ... )
```

```
(attraction
  :proposition DELTA( AT, O, P, (0,4,0), T-delta2, R )
  :focus-mode ASSERT
  :inherent-charge ...
  :total-charge ... )
```

It is up to the other specialists (especially the space specialist) to decide what the identity of P is. The benefit of implementing script matching (and hence path search) as a focus scheme is that you can recruit all the specialists in S6 to help bind a slot in the script.

5.4 Polyscheme integrates the important frameworks of artificial intelligence research

This chapter concludes the description of Polyscheme. I will briefly review how Polyscheme achieves the goal of providing an architecture for integrating several inference schemes into one system.

There are two main methods to implement an inference scheme in Polyscheme. You can encapsulate it in a specialist or you can implement it as a focus scheme. The reason to implement many inference schemes as focus schemes is that they require multiple representations and therefore multiple specialists.

Inference schemes in Polyscheme are “flexible” because of two mechanisms. Specialists (and the inference schemes they implement) can all share information through the focus of attention because they can input and output information using propositions.

Inference schemes that are implemented as focus schemes can operate at nearly the same time because two (or more) sets of attractions (one for each focus scheme) can be chosen over the same period of time.

In the first chapter, I explained why I decided to design an architecture that integrates components based on the various frameworks for building intelligent systems instead of designing a new framework that subsumed them all. In this chapter and the last two, I have shown how to incorporate the following major AI frameworks inside systems based on Polyscheme: rule-based systems, neural networks, scripts, backtracking search, probabilistic reasoning and truth maintenance. In the next chapter, I demonstrate that integrating so many different kinds of computational schemes into one system enables powerful common sense reasoning.

5.5 Appendix. Reasoning about focus

None of the specialists I have built into S6 explicitly reason about what propositions to focus on. Instead, they are designed to recognize certain kinds of metaconsensi (or metacognitive states) and issue the appropriate attractions. Currently, the appropriate attraction to issue is programmed into the specialists and this has been enough to build a lot of common sense. However, there is nothing to prevent someone from improving on this and building specialists that explicitly reason about what foci are most appropriate.

This section offers some examples and suggestions about why and how you could explicitly reason about focus in Polyscheme.

5.5.1 Knowing that you do not know

Do you know George Bush's phone number? Do you know if the window behind you just shattered? The answer to the first question is probably, "No", and the answer to the second is probably, "Yes". In both cases, you try to retrieve either a George-Bush-phone-number memory or a window-shattering memory and fail. Part of the difference here is that (assuming you are not deaf or extremely absent-minded) you would have heard and remembered a window breaking behind you very recently. The fact that you cannot retrieve a memory for such a shattering event is thus strong evidence that one did not occur.

One way to implement this kind of inference in Polyscheme is to create a "metamemory" specialist that has a model of your memory and can reason about memory retrieval successes or failures just as other specialists would about the success or failure of objects moving about the world. It is not hard to believe that we spend many of the first years of our lives building such a model of our own memory as we succeed and fail at finding things we need.

5.5.2 Knowing you can

When you promise someone to do something in the future, you need to remember to do it. For example, if I promise John to call him in "a couple minutes", I am fairly confident that it will "occur" to me to call him in a few minutes. If I promise to call John an hour before our appointment next month, I will probably forget to do that and will need to set an alarm in my calendar to remind me.

This kind of reasoning would be fairly simple in Polyscheme with the metamemory specialist I just described in the last subsection. Using its model of Polyscheme's memory, it could infer that such-and-such action (e.g., writing something down where

you see it every day) would remind you to achieve a goal at a particular time. It would imagine that if the meeting is in a few minutes, you will not forget, but that if it is in several weeks, you will forget.

5.5.3 Dangerous thoughts

Some foci might always lead to states where none of your goals become satisfied or might cause you to become so confused that you cannot think well. If you modify the reflection specialist so that it can detect dangerous states and give it a fairly simple sequence learning mechanism, it can avoid dangerous thoughts. Just like we avoid putting our hands in fire because we have learned that pain follows, the reflective specialist can learn that some thoughts or foci lead to harmful or wasteful states and issue attractions to avoid those foci.

Chapter 6

Polyscheme and Physical Reasoning

As I introduced each new feature of Polyscheme in the last chapter, I explained how I implemented it in S6. In this chapter, I show how S6 works in a number of situations, demonstrating that the principles of Polyscheme enable common sense reasoning. I give a detailed example of S6 reasoning about a particular situation and then more briefly describe how S6 reasons in a wide range of physical scenarios.

6.1 The program

S6 was implemented with the Java programming language and developed using the Java 1.2 SDK. The source code, including comments and white space, has about 12,000 lines that measure roughly 400KB. Most of the examples I show here required, within an order of magnitude, about a minute for S6 to reason about on a PC with two Intel Pentium III 800MHz processors and 512 megabytes of memory.

6.2 The bridge example

The first example of the first chapter showed that effectively combining several inference schemes into one system requires a new computational architecture. In this section, I will describe in detail how S6 reasons about the same example mapped into the world S6 was designed to reason about.

