RA: A Memory Organization to Model the Evolution of Scientific Knowledge

Kishore Swaminathan

Computer and Information Science Department
University of Massachusetts

COINS Technical Report 90-80
September 1990

RA: A MEMORY ORGANIZATION TO MODEL THE EVOLUTION OF SCIENTIFIC KNOWLEDGE

A Dissertation Presented

by

KISHORE S. SWAMINATHAN

Submitted to the Graduate School of the
University of Massachusetts in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 1990

Computer and Information Science

This research was supported by the Advanced Research Projects Agency of the Department of Defense, monitored by the Office of Naval Research under Contract #N00014-87-K-0238, by the Office of Naval Research, under a University Research Initiative Grant, Contract #N00014-86-K-0764 and by an NSF Presidential Young Investigators Award NSFIST-8351863.

To
my parents
Neela and Sundaram

my mentor
C.R. Muthukrishnan

and the memory of my sister
Pappa

RA: A MEMORY ORGANIZATION TO MODEL THE EVOLUTION OF SCIENTIFIC KNOWLEDGE

A Dissertation Presented

by

KISHORE S. SWAMINATHAN

Approved as to style and content by:

Wendy Lehnert, Chair

Paul Cohen, Member

Edwina Rissland, Member

Charles Clifton, Member

Gerald DeJong, Member

W. Richards Adrion, Department Chair
Computer and Information Science

Acknowledgments

The graduate school of the University of Massachusetts says that a doctoral dissertation may have an optional section to acknowledge the author's professional and personal indebtedness, provided it is written in a "professional manner." Professional manner, indeed. The best I can do is not break down and cry.

Wendy Lehnert, my advisor, gave me a totally free hand and let me chase whatever whimsies I wanted to. Still, she nudged me, ever so gently, away from the "parsing pit." When I changed thesis topics more frequently than I changed shirts, she kept her hope that, someday, I'll try some new shirts. Even before I could articulate an idea, Wendy usually knew, in an almost spooky sort of way, what it was, and where it could lead. Her influence on me — both personally and professionally — is very subtle and very real. Perhaps the most important thing that I learned in the last four years came from watching Wendy: a person could not only *have* ideals, but could even *practise* them quite successfully. I thank Wendy for being the best thesis advisor I could have had.

I would also like to thank the other members of my dissertation committee for their input into this work. Paul Cohen is never easy to negotiate, but when you do, the returns are always high. He asks the most substantiatial and difficult questions, and expects you to — oh, my God — *think*. In fact, after a particularly exhilerating meeting with Paul late one friday evening, I was so inspired that I finished my thesis proposal during the weekend. For that meeting and several others, I am very grateful to Paul.

Edwina Rissland has given me a number of suggestions and comments about this work. In particular, her comments led to several ideas in Chapter 5. For these, and for supporting this work in her capacity as the director of the Case-Based Reasoning lab, I thank Edwina most sincerely.

I am also grateful to Chuck Clifton and Gerry DeJong, the two "outside" members of my committee. As a psychologist, Chuck gave me his honest assessment of the way I treat and interpret the psychology literature. Despite his reservation, "But is it science?", Chuck's assurance that it is nevertheless interesting has been a source of relief. I am thankful to Chuck for his many comments and his input into Chapter 5. Gerry DeJong hosted me for almost a week in his laboratory at Illinois, and let me air my ideas with members of his lab. For this opportunity, and his many comments, I am very grateful.

Several friends at UMass have helped me in the process of doing this dissertation. I wish to record my gratitude to them all: to Jamie Callan, David Day, Adele Howe, Philip Johnson, and Dan Suthers for being part of the Thesis Liberation Front; to Ellen Riloff for being a special friend; to Stefan Wermter for going to the hippie festival with

me; to my office-mates for putting up with my hap-hazard schedule; and to Priscilla Coe for her superb handling of everything administrative. I also wish to thank the following wonderful people for their support: Alex, Edie and Ann Vanderburgh, Ken and Lucy Myles, Karen and Eduardo Valcarce, K. Raghunathan and S. Srinivasan.

Faith Vanderburgh, my wife, counsel, and closest friend, has kept me warm and comfortable, and retained her hope that, someday, I will finish. And the day has come. For all her selflessness and charity, I thank her with my heart. For keeping Faith sane during the year I was too immersed in this thesis, I thank Henry the moose, Draggie the dragon, and Placido the platipus.

Abstract

This dissertation addresses the dichotomy between semantic and episodic knowledge by focusing on the evolution of scientific knowledge. Even timeless scientific knowledge about the nature of the world accrues only through discrete episodes, with each scientist building upon the work of his/her predecessors. Hence, a memory organization to model the knowledge of a scientific field should reflect not only the knowledge pertaining to the field, but also the knowledge pertaining to the evolution of the field. A computer program called *RA* is described: RA proposes a memory organization for scientific knowledge in terms of a representational idea called *Research Schemas*. Research Schemas view research papers, not as isolated pieces of text, but as related episodes that contribute to the growth of a scientific discipline. This memory organization is validated by showing that it supports a number of different capabilities: it enables RA to suggest new research directions, acquire new research schemas, retrieve papers that have similar research strategies, and generate both chronological and analogical summaries of research papers. A combination of these capabilities constitutes a framework for 'Computer-Aided Research.'

The RA system also includes a learning technique to acquire new research schemas. While similarity-based techniques use multiple examples (and some form of encoded bias) and explanation-based techniques use a domain theory as the basis for generalization, there is no apparent basis for RA's generalization. An analysis of RA's learning strategy shows that the category structure of RA's world provides a basis for its generalization: RA generalizes instantiations into categories that are both associative and discriminative. Interestingly, this turns out to be precisely the property that characterizes *basic-level* categories that have been studied by psychologists. This dissertation explores the implication of this result to learning and knowledge representation.

Contents

List of Tables

List of Figures

Chapter 1

Overview

The white houses sank into the sea as quickly as they had appeared. We were drifting fast. Good-by, Africa. Good-by, Old World. We have no rudder. We don't need one on this voyage.

—Thor Heyerdahl, The RA Expeditions.

RA is a computer program. RA is an answer to the question, "What kind of representation best captures the evolving nature of knowledge?" More specifically, RA is an answer to the question, "What kind of representation for scientific literature best captures the evolution of science?" RA is also an answer to a related question, "What kind of representation best captures a scientist's competence in one's field?" Finally, RA is my vision of what a 'Computer-Aided Research' environment might look like. RA can perform several tasks in the domain of scientific research:

> (1) it can suggest new research directions and (2) provide relevant reading material; (3) it can learn new research strategies from individual research papers; (4) it can find analogies among research papers; and (5) it can summarize the trends in the field.

All of RA functionalities are supported uniformly by a single representational idea called *research schemas*. Research schemas have two important features: (1) they view research papers not as isolated pieces of text, but as mutually related events through which the knowledge in a research field evolves, and (2) they view research papers at a high level of abstraction to capture the structural relationships among them. For concreteness, RA performs the above tasks within the field of 'Explanation-Based Learning' (EBL).

The RA program and the domain of research literature, while interesting in their own right, are mere crutches to let us explore some fascinating issues in memory, language, and learning. Table 1.1 contains a guide to Chapter 1.

Section	On first reading	Description
1	read	Describes RA. Table 1.2 contains a synopsis.
2	read	Describes the motivations for this research.
3	read	Describes the issues addressed by this work.
4	read	Describes the scope and limitations of this work.
5	skim	Describes how this dissertation is organized.
6	read	Contains a summary of this chapter.

Table 1.1: Guide to Chapter 1

1.1 The RA System

1.1.1 An Example

Consider a typical interaction between a university professor and a graduate student:

> The student states an interest in some problem. The advisor places the problem in a larger context, suggests several directions for attacking the problem, and offers some reading material. End of session.

RA can be viewed as duplicating the functionalities of a research advisor. Shown below is a hypothetical dialog between a professor and a student:

Student: (1) I am interested in EBL and I am looking for some research directions.

Professor: (2) EBL solves the learning-problem. There is a lot of work on EBL. [Mitchell et al 86] puts all this work in a single framework. Perhaps you could try to find out what the limitations of EBL are.

Student: (3) How do I do that?

Professor: (4) One strategy for finding the limitations of a technique is to see if it really solves the problem it purports to solve. Can you find subclasses of learning problems such that EBL solves one subclass but not the other? For example, this is what [Minsky & Papert 69] did with perceptrons[1,2].

[1] ML After [Rosenblatt 57] proposed perceptrons as a technique to learn boolean functions, Minsky and Papert showed that the learning problem for boolean functions consists of two kinds: learning problem for boolean functions that are linearly separable, and for those that are not. They showed that perceptron learning algorithm converges for the former but not for the latter. In general, you don't need to understand the details of machine learning and EBL to follow the dialog; ignore the details and focus on the strategies.

[2] For ease of reading, footnotes in this dissertation are marked in several ways. Section 1.5 lists these marking conventions.

(Assume that the thoughtful student goes away for a couple of weeks, and comes back with a Eureka!)

Student: (5) EBL assumes that you have perfect knowledge of your domain. For a lot of domains this assumption is untenable. So EBL cannot solve this class of 'imperfect-domain problems.'

Professor: (6) Maybe you can now investigate how EBL can be adapted to handle these domains. Since similarity-based techniques can solve learning problems in general, maybe you can combine a similarity-based technique with EBL to solve 'imperfect-domain problem.' 'Version Spaces' is a good example of a similarity-based technique.

Student: (7) I never heard of 'Version-Spaces'. What is it about?

Professor: (8) [Winston 71] proposed a depth-first search technique to solve the concept learning problem. [Vere 75] pointed out a deficiency in Winston's technique and proposed a new method that avoids this deficiency. [Mitchell 78] identified that Vere's techniques had another deficiency. Mitchell proposed his 'Version Spaces' technique to solve the concept-learning problem while avoiding this deficiency[3].

Student: (9) Gee, is 'Version Spaces' technique perfect?

Professor: (10) No, it has a deficiency too, known as the 'fixed representational bias problem'. [Utgoff 84] has proposed a method to solve this problem using a 'bias adjustment' technique.

(Impressed at the guru's wisdom, the student saunters off to think up a technique to solve the 'imperfect-domain problems.')

Student: (11) I have come up with a technique that I will call the 'Super-duper-hybrid-technique'. It can solve the imperfect-domain problem by combining explanation-based learning and similarity-based learning.

Professor: (12) Great! You know what you can do now? [Mitchell 81] showed that all similarity-based learning techniques can be viewed as search. Since yours is a hybrid technique that combines a similarity-based technique with another technique, it might be interesting to see how the search property applies to your hybrid technique...

---

[3] ML When Winston's arch learner is given a positive or negative instance, it has to make sure that its new hypothesis is consistent with all the other instances it has seen in the past; in other words, it has to backtrack. Vere suggested that by using a partial-order among concept descriptions, you need to backtrack only for negative instances and not for positive instances. Mitchell's Version Space approach, by maintaining a most general as well as a most specific concept hypothesis, avoids all backtracking.