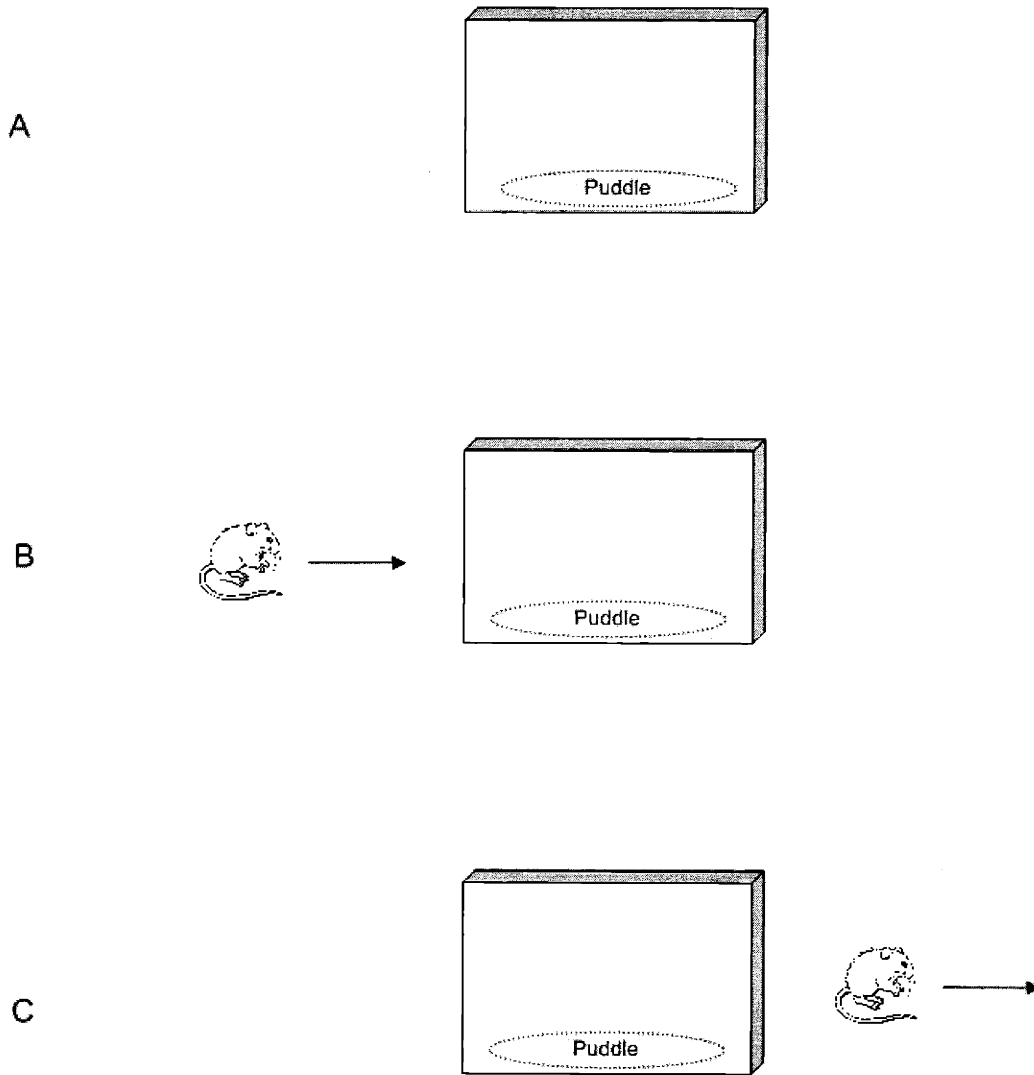


Figure 7. S6 sees a block with a puddle behind it (A); a dry mouse walks behind the block (B) and a dry mouse walks out from behind the block (C). S6 must decide whether the mouse in C can be the same as the mouse in B.

Figure 7 shows the scenario I will describe. Figure 7a shows a screen occluding a region known to contain a puddle of water. In Figure 7b, a dry mouse walks behind the screen from the left and in figure 7c, a dry mouse walks out of the screen to the right. S6 infers that the mouse walking behind the screen from the left is not the same as the mouse coming out to the right because neither mouse is wet. If they were the same mouse, the mouse coming out from the right would be wet because it must have traveled through the puddle behind the screen to be the same mouse that went behind the screen.

This scenario is essentially the same as the bridge example in the first chapter:

- You see an object (person or mouse) and then later see an object that looks similar to it.
- A puddle of water separates each sighting of the object.
- You hypothesize that both objects are the same because they share many properties.
- You infer that they are not the same objects because, if they were, then the second object sighting would be of an object that just crossed a body of water and that it would therefore be wet instead of dry.

Here are the key steps S6 takes to infer that the two mice are not the same. Note that while these were the steps that S6 took to infer that the two mice sightings were not of the same mouse, they were not the only things that S6 focused on. S6 focused on several propositions that had nothing to do with the ultimate inference, which is not surprising given that it did not know what the ultimate inference would be in advance.