The above dialog is a doctored version of an actual dialog between a user and RA, with RA playing the role of the professor and a user playing the role of a student[4]. During the dialog, RA displays the text of the papers it refers to through its hypertext interface. The dialog exhibits several capabilities of RA: RA's ability to suggest research strategies and to provide relevant reading material (interactions 2, 4, 6, and 12), its ability to indicate other places where a research strategy has been used (4), its ability to place a research paper in its historical context (8 and 10), and its ability to reason about the new-terms ('imperfect-domain-problems,' 'super-duper-hybrid-technique') introduced by the user (6 and 12). Variations of these capabilities enable RA to summarize papers in terms of underlying chronological and analogical relationships (see Chapter 3). Finally, RA can also learn the research strategies it uses for suggesting research directions (Chapter 4 and Section 1.1.4 in this chapter). Let's first look at how RA represents research papers.

1.1.2 Representation: Research Schemas

RA's representation for research papers has two important features: (1) papers are viewed as discrete events that have strong connections among them, and (2) papers are viewed at a very high level of abstraction to capture the structural relations among problems, techniques, properties etc.

These two features of the representation give rise to the following claim: "When research papers are viewed this way, one can find standard schema-like patterns of research recurring in science." I'll call these patterns *Research Schemas.* Let's consider some examples:

1. Propose a problem and propose a technique to solve it. For example, [Winston 71] proposed the concept learning problem and a depth-first search technique to solve it.

2. Given a problem P1 and a technique T1 to solve it, identify an emergent problem P2 (a hole, a deficiency) in T1. Propose a new technique T2 that solves P1 while avoiding P2. For example, [Mitchell 78] proposed the version-space technique to solve the concept-learning problem while avoiding the emergent problem of backtracking in Winston's technique[5].

3. For a given problem P1 and a technique T1 to solve it, identify an emergent problem P2 in T1. Propose a technique T2 *to solve P2.* For example, [Utgoff

[4]RA does not do any discourse management; what I showed here is a pruned form of a much larger dialog. Further, all interactions with RA take place through the 'mouse.' The dialog is an English translation of 'mouse-talk.' Chapter 3 revisits this dialog to discuss it in detail.

[5]For the moment, let's ignore Vere's contribution, and credit [Mitchell 78] for the identification and avoidance of the backtracking problem.

84] proposed a bias adjustment technique to solve the emergent problem of fixed representational bias in the version-space technique.

4. For a given problem P1 and a technique T1 to solve it, show that P1 consists of two subclasses P2 and P3 (i.e., P1 dominates P2 and P3). Show that T1 solves one subclass P2, but does not solve the other subclass P3. For example, [Minsky & Papert 69] showed that the learning problem for boolean functions consists of two subclasses — learning problem for those that are linearly separable and for those that are not linearly separable. The perceptron learning algorithm can solve the former but not the latter.

Figures 1.1 and 1.2 depict the instantiated and uninstantiated versions of the above four schemas. Before I explain the difference between the solid lines and the dashed lines in these figures, let us consider how a 'canonical abstract' for the above papers might look:

> [Winston 71]: In this paper, I propose a concept learning problem in the context of learning the structural description of arches. I propose a depth-first search technique to solve it.
>
> [Mitchell 78]: Winston proposed a depth-first search technique to solve the concept learning problem [Winston 71]. In this paper, I show that Winston's technique has an emergent problem called 'backtracking.' I propose a technique called 'Version-Spaces' that solves the concept-learning problem while avoiding the backtracking problem.
>
> [Utgoff 84]: Mitchell proposed the version space technique to solve the concept learning problem [Mitchell 78]. In this paper, I show that the version space technique has an emergent problem called 'fixed representational bias'. I propose a bias-adjustment technique to solve the fixed bias problem.
>
> [Minsky & Papert 69]: Rosenblatt proposed the perceptron learning technique to solve the learning problem for boolean functions [Rosenblatt 57]. In this paper, we show that this learning problem consists of two subclasses — learning problems for boolean functions that are linearly separable and for those that are not linearly separable. We show that the perceptron algorithm can solve the former but not the latter.

In order to state its motivations, each paper first states how it relates to the current state of knowledge in the field. Then it adds some new knowledge to the field. And so the cycle continues, and the knowledge in the field evolves...

A research schema is a compact representation for papers. It consists of two sets of relations called **ref** and **def**. **ref** is the set of relations that a paper references, and **def** is the set of relations that the paper defines. For example, the schema for [Mitchell 78] is shown below:

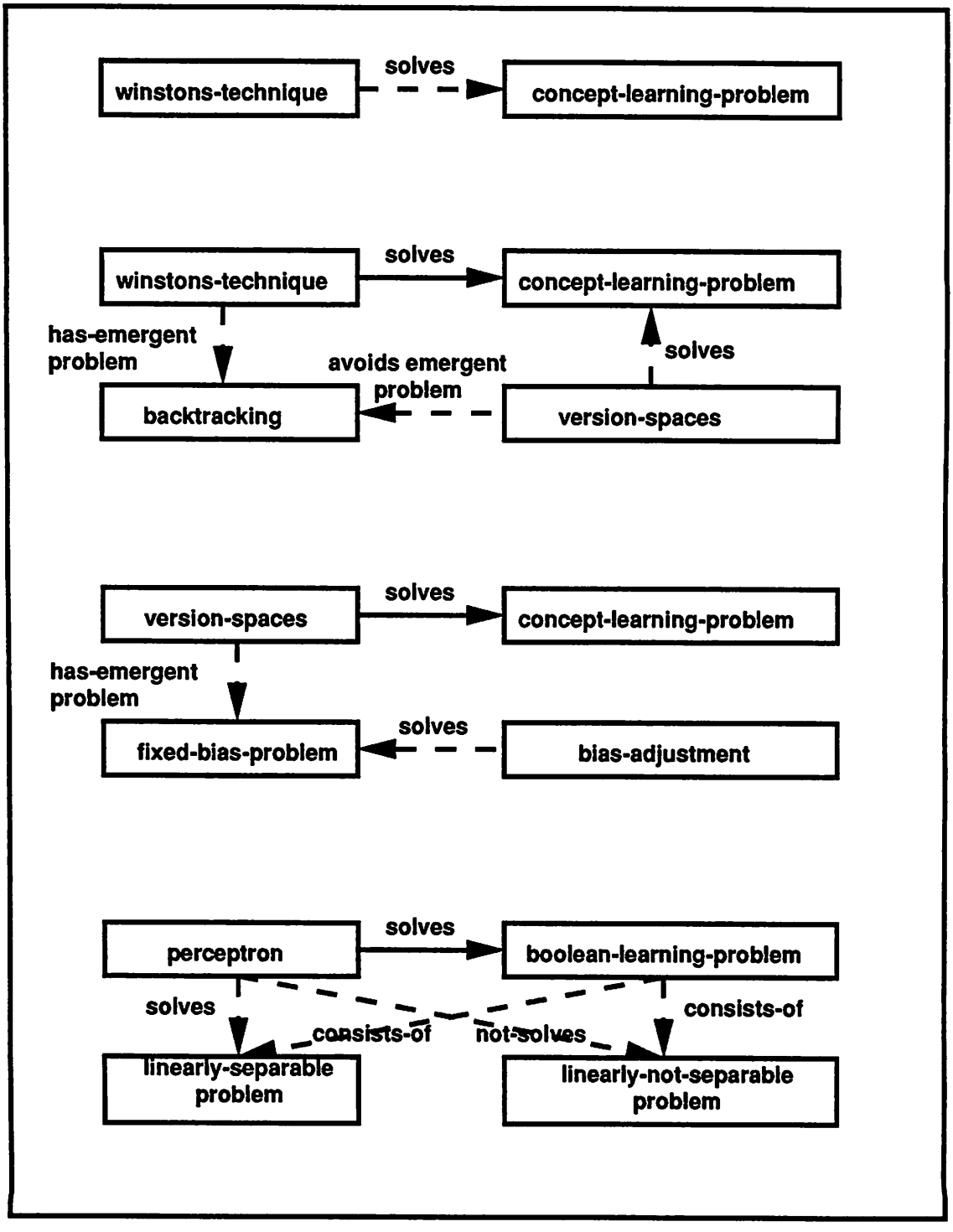


Figure 1.1: Some Research Schemas

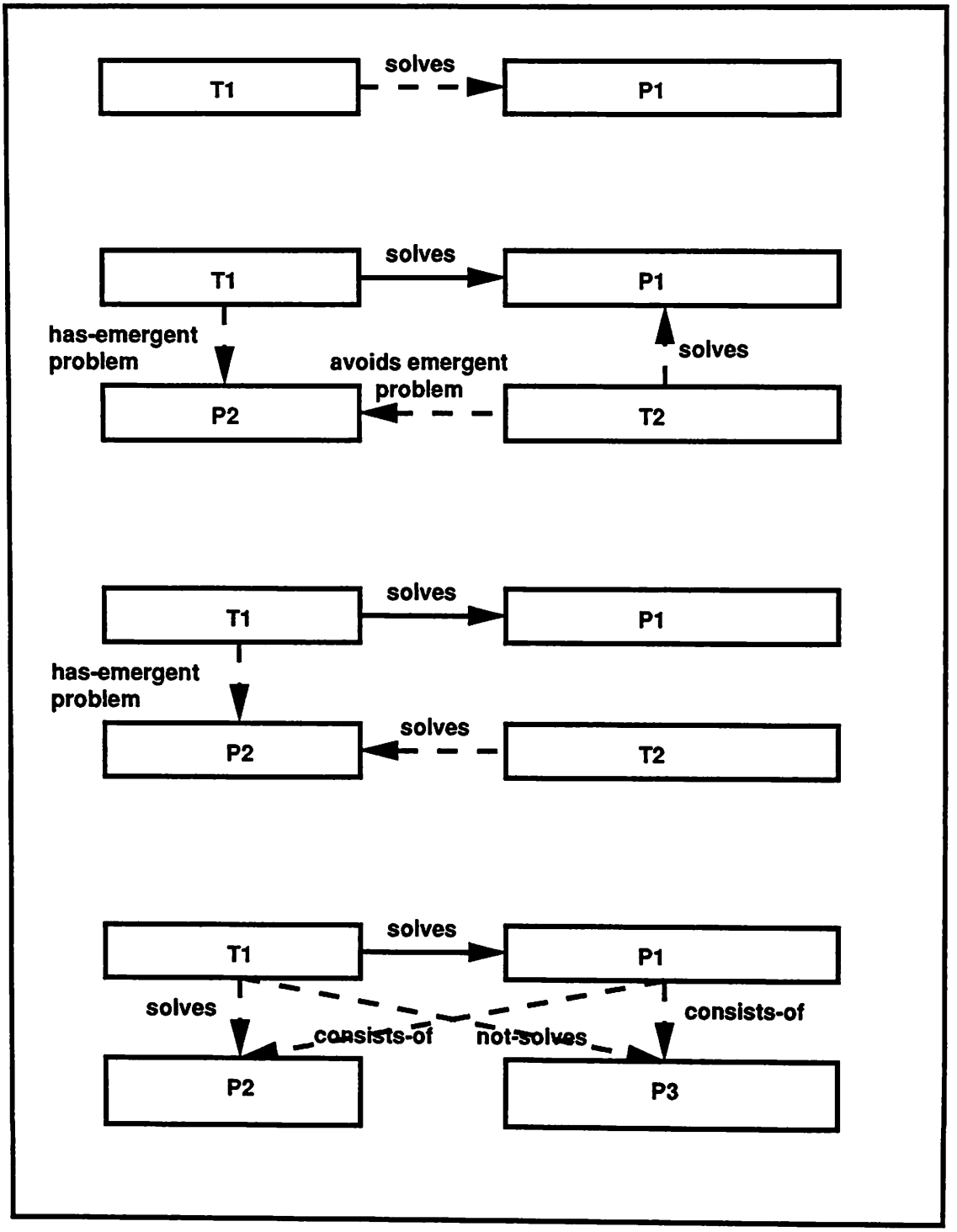


Figure 1.2: Uninstantiated (Skeletal) Research Schemas

ref: {(solves winstons-technique concept-learning)}

def: {(entails[6] winstons-technique backtracking)
(solves version-spaces concept-learning)
(not-entails version-spaces backtracking)}

Note that the term **ref** was chosen to be suggestive, but it does not denote the actual references in the text of the paper. Instead, **ref** denotes a set of semantic relations in RA's knowledge base. These relations provide the immediate context for the relations in the **def** of the paper. I will discuss **ref** and **def** in detail in Chapters 2 and 4.

Figure 1.1 depicts the above schemas pictorially. Solid lines denote relations in **ref** and dashed lines denote relations in **def**. Figure 1.2 shows the uninstantiated schemas. Whenever I want to emphasize the distinction, I will refer to an uninstantiated schema as a 'skeletal schema.'

1.1.3 Schemas as Heuristics and Indices

So far, research schemas have been discussed as a representational mechanism. This is only one use for research schemas. RA uses research schemas also as heuristics to suggest research directions, and as memory indices to to index papers that have similar research strategies. Let me first show how they are used as heuristics. Below is the skeletal schema of [Mitchell 78]:

ref: {(solves T1 P1)}

def: {(entails T1 P2)
(solves T2 P1)
(not-entails T2 P2)}

A skeletal schema is used as an 'if-then' heuristic where the relations in **ref** play the role of the 'if' part, and the relations in **def** play the role of the 'then' part. For example, the rule below is derived from the schema above:

> If there exist T1, P1 such that T1 solves P1, then suggest "You can try to identify some emergent problem P2 of T1. Then you can propose a technique T2 that solves P1 while avoiding P2."

When a user states an interest in something, RA finds the corresponding node, say x, in its knowledge base. This node is called an anchor node. After finding the anchor node x, RA collects all the heuristics whose 'if' conditions are satisfied in the neighborhood of x. This means that (1) some variable in the condition part of the heuristic can be

[6] From now on, 'entails' will be used as a short form for 'has emergent problem.'

instantiated (bound) to x, and (2) there is an instantiation for every other variable in the condition part of the rule. When this is true, the rule is said to be applicable. For example, assume that the user states an interest in learning-problem. The above rule is applicable because (1) P1 can be instantiated to learning-problem and (2) T1 can be instantiated to EBL (because EBL is a technique that solves the learning-problem).

Once the 'if' condition is satisfied, RA generates the suggestion of the 'then' part by maintaining the variable bindings consistent across the condition part and the suggestion part. With the above rule, it generates the suggestion, "You can find some emergent problem P2 of EBL. Then you can propose a technique T2 that solves the learning-problem while avoiding P2."

Now the use of schemas as an indexing mechanism comes into play. In RA, papers are represented as instantiated research schemas. The instantiated schemas are indexed on their skeletal schemas (from which heuristic rules are derived). When a heuristic is executed, RA first finds the corresponding skeletal schema. Then it finds all the papers indexed on that skeletal schema. These papers are provided to the user as examples of how that research heuristic has been used before. In a previous paragraph, I showed how RA uses the heuristic from [Mitchell 78] to suggest that a user could try to identify an emergent problem of EBL. In addition to that suggestion, RA also retrieves the text of [Mitchell 78] as an example of where such a strategy has been used before.

1.1.4 Acquiring New Schemas

The previous sections described one version of the RA system. Let's call it RA I. Since the abilities of RA depended so heavily on the notion of 'research schemas,' an effort was undertaken to acquire research schemas automatically. A second version of the system, RA II, was built for this purpose[7,8]. It was found that RA II could indeed learn research schemas automatically; further it was found that no new representational apparatus was needed to achieve this capability. In other words, RA's representation of research literature (as research schemas) was also an adequate representation from which to learn new research schemas[9].

The input to RA II is the knowledge defined by a paper, i.e., its **def**. RA automatically infers its **ref**. Then it converts this schema into a skeletal schema by replacing constants by typed variables. For example, assume that EBL and SBL (expansion: Similarity-Based Learning) are two techniques to solve the 'learning-problem.' SBL has a known property that "All similarity-based techniques can be viewed as search"

[7] Btw At this point some readers may be reminded of Lenat's AM spawning Eurisko, and DeJong's FRUMP spawning GENESIS. Why? RA II, Eurisko, and GENESIS all have the same skeletal schemas. Ponder that one! What about the relationship between SAINT and LEX?

[8] Btw The phrase 'Ponder that one!' is lifted from a famous AI dissertation. Ponder that one!

[9] Btw At this point, the analogy between RA II and Eurisko breaks. For a comparison of RA and AM, see Chapter 6 (section 6.3.1).

[Mitchell 81]. Let's call it search-property. Now RA is given a new paper, '[Rosenbloom 88].' The **def** of this paper is:

{(exhibits SBL Rosenbloom88-property)
(exhibits EBL Rosenbloom88-property)
(generalizes Rosenbloom88-property search-property)[10]}

This paper says that it is proposing a new property called Rosenbloom88-property that is a generalization of search-property and that this new property applies to EBL and SBL. To find the **ref** of this paper, RA II assumes that there is something about EBL, SBL, and search-property that enabled Rosenbloom's research. In other words, Rosenbloom did not arbitrarily pick two random techniques and a random property and combine them all to write a paper[11]. So RA II attempts to find a connection among EBL, SBL, and search-property. First it finds that search-property is in fact a direct property of SBL. Next it finds that the relation between EBL and SBL is that they both solve the same problem, namely, learning-problem. Now RA II infers the following schema for [Rosenbloom 88]:

ref: {(solves EBL learning-problem)
(solves SBL learning problem)
(exhibits SBL search-property)}

def: {(exhibits EBL Rosenbloom88-property)
(exhibits SBL Rosenbloom88-property)
(generalizes Rosenbloom88-property search-property)}

This schema corresponds to the following canonical abstract:

> SBL [ref] and EBL [ref] are two techniques to solve the learning problem. Mitchell [Mitchell 81] has shown that SBL can be viewed as search. In this paper, I generalize Mitchell's view so that it applies not only to SBL but also to EBL[12].

From this schema, RA II learns a skeletal schema by replacing constants by typed variables. This results in the following:

ref: {(solves T1 P1)
(solves T2 P1)
(exhibits T2 Pr1)}

[10] ML Actually, this paper generalizes search-property to apply to EBL, SBL, and rote-learning.

[11] Biw Newell would call this the rationality assumption [Newell 82], i.e., there is an inherent assumption that Rosenbloom is rational. RA II's learning, in general, cannot be considered rational under Newell's definition, and RA II may be said to learn at the 'knowledge level' (see [Dietterich 86] for a discussion on knowledge level learning). However, it is rational in a more heuristic sense proposed by [Rosch et al 76]. Most of Chapter 5 is devoted to this topic.

[12] This abstract is a reasonable summary of the introductory section of [Rosenbloom 88].

def: {(exhibits T1 Pr2)
(exhibits T2 Pr2)
(generalizes Pr2 Pr1)}

This skeletal schema is depicted in Figure 1.3. To understand it, let's look at it as a research heuristic:

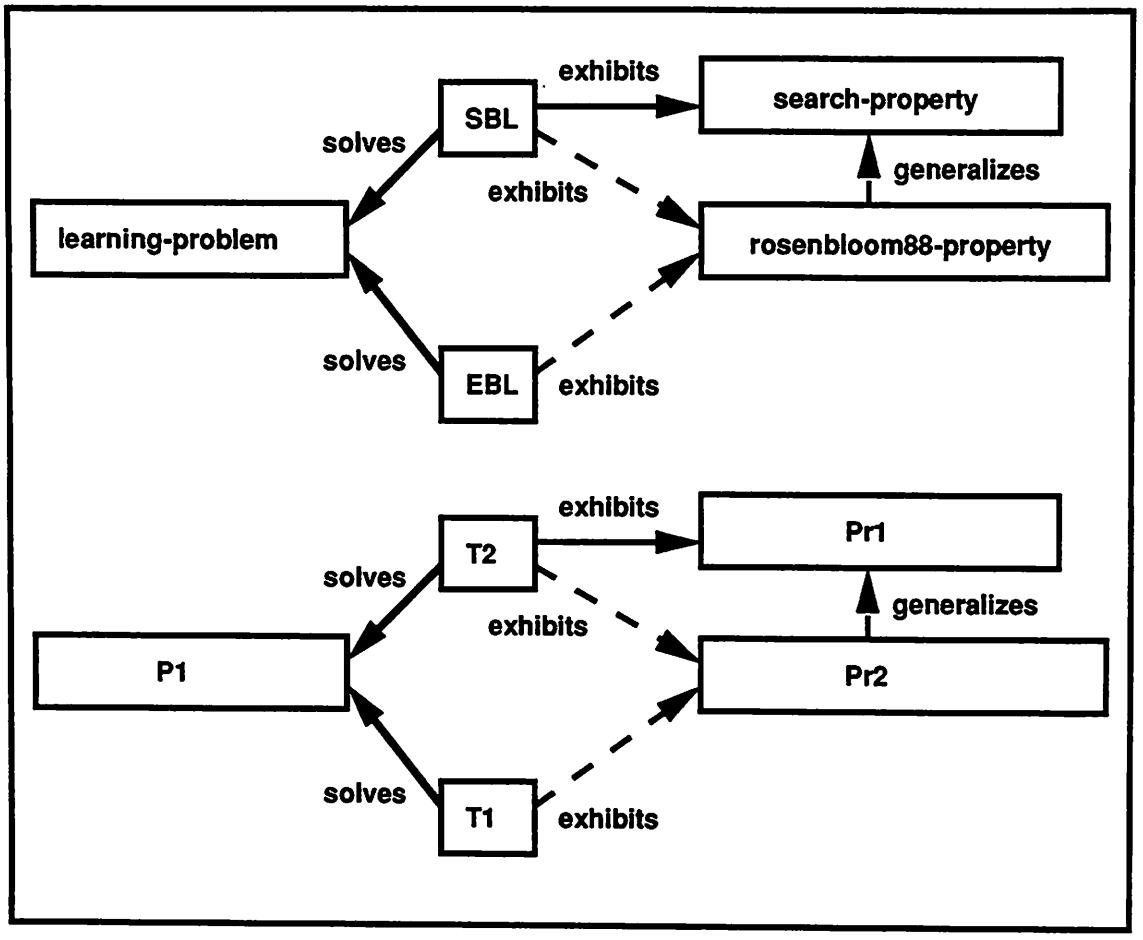


Figure 1.3: Instantiated and Skeletal Schemas of [Rosenbloom 88]

> If there are two techniques T1 and T2 to solve a problem P1, and one of them, T2, has a known property Pr1, then suggest "You could try to generalize Pr1 so that it applies not only to T2 but also to T1."