1. The perception specialist perceives a dry white mouse behind the screen from the left. At regular intervals, the perception specialist issues attractions to report what it sees. The attractions have high charges and have the effect of forcing S6 to focus on these new perceptions almost immediately. In this case, the foci are

for the following propositions: $\text{mouse}(x, E, R)$, $\text{white}(x, t1, R)$, $\text{wet}(x, t1, R)$, $\text{at}(x, p1-1-1, t1, R)$ (the mouse is to the left of the screen at $t1$), $\text{at}(x, p1-2-1, t2, R)$, (the mouse is behind the screen at $t2$), and $\text{at}(\text{screen}, p1-2-0, t1, R)$. The perception specialist takes a true stance on all of these propositions except for $\text{wet}(x, t1, R)$, on which it takes a false stance because the mouse is dry.

2. For a few moments, the perception specialist perceives the screen and nothing else: $\text{at}(\text{screen}, p1-2-0, t2, R)$, $\text{at}(\text{screen}, p1-2-0, t3, R)$, $\text{at}(\text{screen}, p1-2-0, t4, R)$.
3. Then the perception specialist perceives a dry white mouse walk out from behind the screen to the right: $\text{mouse}(y, E, R)$, $\text{white}(y, t5, R)$, $\text{wet}(x, t5, R)$, $\text{at}(x, p1-2-1, t4, R)$ (the mouse is behind the screen at $t4$), $\text{at}(x, p1-3-1, t5, R)$ (the mouse is to the left of the screen at $t1$), and $\text{at}(\text{screen}, p1-2-0, t5, R)$. The perception specialist takes a true stance on all of these propositions except for $\text{wet}(x, t1, R)$, on which it takes a false stance because the mouse is dry.
4. The identity hypothesis specialist hypothesizes that the two mouse glimpses are of the same mouse because they share many of the same properties. It issues an attraction for the proposition that the two glimpses are of the same object: $\text{id}(x, y, E, R)$.
5. The attribute specialist infers a change-in-position event. When the attribute specialist focuses on a proposition that asserts the identity of two objects, it fuses their properties together and checks for changes or mismatches. In this case, the attribute specialist combines the properties of the two mice (x and y) and detects a change in position. It issues an attraction to focus on that change: $\text{exists}(e, E, R)$, $\text{delta}(e, AT, x, p1-1-1, p1-3-1, t\text{-delta}, E, R)$, $\text{after}(t\text{-delta}, t1, E, R)$, $\text{before}(t\text{-delta}, t5, E, R)$. This essentially says that

there is an event e that is a change of x 's position (AT) attribute's value from p_{1-1} to p_{1-3-1} and that this change occurred between t_1 and t_5 .

6. The path specialist matches the left-adjacency-script path planning script and asks the system to find the first point that x was at after p_{1-1-1} . The path specialist calls this point p , and issues an attraction indicating that it exists and that it is adjacent to p_{1-1-1} : $exists(p, E, R)$, $adjacent(p, p_{1-1-1}, E, R)$. The path specialist also issues attractions for propositions that indicate that x was at p at some point during the time of the change in position: $at(x, p, t-p, R)$, $during(t-p, t-delta, E, R)$.
7. The space specialist says that p can be one of 6 points adjacent to p_{1-1-1} and issues attractions with a focus mode of WONDER for propositions indicating the possible identity. For example, $id(p, p_{1-2-1}, E, R)$.
8. As possible places are suggested that might be identical to p , the identity specialist issues attractions for what that identity would imply for the location of x . For example, $at(x, p, t-p, R)$ and $id(p, p_{0-1-1}, E, R)$ imply that x was at p_{0-1-1} or $at(x, p_{0-1-1}, t-p, R)$.
9. When possible positions for x at time $t-p$ are focused on, the perception specialist rules out those positions that it saw were empty. For example, the perception specialist takes a stance that the proposition, $AT(x, p_{0-1-1}, t-p, R)$ is false because it never saw an object at p_{0-1-1} .
10. All of the possible places that p might be identical to are ruled out except p_{1-2-1} , the place just behind the screen. When the identity specialist has only one possible candidate for an identity, it issues an attraction for that identity proposition – in this case, $id(p, p_{1-2-1}, E, R)$. When S6 focuses on that proposition, the identity specialist takes a true stance on that proposition.

11. When S6 focuses on $id(p, p_{1-2-1}, E, R)$ the identity specialist infers from $at(x, p, t-p, R)$ that $at(x, p_{1-2-1}, E, R)$. In other words, it infers that the mouse is behind the screen at that time step.
12. When S6 focuses on the mouse being behind the screen, the causal specialist, which, along with all of S6, has already focused on the place behind the screen being wet, infers that the mouse will be wet from now on. It therefore issues an attraction for the propositions: $wet(x, t-wet, R)$ and $after(t-p, t-wet, E, R)$.
13. Whenever the identity specialist focuses on a new proposition that S6's specialist reach a true consensus on, it issues an attraction for propositions that might be true or false of the same object. It does this to make sure that a property being posited on an object is not inconsistent with that property being negated on another object that happens to be identical to the first object. In this case, when the identity specialist focuses on $wet(x, t-wet, R)$, it issues an attraction for $wet(?o, ?t, R)$.
14. When the attribute specialist focuses on $wet(?o, ?t, R)$, its memory for objects that are wet or not wet and finds that y , which is identical to x , was wet at time t_5 . The specialist therefore issues an attraction for $wet(y, t_5, R)$.