Thus, RA II infers a paper's **ref** given its **def**. This results in the paper's schema. From this, RA II learns a skeletal schema and hence a research heuristic.

In summary, RA uses a single representational mechanism, namely, research schemas, to support a number of functionalities. Research schemas view research papers as discrete events, each of which adds some new knowledge to a research field, contributing

to its evolution. In addition to their role as a representational mechanism, schemas also act as research heuristics and memory indices. Finally, the same representation supports yet another functionality: it also enables RA II to learn new research schemas. Table 1.2 contains a synopsis of RA.

Was all this so completely natural? It was indeed impossible to explain as the result of someone merely stacking stones in a heap.

—Thor Heyerdahl, The RA Expeditions.

- The **ref** of a paper refers to the knowledge that constitutes the context of the paper.
- The **def** of a paper refers to the new knowledge defined by a paper.
- The *instantiated* research schema of a paper is the combination of its **ref** and **def**.
- The *skeletal* research schema of a paper is its instantiated schema with the constants replaced by typed variables. The skeletal schema represents the structural pattern or research strategy used by the paper.
- The skeletal schemas are used as both heuristic rules for suggesting research directions and as memory indices for retrieving papers that use similar research strategies.
- A memory organized in terms of research schemas supports all of RA's capabilities.
- The schema-acquisition strategy acquires the schema of a paper given only its **def**. The **ref** is inferred as the most specific relations that connect the objects in the **def**.

Table 1.2: Synopsis of RA

1.2 Motivation

This research is motivated by the broad question, "What sorts of knowledge are involved in a researcher's understanding of his field?" An obvious answer to the question is that

a researcher has to have a deep knowledge of the research papers in his field. This knowledge — which I call *deep semantic* knowledge — involves the definitions of the various terms, how they relate to each other, the details of the proposed techniques, why they are important, how they are evaluated and so on[13]. While not denying the existence and use of this kind of knowledge, I consider what other kinds of knowledge might be involved by looking at several tasks that researchers typically perform:

- I show that, in addition to the deep-semantic knowledge, there is also another kind of knowledge — what I have called *structural* knowledge — involved. This knowledge suppresses the details, and models the domain in terms of abstract notions such as *problems, techniques,* etc.

- I show that, in addition to the knowledge of the subject domain, there is also another kind of knowledge involved, a knowledge of the evolution of the subject domain. There is a strong interplay between these two kinds of knowledge.

These two kinds of knowledge are modeled by research schemas: research schemas view the field from a high level of abstraction, and they model each paper as an episode that contributes some knowledge leading to the evolution of the research field.

- Further, this dissertation also considers how a computer program may acquire these research schemas automatically.

The first two claims above are supported by the fact that a theory of research in terms of research schemas has considerable explanatory power: it explains several of the capabilities attributable to researchers. The third point is illustrated with a learning technique to acquire research schemas automatically.

While the domain of research and research literature are interesting in their own right, the scope of this work transcends the domain, and addresses several issues of importance to Artificial Intelligence. These are discussed in the next section.

1.3 The Issues Addressed by RA

RA addresses several issues of central concern to AI. These issues can be grouped under three broad categories: memory, language, and learning. This section gives a brief account of the issues; Chapter 6 provides an extended discussion of related work. Table 1.3 contains a synopsis of this research.

[13]In fact, some of the so-called *knowledge-based* approaches to Information Retrieval attempt to use precisely this kind of knowledge to index and retrieve research papers. See [Lewis et al 89] for a survey.

1.3.1 Memory

For most researchers in AI, the two terms *knowledge-base* and *memory* mean the same thing, though a few may prefer to call semantic nets [Quillian 67] a representational technique for knowledge-bases and MOPs [Schank 82] an organizational technique for memories. The distinction between a knowledge-base and a memory stems from a corresponding distinction between *semantic* and *episodic* knowledge. Several researchers, particularly those concerned with expert systems (see [Barr & Feigenbaum 81] for a discussion of expert systems), have used a static conception of knowledge and have ignored the notion of episodes and how they are related to timeless semantic knowledge. At the other extreme, those concerned with language understanding, particularly story understanding (e.g., [Schank & Abelson 77], [Wilensky 83], [Lehnert 81], [Lebowitz 83]), have focused primarily on episodic knowledge and have treated the role of semantic knowledge, i.e., knowledge not directly pertaining to events, as somewhat secondary.

This dissertation addresses the distinction between semantic and episodic knowledge squarely by focusing on *knowledge evolution*. Knowledge evolution cannot ignore either of these kinds of knowledge — even timeless knowledge accrues through discrete events. The domain of research literature was chosen as a stylistic domain in order to study the nature of knowledge that gradually evolves. Scientific research that purports to reveal the timeless nature of the world can do so in only small discrete steps, with each scientist "standing on the shoulders of a series of giants from the past." Further, a scientist can not only reason about the knowledge of her field, but also about the research trends through which the knowledge of the field accrued. This makes the domain of scientific research an excellent domain from which to address the relationship between semantic and episodic knowledge.

In order to make our discussion easier, I will coin two terms, *S-knowledge* and *E-knowledge*. S-knowledge refers, in general, to semantic knowledge, but in relation to RA, it refers to RA's *subject* knowledge or knowledge pertaining to EBL. E-knowledge refers, in general, to episodic knowledge, but in relation to RA, it refers to RA's *evolutionary* knowledge or knowledge pertaining to the evolution of EBL[14].

RA's memory may be seen as consisting of two conceptually, though not physically, different stores. One store contains RA's S-knowledge, i.e., its knowledge about EBL. This stores may be seen as consisting of nodes and links that stand for subject entities and their relations. For example, when a user asks, "What technique solves the concept learning problem?", RA accesses this store to find the relation (solves EBL concept-learning). Sometimes I will refer to this store as RA's 'knowledge-base' or 'the current world ontology.'

[14] [BIW] The prefixes S- and E- also stand for the first letters of the two Spanish verbs *ser* and *estar*, both of which mean *to be*: *ser* is used to denote a timeless characteristic as in "Yo soy guapo" (I am handsome), whereas *estar* is used to denote a temporally based state of being, as in "Yo estoy bien" (I am fine).

The second store in RA stores the 'evolutionary' or E-knowledge about EBL. This store does not contain explicit EBL entities, but research schemas. Research schemas are configurations of relations that belong to RA's subject knowledge. When a user asks, "What is the relationship between Mitchell-78 and Utgoff-84?," RA accesses this store to find the relationship between the schemas of these two papers. I will refer to this store as 'RA's knowledge about the evolution of EBL.' The term 'RA's memory' is used to refer to the overall memory organization of RA and includes both S-knowledge and E-knowledge.

In summary, the distinction between semantic and episodic knowledge is an important one. RA addresses this distinction squarely by focusing on knowledge evolution, since even timeless semantic knowledge accrues through discrete events. The domain of scientific literature is a very stylistic domain from which to study knowledge evolution.

> *The year 1492 has a magical effect on us all. It was then that Columbus sailed to America. It was then that the world first became round. Before that time, the earth had been flat.*
>
> —Thor Heyerdahl, The RA Expeditions.

1.3.2 Language

The issues addressed by this dissertation under this heading are simply an expansion of the S-knowledge, E-knowledge dichotomy. While this dissertation is not explicitly concerned with language processing per se, it does propose a representation for a large class of expository language texts, namely research papers. This representation should be of considerable interest to researchers in language processing.

Researchers concerned with natural language processing have long recognized the difference between *narrative* and *expository* texts. Narrative texts, such as short stories, describe a set of related events that have 'episodic relations' among them (such as enablement, causality etc.). In contrast, expository texts are characterized by a preponderance of 'semantic relations' such as 'isa' 'has-parts' etc. In general, episodic relations are predictive, semantic relations are not. For example, assume that you are told "John was hungry. He went to Pizza Hut and ordered a pizza." On hearing the first sentence, you could predict that John will execute a plan to satisfy his hunger. You could predict John has the goal of satisfying his hunger. Perhaps you will activate a number of schemas that are reasonable ways to achieve this goal. When told that he went to Pizza Hut, you activate a 'restaurant' schema that provides further predictions (that he will probably eat his pizza), fills in missing details (that he was probably seated by the hostess) etc.

In contrast, suppose you are told "the sea is boiling hot," there is little else you can predict, the least of which is that "pigs have wings[15]." You may wonder why somebody is telling you such a strange thing, and that is an inference about the 'event' of the person telling you something rather than about the sea being hot.

Language processing is hard: processing expository language is considerably harder than processing narrative language. Whatever success AI has had with language has been in the context of narrative processing because narratives can be processed with various kinds of schemas (such as scripts [Schank & Abelson 77], plans [Wilensky 83], story trees [Rumelhart 75] etc.) that provide top-down prediction. In contrast, expository texts are typically processed using unconstrained bottom-up techniques (e.g., [Jacobs 86] [Reimer 87] [Salton 88]).

Without fear of contradiction, I will claim that scientific research papers, such as this dissertation, are expository texts. As much as I would like it to be, this dissertation is not a narrative about events in the world, but an exposition relating different bodies of knowledge to each other. However, research papers are expository only when viewed as isolated pieces of language text. When the research literature in a field is viewed in its totality, one finds that research papers are in fact events with stereotypical (episodic) relations among them. This view of research papers, therefore, provides a schema representation for a class of expository texts in a manner similar to the various schema representations that have been proposed for narrative texts[16].

This representation begs the question, "Can a library of research schemas be used to process research papers predictively?" While this question is not explicitly addressed by this work, a side experiment was undertaken by three of my colleagues to see how this might be done [Lehnert et al 90]. Since research schemas attempt to capture the relationship of a paper to other papers in the field, this experiment focused on reference sentences, i.e., sentences in research papers that explicitly reference other papers. Lehnert et al. propose a notion called *conceptual references*, i.e., the conceptual reason behind references, and provide a taxonomy of conceptual reference types. Based on a small corpus, this work suggests that it might be possible to map reference sentences to

[15]This is just a quaint reference to Lewis Carroll's poem:

> The time has come, the walrus said,
> To talk of many things,
> Of shoes, and ships, and sealing wax,
> Of cabbages and kings,
> And why the sea is boiling hot,
> And whether pigs have wings.

Think about understanding this poem predictively!

[16]Some psychologists have proposed mechanisms for top-down processing of expository texts based on normal expectations about discourse, for example, a reader normally expects a thematic sentence to begin a paragraph (see [Voss & Bisanz 85]). Note that such predictions are still predictions about discourse events rather than predictions based on the content of the text.

their conceptual reference types without recourse to any subject (i.e., EBL) dependent knowledge. The idea of conceptual references is discussed again in Chapter 7.

In summary, the distinction between expository texts and narrative texts can be traced to the distinction between S- and E-knowledge. Processing expository texts is much harder than processing narratives because expository texts do not allow schema-based predictive processing. Research schemas show how one can find schema-like structure in a class of expository texts, namely research papers, by treating each publication as a discrete event contributing to the evolution of the knowledge in the field.

1.3.3 Learning

In RA II, I show that the acquisition of research schemas is accomplished as the natural process of assimilating a paper into the memory. When RA II is given the knowledge defined by a paper (i.e., the **def**), it assumes that this knowledge is not a random set of relations, but is in fact a well-motivated piece of research. To understand this motivation, RA II infers the **ref** of the paper as the most specific knowledge that connects the various objects in the **def**.

On a first glance, RA II's learning strategy is similar to explanation-based learning (EBL) [Mitchell et al 86] [DeJong & Mooney 86]. Like EBL, RA II's assimilation phase constructs an instantiated 'explanation' (the instantiated schema); and like EBL, RA II's generalization phase generalizes its explanation by replacing the constants by variables. A closer look, however, reveals that these two techniques are quite dissimilar. In the EBL approach, an explanation is constructed using general semantic knowledge of the domain, which is typically expressed as (uninstantiated) causal rules, whereas RA II infers the **ref** of a paper by directly accessing instantiated knowledge in memory. Thus RA II's learning may be seen as a memory-based learning technique as opposed to EBL which may be seen as a knowledge-based learning technique.

Secondly, RA II's generalization technique is also quite dissimilar to the EBL generalization technique. In EBL, the constants in the generalization are generalized into variables that reflect the mutual constraints among the various constants in the explanation; these constraints are obtained from the rules that were used to construct the explanation. In contrast, it would appear that there is really no basis for RA II's generalization: for example, given that (solves winstons-technique concept-learning), RA II generalizes this into the skeletal schema (solves T1 P1). Why did RA II choose the types technique and problem? Why not learning-technique and learning-problem? Or anything and anything? What is the basis of this inductive leap?

It turns out that RA's generalizations are justifiable for fairly deep reasons: among the various types in RA's world, the types problem, technique etc are the most differentiated categories: they have two important properties called *associativity* and *discriminability*. Interestingly, these two properties are also just those that characterize the

basic-level categories for natural kind objects [Rosch et al 76]. Chapter 5 analyzes RA II's assimilation and generalization phases and relates them to basic-level theories of categorization.

- S-knowledge refers to RA's subject knowledge, i.e., RA's knowledge of EBL. S-knowledge also stands for *semantic* knowledge.
- E-knowledge refers to RA's evolutionary knowledge, i.e., RA's knowledge of research schemas and how they contribute to the evolution of S-knowledge. E-knowledge also stands for *episodic* knowledge.
- RA proposes a set of structural abstractions to capture the relationship among the various papers in a scientific field.
- RA proposes a memory organization for integrating the subject- and evolutionary-knowledge of a scientific domain. This organization is based on the idea of research schemas.
- This memory organization is supported by the fact that it can explain several capabilities of researchers.
- RA also shows how research schemas are acquired.
- Despite superficial similarity, RA's schema acquisition is different from EBL. RA may be seen as memory-based as opposed to EBL which is knowledge-based.

Table 1.3: Synopsis of This Research

1.4 Scope and Limitations

The main thrust of this dissertation is a memory organization to integrate the subject and evolutionary knowledge pertaining to a scientific field. All issues that would detract from this primary goal were either finessed or ignored. In building the RA program, the capabilities of RA were implemented primarily to illustrate the various aspects of the memory organization: the chronological summarization component illustrates how research schemas reflect the evolutionary knowledge of the domain; the suggestion component illustrates how skeletal research schemas can be interpreted as intentional rules of action; the analogical summarization component illustrates how skeletal research

schemas stand for the underlying strategies of research papers. The implementation of each of these components was halted when it appeared that further implementation would not contribute toward the memory organization in any significant way. Hence this dissertation makes no specific claims about the individual components of RA beyond the fact that RA's memory organization supports them in a natural way. Even though RA constructs summaries and analogies, and uses rule application to generate research suggestions, I do not believe that RA contributes in any way to research in summarization, analogical reasoning, or rule application.

For some readers, perhaps the most glaring omission is the issue of control: how does RA control its suggestions? The simple answer is: it doesn't. The suggestions are generated in an essentially arbitrary order. The reasons for this omission are stated below.

The control issue can be addressed substantially (i.e., at the 'knowledge level' [Newell 82]) in one of two ways: (1) by infusing RA with a sense of the 'goodness' of its suggestions, so that RA attempts to generate only good suggestions (in a manner similar to Lenat's view of 'interestingness' in his AM system [Lenat 76]), or (2) by providing RA with facilities for discourse management so that there is a sense of continuity and connectedness in its dialog with the user. Both of these are significant research topics that would have taken us far away from the task at hand. In Chapter 7, I discuss how one might address the issue of generating only good suggestions and why this might be difficult[17]. Chapter 3 outlines some simple discourse management facilities that could be built for RA.

Given that the control issue is not addressed in any substantial way at the knowledge level, I decided that addressing it at the symbol level would be quite irrelevant and misleading. While issues such as conflict resolution are important issues to address from a technological viewpoint, they were omitted because they were tangential to the goals of this research, and it was felt that RA was unlikely to contribute to these research areas in any significant way.

In summary, this work is concerned with a memory organization to integrate the subject and evolutionary knowledge of a scientific domain. The RA program is simply a straight-forward implementation to illustrate the various aspects of this memory organization. As a reader, you will time and again see me make a strong distinction between knowledge- and symbol-level issues. While the symbol level issues are important in their own right, this work does not claim to address them in any significant way. For some perspectives on the approach taken in this work (and for a discussion on knowledge- vs symbol-level theories), see Chapter 6.

[17]Good suggestions should be distinguished from good research that might result from the suggestion. For example, a piece of research that proves $P = NP$ is a good piece of research, but a suggestion, "Why don't you prove $P = NP$"is not a terribly bright one.

1.5 Reading This Dissertation

I decided to put together some sort of home-made device that would show us our latitude without our needing any special skills or modern instruments.

—Thor Heyerdahl, The RA Expeditions.

This chapter gave a quick overview of the RA system and the issues addressed by this dissertation. The rest of the chapters are described in Table 1.4. Chapters 2, 3, and 4 describe the various aspects of RA. In Chapter 5, I first step back from RA to describe psychological theories of categorization, and then relate RA to this work. In Chapter 6, once again, I step back from RA to develop a framework for classifying representational theories, and then come back to RA to state the contributions of this work within this framework. Chapter 7 concludes the dissertation.

Chapter	Importance	Description
2	Important	Describes RA's memory organization. It also includes some discussions which may be skipped.
3	Heh	Describes what RA does with its memory. Also describes the idea of computer-aided research.
4	Important	Describes RA's learning strategy and provides several examples. Some of the examples may be skipped.
5	Interesting	Analyzes RA's learning strategy and why it works. If familiar with basic-levels, skip section 5.2.
6	Somewhat Important	Provides a framework for representational theories. Describes RA's contributions. Read Section 6.2.
7	Somewhat Important	Conclusion and wrap-up. Tells you where we have been and where to go.

Table 1.4: Organization of This Dissertation

In some ways, this is a very simple dissertation: there are only a couple of major ideas here; in other ways, this is a very complex dissertation: the same ideas keep reappearing in various different guises. For those unfamiliar with the EBL domain, a significant amount of overhead is involved in following the examples. To help you wade through all this, I have followed several conventions throughout.

- **Examples:** As far as possible, I have tried to use the same set of examples to illustrate the various aspects of RA.
- **Chapter Organization:** Each chapter opens by highlighting the interesting points covered in that chapter, and provides a readers guide to help you decide what sections

to read. There are also several tables in each chapter that provide various kinds of synopses.

- **Marked Sections:** Some sections and subsections are marked with an asterisk. These discuss advanced material, not necessarily relevant for understanding the material that follows. These sections may be skipped on a first reading, particularly if you are a novice to AI.

- **Footnotes:** Having read the first chapter, you've probably realized that this dissertation is going to be a footnote heaven. Not all footnotes are meant to be read, at least not on a first reading. To help you decide what to read, most footnotes are marked with a visually prominent sign. Unmarked footnotes are intended to be read: they introduce some important detail that did not fit in the main text. Footnotes marked with the sign [ML] are aimed at machine learning afficianados. These footnotes clarify details pertaining to the machine learning papers that are used as examples. They may be safely skipped without any serious gap in your understanding of this dissertation. Footnotes marked with the sign [Btw] are 'By the way' statements that are tangential to the ideas in the main text. A majority of footnotes in this dissertation are of this kind; insisting on reading them all will get you hopelessly lost, so you may want to skip them on a first reading[18,19].

- **Typographical Conventions:** Most of the text is in Roman font, and RA symbols are in sans-serif font. Thus 'EBL' is short form for the learning technique called 'Explanation-Based Learning' whereas EBL is a symbol in RA. **ref** and **def** are always written in bold letters.

Some other points of general interest. All quotations in this dissertation are drawn from the book *The RA Expeditions* by Thor Heyerdahl. This book chronicles the author's voyage across the Atlantic on a reed boat called RA. In its original conception, RA, the program, was something like a hypertext system for browsing through research literature, and the name was supposed to stand for 'Research Assistant.' Over time RA became something else, and 'Research Advisor' is probably a better name for its current incarnation. Either way, the acronym works, so take your pick!

1.6 Summary

RA is a computer program that performs several tasks in the domain of scientific research. All of its capabilities are supported by a single uniform representation called

[18] [Btw] A fourth categories of footnotes, marked with ⚓, were eliminated from the final version of the manuscript in the interest of appearing sober and scholarly.

[19] [Btw] The disease of excessive use of footnotes is called *footnotitis*.

research schemas. This representation views research papers as research events through which the knowledge in the field evolves. RA can also acquire its research schemas without any extra representational gear.

RA addresses several issues of central concern to Artificial Intelligence, namely, memory, language, and learning. By focusing on the evolution of timeless semantic knowledge through discrete episodes, RA recognizes the roles of both semantic and episodic knowledge in intelligent behavior. The domain of research literature was chosen as a stylistic domain to study knowledge evolution.

The language aspects of this dissertation are closely related to the distinction between semantic and episodic knowledge. RA proposes a schema representation for research papers by conceptualizing them, not as isolated pieces of natural language texts, but as stereotypical events that tend to recur in a research domain.

While superficially similar to EBL, RA's learning strategy is quite dissimilar from EBL in that RA does not construct a causal explanation based on general uninstantiated rules. It directly accesses instantiated knowledge from memory; thus RA's learning strategy may be seen as memory-based, as opposed to EBL which is knowledge-based. RA's generalization strategy is also different from EBL's generalization strategy.

Our first day on board the Ra *was over.*

—Thor Heyerdahl, The RA Expeditions.

Chapter 2

RA's Memory

To build a reed boat, you must have reeds.

—Thor Heyerdahl, The RA Expeditions.

RA's memory for research literature consists of two conceptually, though not physically, distinct stores. One store corresponds to RA's knowledge of EBL (S-knowledge), and the other corresponds to RA's knowledge of the evolution of EBL (E-knowledge). Research papers in EBL are represented in terms of research schemas; research schemas are simply a configuration of S-knowledge relations reflecting the new knowledge added by a paper (**def**) and the existing knowledge that provides its context (**ref**).