15. Just as the identity specialist checks for propositions on identical times that might contradict a proposition, so does the temporal specialist check for propositions on overlapping intervals that might contradict each other. In this case, when it focuses on $wet(y, t_5, R)$, it determines that t_5 (the time of the mouse coming out of the screen) is included in $t-wet$ (the time after the mouse went through the puddle and was wet). But the mouse was wet after it went out of the screen ($t-wet$) and dry when it came out of the screen (t_5). These both cannot be true because t_5 is included in $t-wet$. Because S6 "directly" perceived that y was dry at t_5 , then y cannot be dry at $t-wet$. The temporal specialist

therefore issues an attraction for $wet(y, t-wet, R)$ and takes on false stance when S6 focuses on that proposition.

16. When S6 focuses on $wet(y, t-wet, R)$ and the specialists take a negative stance, the identity specialist must deal with an internal contradiction. It assumed that y was identical to c , but that cannot be because it knows for certain that x was wet when it was behind the screen ($t-wet$). Thus x and y cannot be the same object. The identity specialist therefore issues an attraction for $id(x, y, E, R)$ and takes a false stance when S6 focuses on it.

There are several points to make about this series of inferences that illuminate how S6 enables several inference and knowledge representation schemes to be combined together into one system.

First, notice that this inference is basically a script matching inference. S6 thinks it sees mouse move from one place to another and matches a script to fill in the gaps in the motion. The script includes a "slot" for an intermediate point between the two mouse sightings. Much of the inference in this example is about trying to fill the slot, deciding that there is no possible filler and therefore retracting the identity hypothesis.

Notice all the inference that went into the simple task of filling the slot with a place the mouse might have been at:

- *Perception.* S6 rules out some places because the perception specialist remembers seeing nothing in those locations.
- *Identity.* S6 rules out the place behind the screen because that leads to an inference that the mouse on the right would be wet and it manifestly was not. This inference depends on knowing that the mouse on the right should have the same properties as the mouse behind the screen if they are the same mouse.

- *Temporal Constraints.* The inference also involves keeping track of temporal constraints so that you know that the time of the inferred wetness of the mouse includes the time where you saw the similar mouse being dry.

The achievement of Polyscheme is to create an architecture where seemingly elementary tasks such as filling a slot in a script can be informed by multiple knowledge representation and inference schemes.

6.3 More examples

In this section, I present some scenarios that S6 reasons about. Though I do not describe these cases in as much detail as in the previous section, I do describe them enough for a motivated reader to fill in the blanks. These examples are intended to give an idea of how good S6 is at combining several knowledge representation and inference schemes and how this ability enables S6 to be a very versatile and resourceful reasoner.

6.3.1 Gravity and imagination

In this scenario S6 begins by seeing a group of objects and then sees a block dropped above and behind the objects, as in Figure 8. S6 infers that the block will fall behind the objects and come to rest on the floor. To make this inference, S6 must predict that the ball falls behind the screen even though it cannot see that motion. It must also “imagine” the collision with the floor.

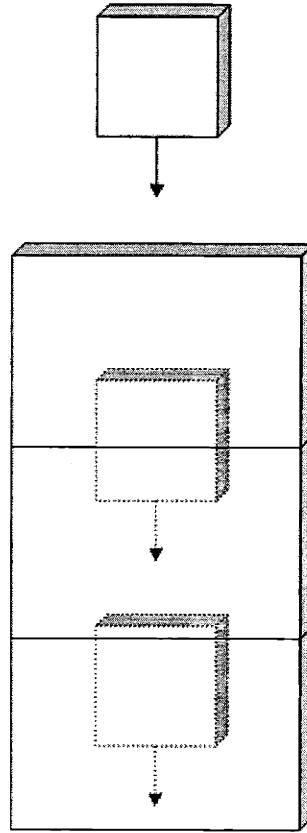


Figure 8. S6 sees the block fall behind the stack of blocks and infers, without being able to see the block, that the block continues to fall until it hits the ground.

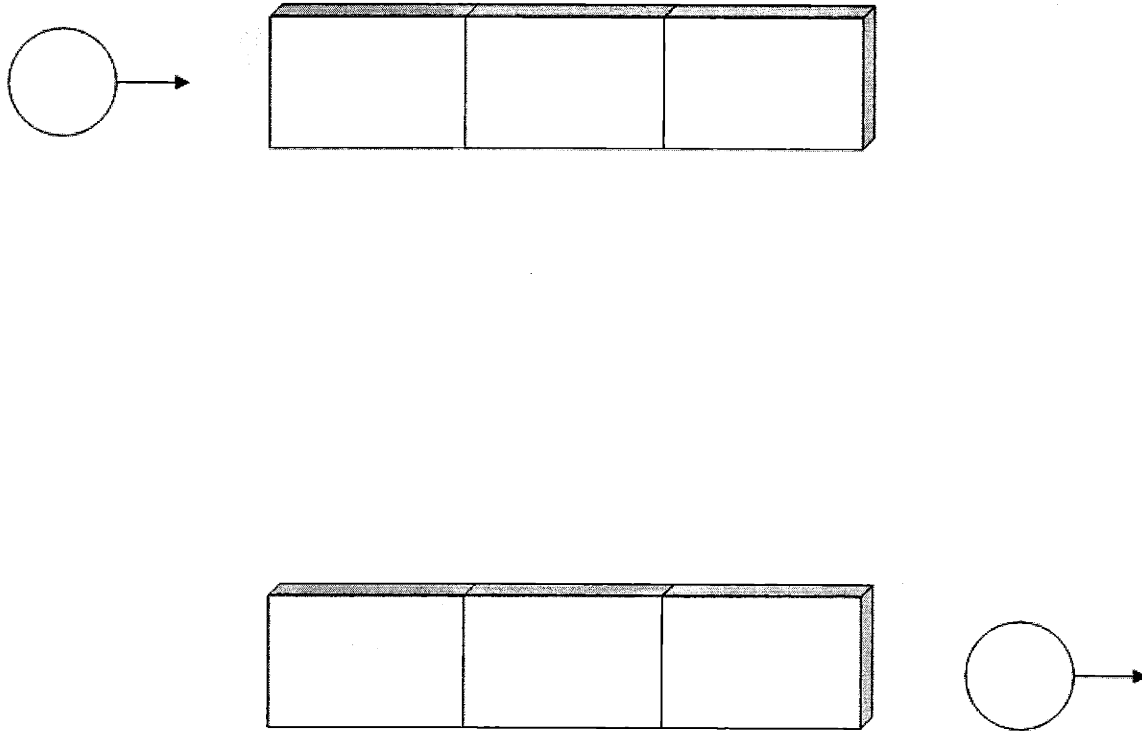


Figure 9. S6 sees a ball roll behind the blocks and a ball that looks the same roll out. S6 assumes that the second ball is the same as the first ball and posits a continuous path between the two.

6.3.2 Belief Revision

This scenario is just like the previous one, except that after S6 sees the block fall behind the screen, it is told that there is and was a block directly beneath the block behind the screen. S6 infers that instead of falling to the floor, the first block came to rest on the second block. S6 does this by issuing attractions to refocus on propositions involving the region behind the screen being empty. When S6 refocuses on those propositions, the gravity specialist now knows that the block *is* supported just as it falls behind the screen and predicts that it will come to rest there and not on the floor.

6.3.3 Object permanence and identity

In this example, illustrated in Figure 9, S6 sees a red ball go behind some occluding objects from the left and a red ball roll out from the right. S6's identity hypothesis specialist first suggests that the first ball is the same as the second ball. The path specialist then posits an intermediate point between the two object sightings and the only two points that are not ruled out by perception and memory are the points behind the screen.

6.3.4 Identity blocked by a failure to find a path

Spelke and Kassebaum (1987) used an example similar to this one to show that infants "know" that objects trace continuous paths. As illustrated in Figure 10, S6 begins by viewing a scene with two blocks separated by a space. A ball rolls behind the left block and in a few moments a ball emerges from the right block. S6 infers that the two balls are different because there is no continuous path in space and time that could have connected them. Note that, except for the visible empty space between the left and right blocks, this example is the same as the previous example where S6 did infer identity.

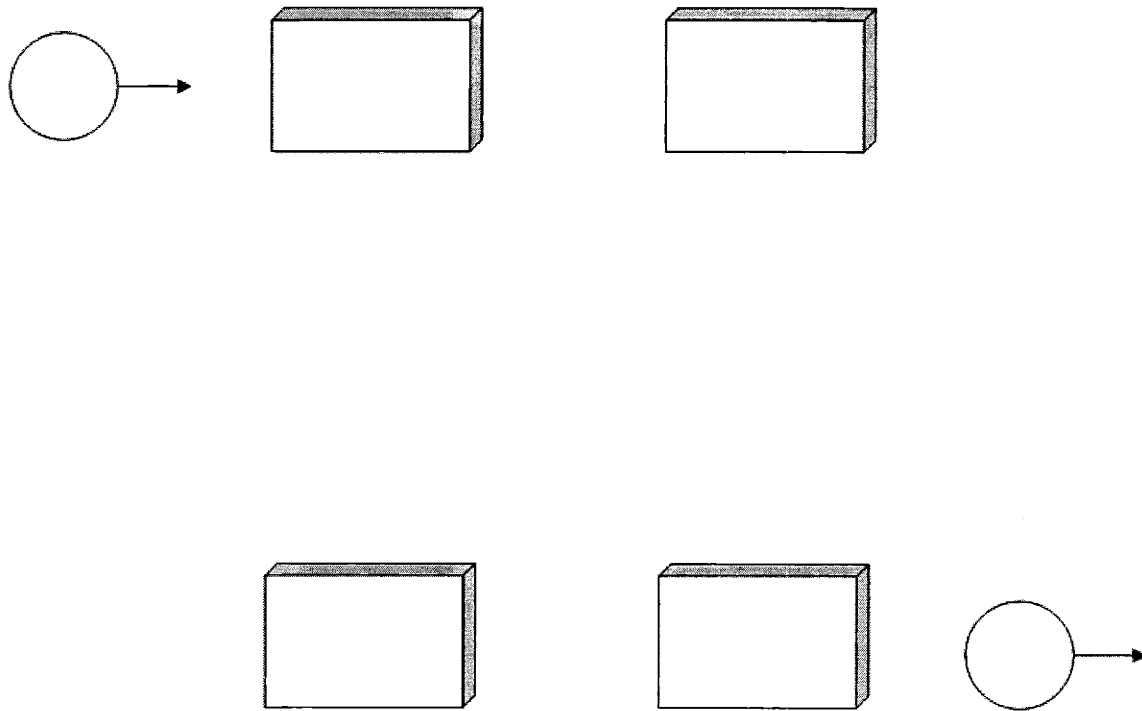


Figure 10. This scenario is exactly like the scenario in figure 9 except for that the middle block is missing. S6 infers that the ball that went behind the screen cannot be the same as the ball that came out of the screen because it did not see the ball go between the screens.