The uninstantiated version of a research schema is called its skeletal schema. Skeletal schemas are used not only to index papers with similar research strategies, but also as intentional rules of action to suggest possible research directions. In other words, skeletal schemas are used both as memory indices and research heuristics.

Research fields are not homogeneous entities. While a majority of researchers in a field are concerned with what might be called the 'first-order' problems of the field, a small number are engaged in second and higher order problems. For example, a first-order problem of EBL has to do with developing learning techniques to solve learning problems. In contrast, a second order problem of EBL is to show that all EBL techniques may be viewed as program transformation. To handle this heterogeneity, the representational language of RA defines four kinds of objects: 'problem' and 'technique' represent the first-order objects; 'property' represents higher order objects; 'concept' represents lower order objects. The representational language also defines nine predicates to assert relationships among these four types of objects.

Theories of scientific research typically view science from one of two view-points. The process view, characteristic among the philosophers of science, sees science in terms of the underlying processes: the researchers are simply willing participants with little control over the process level entities. The method view, common mostly among mathematicians

(e.g., Polya), sees science (rather mathematics) in terms of a set of methods that the scientist uses in his trade. The dichotomy between the two views is related to the level of abstraction from which one views science. RA's level of abstraction is probably a happy medium between the process view and the method view, resulting in the use of research schemas both as a representation and as heuristic rules. Table 2.1 contains a guide to Chapter 2.

Section	On first reading	Description
1	skim	Describes memory organization. The EBL details are not important.
2	read	Describes the representational primitives. This is interleaved with a discussion of their choice.
3	skip	Discussions on assorted topics. Some readers may want to skim 2.3.1 and 2.3.3.
4	read	Summary.

Table 2.1: Guide to Chapter 2

2.1 RA's Memory

The combination of RA's subject knowledge (S-knowledge) and evolutionary knowledge (E-knowledge) is referred to as 'RA's memory.' This memory reflects RA's knowledge of EBL, as well as RA's knowledge of the evolution of EBL. In a narrow sense, this thesis is concerned with how one could organize both aspects in a single memory organization. In a broader sense, this thesis is concerned with how timeless semantic knowledge and temporally based episodic knowledge might be organized in memory.

RA's S-knowledge may be seen as a semantic net that states various relations among EBL entities — e.g., EBL solves the learning-problem; EBL entails intractable-theory-problem and so on. EBL entities such as 'problems' and 'techniques' are represented as nodes, and relations such as 'solves' and 'entails' are represented as links[1].

RA's E-knowledge is represented in terms of research schemas. All research schemas have two sets of relations, **ref** and **def**. **ref** contains the S-knowledge relations that constitute the context for a given paper, and **def** contains the new EBL relations asserted by that paper. The idea of research schemas is based on the claim that standard **ref**, **def** configurations recur in science.

[1]The next section lists the representational primitives and the notational conventions used. You should be able to follow this section based on what was introduced in Chapter 1. If you have some difficulty, you may want to refer to the next section occasionally.

To show RA's memory organization, the following strategy will be used. I will show how the memory evolves as a series of example papers from the early phase of EBL are added to the memory. The examples have been chosen to illustrate the memory, the instantiated and the skeletal versions of the schemas, and the indexing mechanism. Since the papers are from the early phase of the paradigm, their research strategies (i.e., the 'if-then' nature of the research schemas) are not very compelling; in Chapters 3 and 4, we will see better examples of research schemas used as research strategies.

A major difficulty arises in describing any evolving thing — "But where did it all start?" Without explanation or apology, let's assume that, on the eighth day, God created 'problem'. Further, assume that God asserted a number of relations between problem and several other objects and wrote a paper about it. This paper, [God 08], had the following schema:

ref: ϕ

def: {(dominates problem performance-problem)
(dominates problem learning-problem)
(dominates performance-problem language-understanding-problem)
(dominates performance-problem equation-solving-problem)
(dominates learning-problem concept-learning-problem)
(dominates learning-problem control-learning-problem)}

This is the initial paper given to RA. Let's look at a couple of nodes created by RA when [God 08] is assimilated. Figures 2.1 and 2.2 show two frames, the frames corresponding to problem and learning-problem. The former contains a slot called dominates that has the latter as its value. In reverse, the learning-problem frame contains a slot called dominated-by that has problem as its value. problem, learning-problem, and dominates are EBL items and constitute RA's S-knowledge[2].

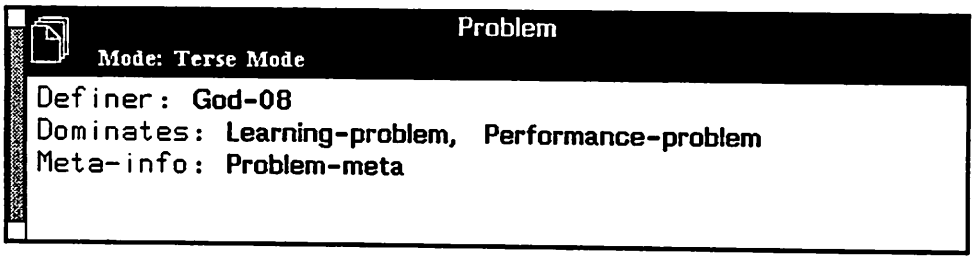


Figure 2.1: Frame for Node Problem

[2]There are some other slots shown in these frames. These slots and values are used for several book-keeping purposes as well as for the convenience of the hypertext system. They do not partake in RA's schema representation.

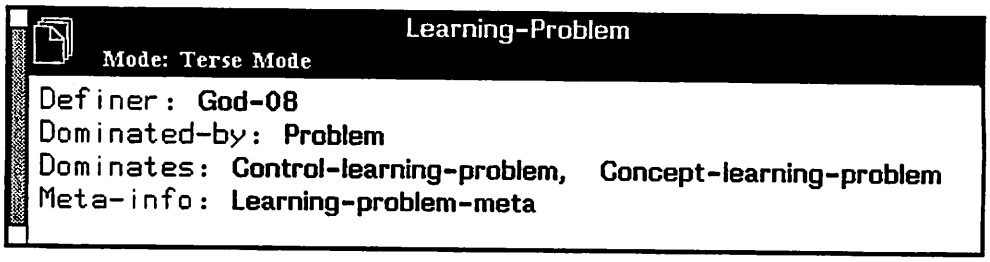


Figure 2.2: Frame for Node Learning-problem

The information about the paper, [God 08], is shown in Figures 2.3 and 2.4. The first frame (Figure 2.3) contains fields such as the author's name, title, year and place of publication. This frame also contains a pointer to the paper's schema, called God-08-schema. Figure 2.4 depicts God-08-schema. Normally, all schema frames contain two slots ref and def[3]. God-08-schema does not have a ref slot because it is null. The def slot contains several values with the prefix 'mvalue-frame.' These are simply pointers to the S-knowledge relations such as '(dominates problem learning-problem)'. To visualize this, refer to Figure 2.5. problem, learning-problem and performance-problem are nodes in memory. dominates is a labelled link, i.e., a pointer, from problem to learning-problem. When I say that the **ref** and **def** of research schemas contain S-knowledge relations, I mean that they contain pointers to these relations, which are themselves pointers.

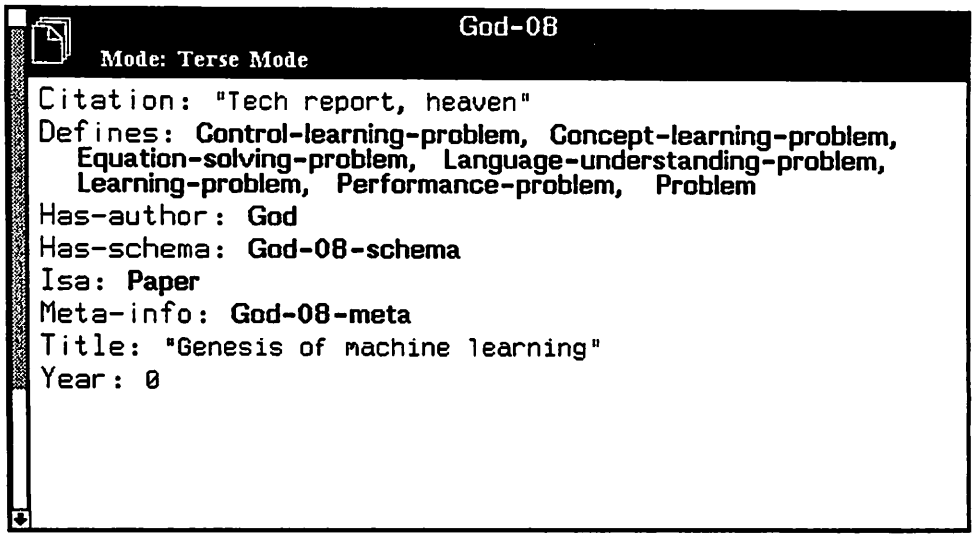


Figure 2.3: Frame for Paper God-08

[3]Note that the bold letter versions, **ref** and **def** are conceptual objects; the sans-serif versions ref and def are symbols in RA that happen to stand for ref and def.

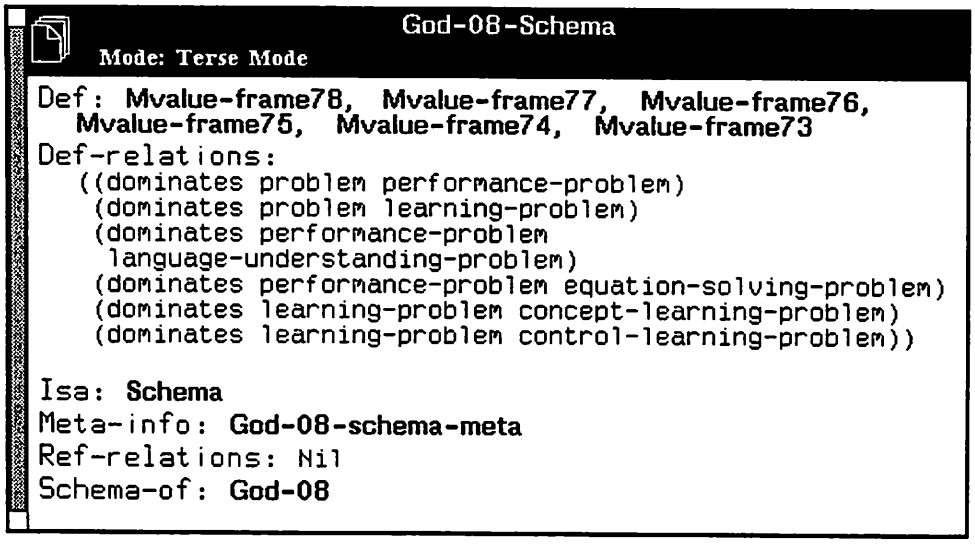


Figure 2.4: Frame for Schema, God-08-schema

For expository purposes, I will show all schema frames as containing two additional slots, called ref-relations and def-relations. These two slots are not used by the system but they help you pretend that the **ref** and **def** relations are physically stored in the schema as Lisp expressions. It does not matter at a conceptual level, so assume that the slots ref-relations and def-relations stand for **ref** and **def**. Ignore the slots named ref and def.