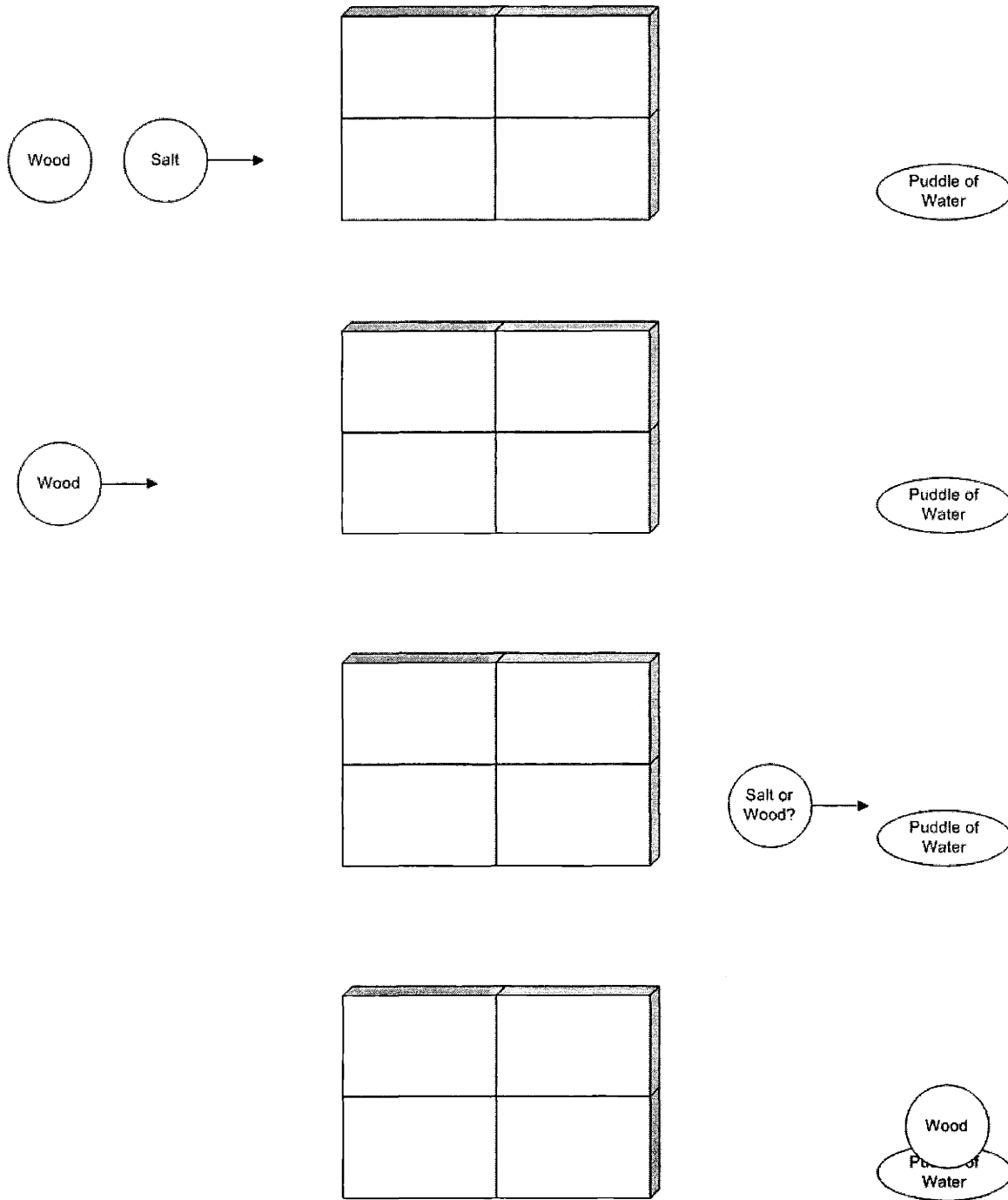


Figure 11. In this scenario, S6 sees the ball come out from behind the blocks and must reason about it with knowing if it is the wooden ball or the salt ball. When the ball ultimately does go into the puddle of water and does not melt, S6 infers that it is not made of salt and that it therefore must be the wooden ball.

When S6 sees the ball emerge from behind the right block, the identity hypothesis specialist suggests that this ball is the same ball that went behind the left block because they have the same size and shape. As in the previous example, the attribute specialist detects a change in position and the path specialist matches a path script with an intermediate point between the two scripts. When the perception and space specialists try to find that intermediate point, they fail because they remember seeing that all the intermediate points were empty during the scenario. The path specialist then concludes that a change of position did not happen and thus the attribute specialist concludes that these balls are not the same.

6.3.5 Postponing identity

The world S6 lives in is like traditional “blocks worlds” in artificial intelligence research with one important difference. In most blocks worlds there is no question about the identity of an objects. When a system perceives block17, e.g., at time 3 and it perceives that object at a later time, that block has the same label. Those blocks worlds thus did not require systems to deal with the identity problem.

In this example, S6 is able to think about an object without establishing its identity and then, after having received decisive information, firmly establish the object’s identity. In Figure 11, two blocks go behind the screen. S6 is told that the first block is made of salt and the second is made of wood. When a block comes out of the right side of the screen and looks like both blocks, S6’s identity hypothesis specialist thinks that this could be the salt block, S, or the wooden block, W. When the block moves to the right into a puddle of water and does not begin to melt, then S6’s causal specialist concludes the new block is made of wood and the identity specialist then concludes that the new block is W and not S.

Chapter 7

Conclusions

I began this thesis by describing the problem of integrating several computational techniques and knowledge representation schemes into one intelligent system. Instead of trying to find a single paradigm that subsumed all techniques, I decided to develop an architecture for integrating components based on each. I have explained the design of Polyscheme in this thesis and shown how to build a versatile common sense reasoner, S6, with it. S6 demonstrates that the principles of Polyscheme enable a significant advance in machine intelligence.

Polyscheme is based on three principles. First, different (aspects of) situations that intelligent systems must deal with are best modeled with different schemes for representing knowledge and making inferences. Polyscheme includes several specialists that each specializes on some aspect of the world by using specialized representation and inference techniques.

Second, specialists must communicate with other specialists frequently so that each specialist uses the most complete, accurate and relevant information when it deals with a situation. Specialists in Polyscheme communicate and combine information by simultaneously concentrating on the same focus of attention.

Finally, because information about some aspects of a situation is more important than other aspects and because the order that specialists focus on those aspects is important, a system of focused specialists must have mechanisms that decide where to focus. Polyscheme's specialists, especially the reflective specialist, use the attraction mechanism to decide where to focus the specialists' attention.

Polyscheme enables multiple inference techniques to be integrated in dealing with a situation because each inference technique can be implemented with one or more focus

schemes, which Polyscheme is designed to easily combine. I described how to implement several important inference techniques (e.g., script matching, backtracking search, reason maintenance, stochastic simulation and counterfactual reasoning) as focus schemes.

I have used Polyscheme to implement the S6 system for common sense physical reasoning. S6 views interactions in a simple physical world through a 2-dimensional projection of that world. S6 keeps track of the identity of objects, infers the character and existence of events it cannot see, predicts the outcome of events, explains events and nonevents, and revises its inferences when it receives new information. S6 successfully reasons about many scenarios researchers present to infants and young children to study their knowledge of the physical world.

S6 combines specialized representation and inference techniques for identity, time, events, causality, space and paths to successfully deal with a wide range of situations. The knowledge representations schemes S6 uses include scripts, frames, logical propositions, neural networks and constraint graphs. The inference schemes S6 implements include script matching, rule matching, backtracking search, neural network propagation and counterfactual reasoning.

These representation and inference schemes form part of a common sense substrate that underlies much of human cognition. The success of S6 therefore demonstrates that Polyscheme is a powerful model for understanding and building intelligent systems in most any domain of human cognition.

Appendix

Here are some sample inference schemes. Many are not original and many more could be added. The list is only intended to show that common sense reasoning involves a wide variety of inference schemes.

Identity

1. *Look for identity.* When you see an object, look for objects have seen in the past that it can be identical to.
2. *Persistent-property comparison.* Some properties of objects are persistent. When you have two objects that might be identical, compare these attributes to confirm identity.
3. *Path-search.* If two object sightings are possible of the same object, then look for a spatio-temporally continuous path between them.
4. *Changed property comparison.* When an object changes in an attribute look for a cause for that change.
5. *Ambiguous-identity resolution.* If you are uncertain of the identity of an object, find out more of its attributes (in hopes that these will give you discriminating information). Look at attributes that vary across possibilities.

Temporal

6. When you see the same relation holds at two different, but adjacent, intervals combine them and simulate the result.
7. When you see the same relation at two different moments, posit that it was true over the intervening interval.

8. When you posit something as true over an interval, look for assertions that confirm or falsify this.
9. Project persistent properties for a long time forward and a long time backwards.

Understanding other people

10. If X wants G, then look for plans that x can execute to get G and predict that he will chose one of those plans.
11. If X does A, look for plans that involve A and satisfy a goal that X has.
12. If X does something unexpected, find something about the situation that X might not know.

Categorization

13. *Ambiguous-kind resolution.* Execute ambiguous identity resolution, except be less stringent in property comparisons.
14. *Look for confirmation.* When you wonder if an object is of a certain category, look to see if it has the properties objects of that category typically have.

Problem solving techniques

15. *Case-based reasoning.* When you are uncertain of a property of an object, find a similar object and see if it has that property.
16. *Search.* If you are unsure which of two actions you should execute, think about the consequences of one for a while and then think about the consequences of the other.

17. *Difference reduction.* If you want X1 but you have X2 and action, A, reduces X2-X1, then do A.
18. *Means-ends analysis.* If you want to achieve G and an action A achieves G then either do A or try to make the preconditions of A true.
19. *Plan anticipation.* Whenever you are executing a plan, be constantly simulating the next few actions and their consequences to find dangers or flaws in the plan.
20. *Bug finding.* When something does not work as you expected, resimulate the event in much greater detail, examining the cause for each step.
21. *Apply remedy.* If an event causes pain, look for an action that will reverse the event or neutralize the pain.
22. *Subgoal interaction.* If you've established certain relations in the world to help achieve one subgoal, do not let actions for another subgoal undo it.
23. *Hierarchical planning.* First make a plan with general actions and once you find one fill in the details.

Learning

24. *Uniframe induction.* When you successfully match a case, see what the two cases had in common and make a general frame or script that abstracts over both cases.
25. *Explanation-based learning.* When you go through a successful chain of inferences, go through it again being as vague and general as possible.
26. *Parameter setting.* When each extreme of an action produces different results, try intermediate extents to get the threshold for the difference.

27. *Confirmation with memory.* When you form a hypothesis, check if it coheres with what you already know.

28. *Confirmation by experiment.* When you form a hypothesis, confirm its predictions in various novel circumstances.

29. *Try supercategory.* When you find a regularity about a category, see if it works on supercategories.

30. *Try subcategory.* If a regularity you induced on a category fails, look for a subcategory it still holds in.

Miscellaneous

31. *Consistency checking.* When you assert something about the attribute of an object, see if you have asserted something different in the past.

32. *Bookkeeping.* When you see that an attribute of an object has one value, erase assertions that it has other values.

33. *Script application.* When you apply a script, make sure the goals each action in the script achieves are achieved in the present circumstances.

34. *Truth-maintenance.* When you find that you were wrong about an object's properties, resimulate the events and inferences that that object participated in (especially., those involving that property).

35. *Conflict management.* If you predict two different values for an attribute, try to perceive that attribute's value to rule out the incorrect prediction and the assumptions that led to it.

36. *Conflict management.* When you make two conflicting predictions, re-derive the predictions at a more elaborate level of detail, hoping that one of the prediction-traces assumed something that is not so in this situation.

37. *Finding things out.* If $x \rightarrow y$, and you want to know y , then try to find x out.

38. *Falsification.* If $x \rightarrow y$ and you see not- y , then assert not- x .

39. *Check for opportunities.* When you find something new about the world, look for a goal whose statement or partial plan involves this region of the world in case this might help achieve that goal.

40. *Explaining away.* If you assumed a because you wanted to explain x and $a \rightarrow x$ and you see that b is true and $b \rightarrow x$, then retract a .

41. *Frame matching.* If an object or event typically has certain parts, look for those parts when you think you are viewing that object or event.

Bibliography

Allen, J. F. 1983. *Maintaining knowledge about temporal intervals*. Communications of the ACM, 26:832--843.

Baars, B. J. 1988. *A Cognitive Theory of Consciousness*, Cambridge University Press.

Barwise, J. & Perry, J. 1983. *Situations and Attitudes*. MIT Press, Cambridge, MA.

Doyle, J. 1979. *A truth maintenance system*. Artificial Intelligence, 12:231-272.

Doyle, J. 1992. *Reason maintenance and belief revision: Foundations vs. coherence theories*. In Peter Gardenfors, editor, *Belief Revision*, pages 29--51. Cambridge University Press, Cambridge, 1992.

Hermer, L., & E. Spelke. 1996. *Modularity and development: The case of spatial reorientation*. *Cognition* 61:195-232.

Jackendoff, R. 1990. *Semantic Structures*. Cambridge, MA: MIT Press.

Kestenbaum, R., Termine, N., & Spelke, E. S., 1987. *Perception of objects and object boundaries by 3-month-old infants*. *British Journal of Developmental Psychology*, 5, 367 - 383.

Miller, A. G. & JohnsonLaird, P. N. 1976. *Language and Perception*. Cambridge Univ. Press, Cambridge.

Minsky, M. 1986. *The Society of Mind*. New York: Simon and Schuster.

Minsky, M. 1992. *Future of AI Technology*. *Toshiba Review*, Vol.47, No.7, July 1992.

Stroop, J. R. 1935. *Studies of interference in serial verbal reactions*. *Journal of Experimental Psychology*, 18, 643-622.

Talmy, L. 1985. *Force Dynamics in Language and Thought*. Papers from the Parasession on Causatives and Agentivity.

Treisman, A. M. & Gelade, G. 1980. *A feature integration theory of attention*. *Cognitive Psychology*, 12, 97-136