Several ice ages later (after this eventful eighth day), in May 1979, Gerry DeJong completed his doctoral dissertation at Yale University. This document, [DeJong 79], had the following schema[4]:

ref: {(dominates performance-problem language-understanding-problem)}

def: {(instantiates language-understanding-problem story-understanding-problem)
(solves schema-based-technique story-understanding-problem)}

[4] [ML] The basic idea of EBL was perhaps simultaneously developed by Gerald DeJong and his colleagues in Illinois, Thomas Mitchell and his colleagues in Rutgers, and Bernard Silver in England. Before the emergence of EBL as a paradigm, there were at least three other attempts at using domain knowledge for learning. The earliest attempt was perhaps by Waterman in his poker playing program [Waterman 70]. The second attempt was in the STRIPS system for robot planning [Fikes et al 72]. The third attempt was by Eliot Soloway in his Baseball program [Soloway 78]. None of these quite caught on; EBL co-alesced into a paradigm when three separate research groups 'discovered' it simultaneously approaching from three very different angles: Utgoff (at Mitchell's group) arrived at EBL as a way of learning a new representation for inductive learning systems; DeJong arrived at it as a way of learning schemata for natural-language processing; Silver arrived at it as a way of learning control knowledge. All this happened at around the same time (IJCAI 83), and the EBL paradigm was born. Here, I consider the work of DeJong and Silver.

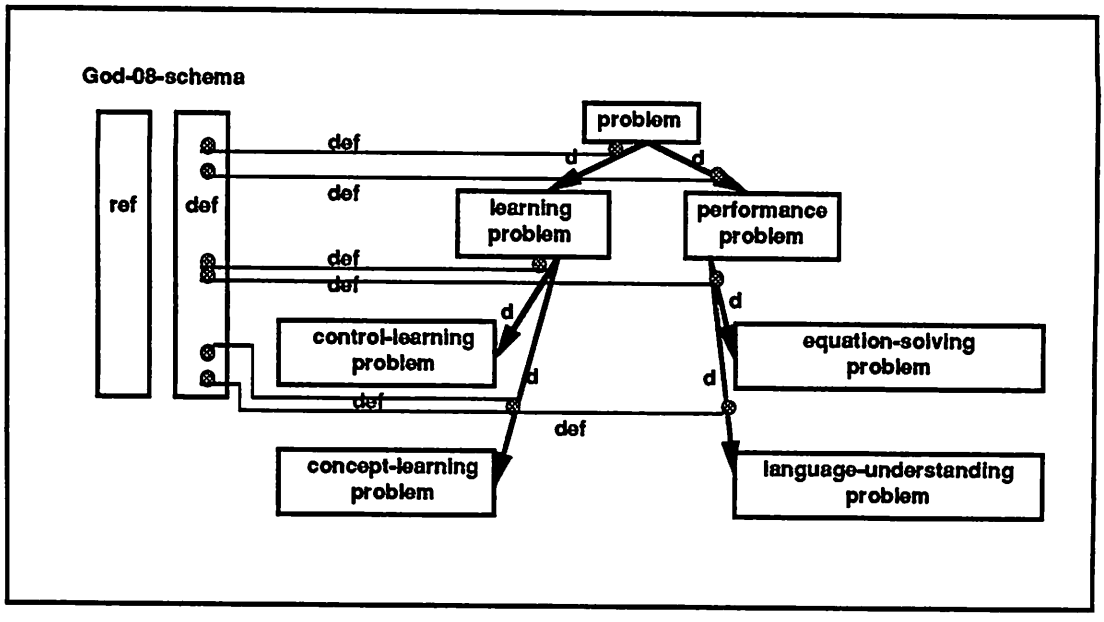


Figure 2.5: Connection Between S-knowledge and E-knowledge

A few years later, in 1985, DeJong and his student Ray Mooney presented a paper, [DeJong & Mooney 85], which had the following schema:

ref: {(solves schema-based-technique story-understanding-problem)
(dominates learning-problem concept-learning-problem)}

def: {(acq[5] schema-based-technique schema-acquisition-problem)
(instantiates concept-learning-problem schema-acquisition-problem)
(solves explanatory-schema-acquisition schema-acquisition-problem)}

These schemas correspond to the following abstracts:

> [DeJong 79]: One kind of performance problem is the language understanding problem [God 08]. In this paper, I propose that story understanding problem is an instance of the language understanding problem. I propose a technique called schema-based-technique to solve the story understanding problem.

[5]The 'acq' relation connects a non-learning technique, such as schema-based-technique, to a learning-problem, such as schema-acquisition-problem, that arises out of that technique. See Section 2.2.2 for a full description of the 'acq' relation.

[DeJong & Mooney 85]: Schema-based techniques have been shown to solve the problem of story-understanding [DeJong 79]. In this paper, we propose the problem of learning schemas automatically. Let's call it the schema-acquisition-problem. Explanatory-schema-acquisition is a technique that solves the schema-acquisition-problem.

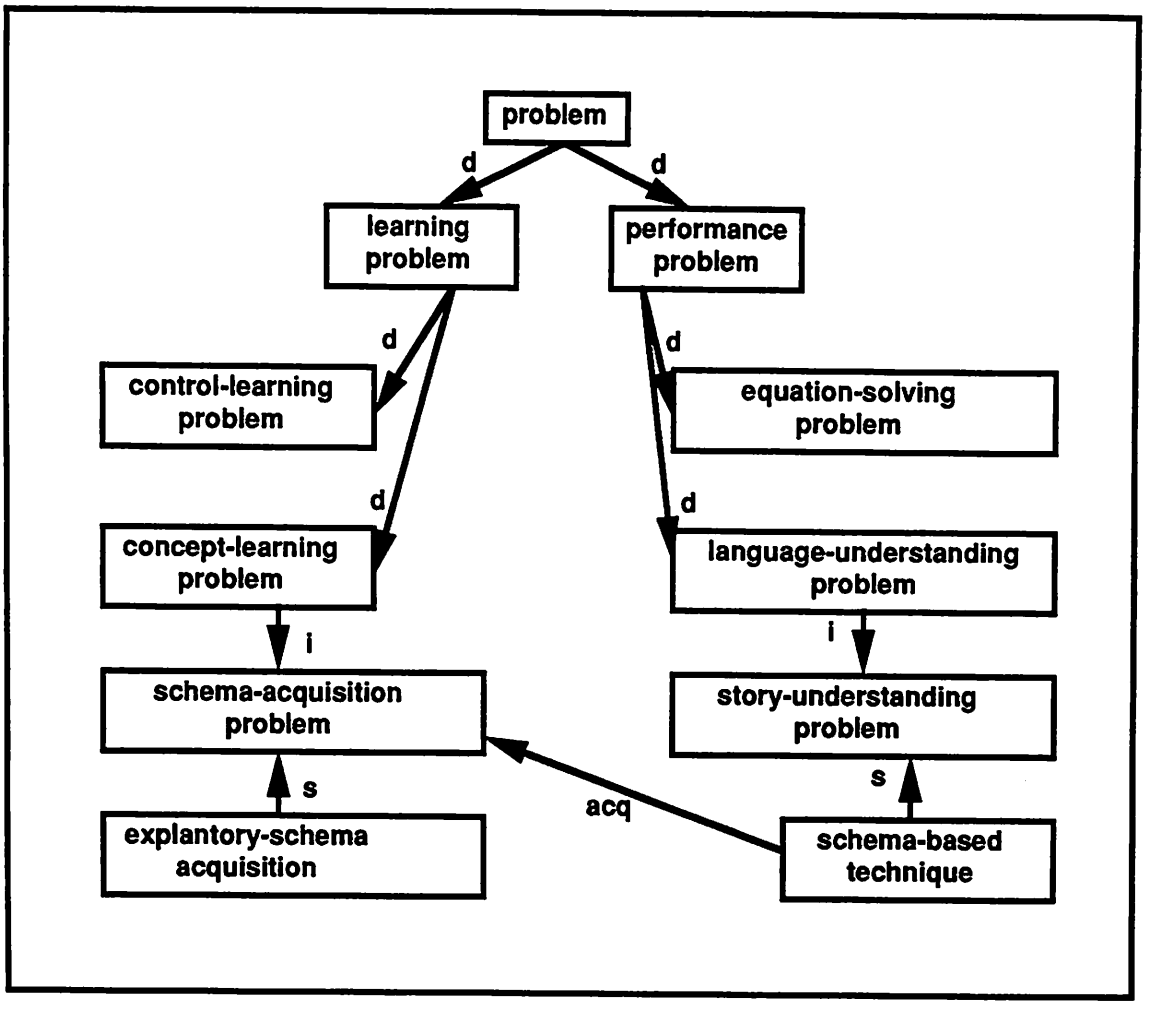


Figure 2.6: S-knowledge After [DeJong 79] and [DeJong & Mooney 85]

Figure 2.6 shows the S-knowledge of the field as it stands after the assimilation of DeJong-79 and DeJong-85. Figures 2.7 and 2.8 show the schemas of DeJong-79 and DeJong-85. They contain the slots ref-relations and def-relations that constitute the *instantiated* versions of their schemas. Before we see where the skeletal versions of these schemas are stored, let's look at another parallel development of EBL.

Across the Atlantic from New Haven, in England, Alan Bundy and his colleagues had developed a system called PRESS to solve algebraic equations (symbolically). The paper describing this work, [Bundy et al 81], had the following schema:

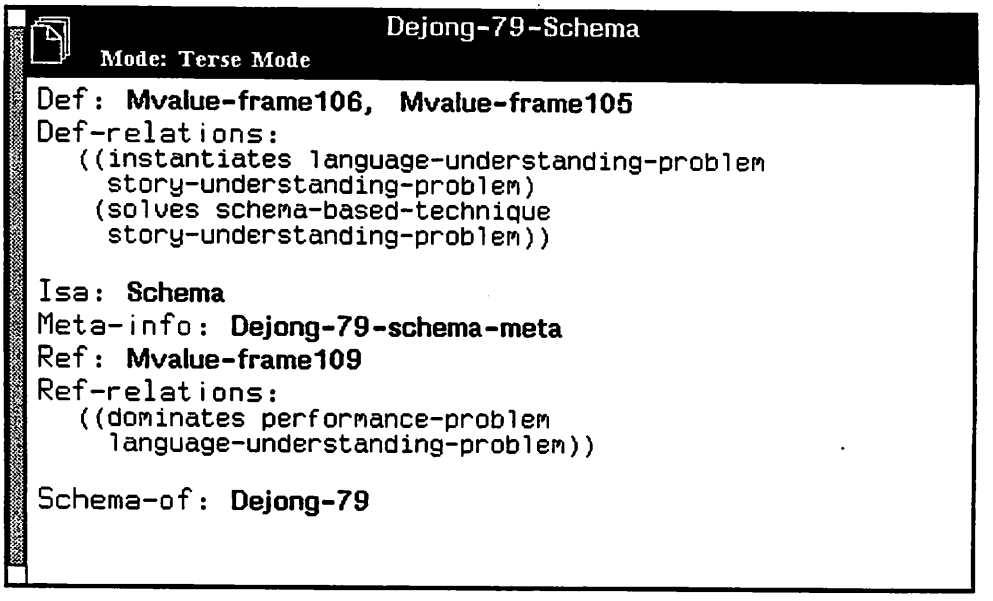


Figure 2.7: Frame for Schema, DeJong-79-schema

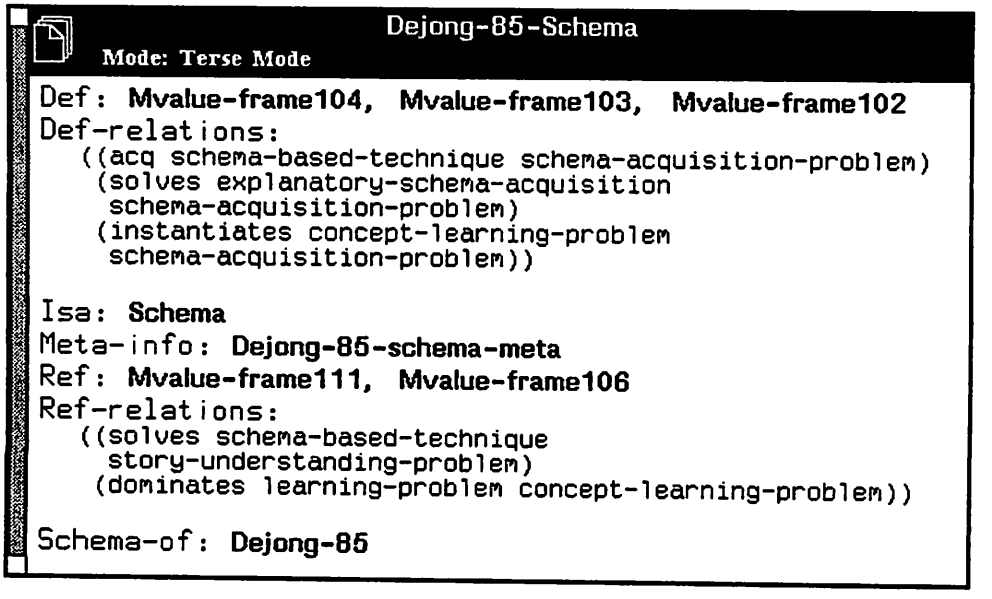


Figure 2.8: Frame for Schema DeJong-85-schema

ref: {(dominates performance-problem equation-solving-problem)}

def: {(instantiates press-problem equation-solving-problem)
(solves press-technique press-problem)}

After the work on PRESS, Bernard Silver, a student of Bundy, considered how one could automatically learn the 'control' knowledge in solving equations. He developed a system, LP, for this purpose. Silver's paper, [Silver 83], had the following schema:

ref: {(solves press-technique press-problem)
(dominates learning-problem control-learning-problem)}

def: {(acq press-technique LP-control-learning-problem)
(instantiates control-learning-problem LP-control-learning-problem)
(solves LP-learning-technique LP-control-learning-problem)}

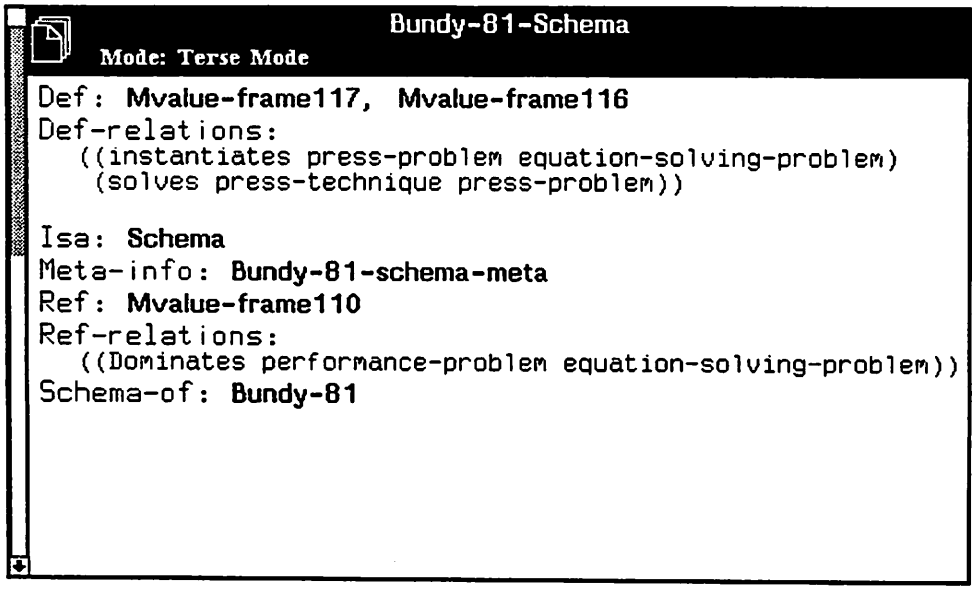


Figure 2.9: Frame for Schema, Bundy-81

The schemas of [Bundy et al 81] and [Silver 83] are shown in Figures 2.9 and 2.10. These papers correspond to the following abstracts:

> [Bundy et al 81]: One kind of performance problem is the equation solving problem [God 08]. In this paper, we propose that PRESS-problem is an instance of the equation solving problem. we propose a technique called PRESS-technique to solve the PRESS-problem.

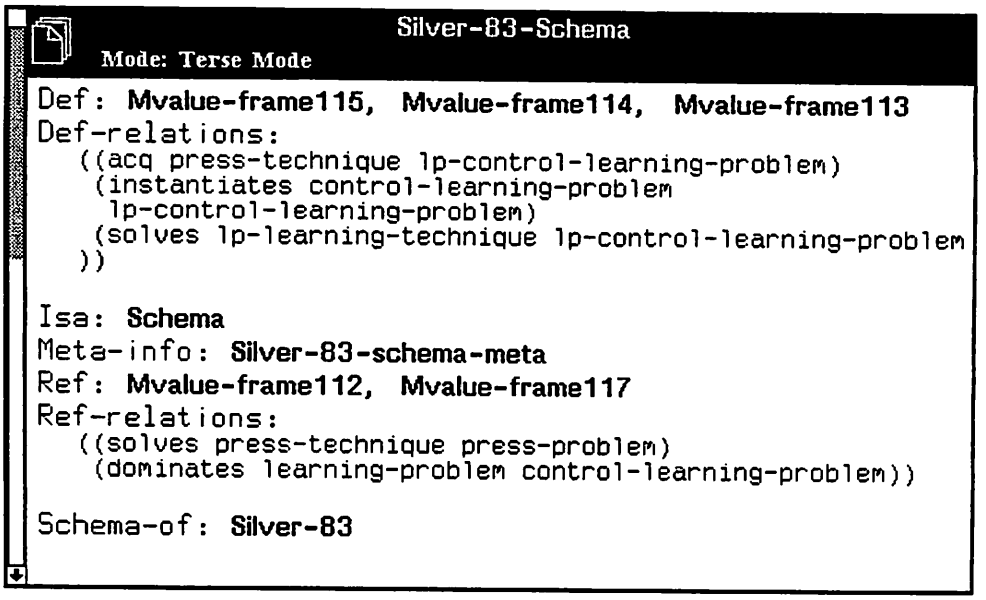


Figure 2.10: Frame for Schema Silver-83

> [Silver 83]: PRESS-technique has been shown to solve the PRESS-problem [Bundy et al 81]. In this paper, I propose the problem of learning control knowledge for PRESS-technique automatically. Let's call it the LP-control-learning-problem. LP-learning-technique is a technique that solves the LP-control-learning-problem.

Compare the the abstract of [Bundy et al 81] with that of [DeJong 79] (page 28) and the abstract of [Silver 83] with that of [DeJong & Mooney 85] (page 29). The similarity is due to the fact [Bundy et al 81] and [DeJong 79] have the same skeletal schema shown below:

ref: {(dominates P1 P2)}

def: {(instantiates P2 P3)
(solves T1 P3)}

Also, [Silver 83] and [DeJong & Mooney 85] have the same skeletal schema shown below:

ref: {(solves T1 P3)
(dominates P1 P2)}

def: {(acq T1 P4)
(instantiates P2 P4)
(solves T2 P4)}

As we saw in Chapter 1, skeletal schemas act as indices for papers. Hence papers with identical schemas should have the same index. Figures 2.11 and 2.12 show the indices for the two sets of papers above. The frame Index-001 indexes '[Bundy et al 81]' and '[DeJong 79]', and the frame Index-002 indexes '[Silver 83]' and '[DeJong & Mooney 85].' The slot 'Index' specifies the index (i.e., the skeletal schema as two sets of relations) and the slot 'Indexes' points to the papers that this index indexes.

Index frames also contain two slots called If and Ifthen[6]. The If slot contains the relations in the **ref** of the skeletal schema and Ifthen slot contains the relations in the **def** of the skeletal schema. These two slots are interpreted as a research heuristic by RA to suggest new research directions. The component of RA that suggests research directions accesses the index frames, interpreting indices as suggestion heuristics. When a heuristic is applicable, RA generates a suggestion and retrieves the papers indexed on that index frame as examples of where that research heuristic has been used before[7,8]. The suggestion and the retrieval component of RA are described in Chapter 3.

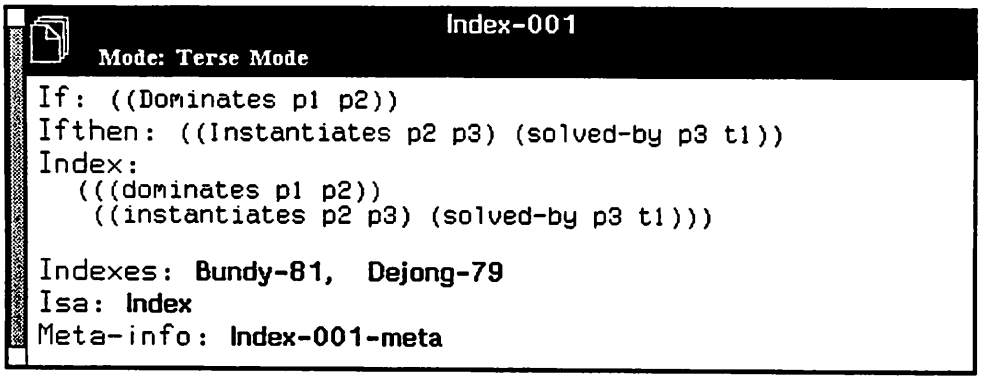


Figure 2.11: Frame for Index Index-001

Before closing this section, I would like to show one more example. In 1986, Mitchell et al. published a paper in the *Machine Learning* journal putting all the various work on EBL within a single framework. This paper is represented in RA as follows:

[6] Btw Why 'Ifthen' and not just 'Then'? The underlying knowledge representation system that displays the frames sorts the slots in alphabetical order. With Ifthen, the two slots are kept adjacent to each other.

[7] In Chapter 1, we saw that a heuristic is applicable if its 'if' conditions are satisfied in the neighborhood of the 'anchor' node. An anchor node is the node in which a user is 'interested' and wants some suggestions for research.

[8] Btw The heuristics embodied by the index frames Index-001 and Index-002 are not the greatest ones partly because they are derived from papers too early in the paradigm, and partly because the relation 'acq' requires you to distinguish between learning problems and performance problems. Hence our strategy of replacing constants by typed variables, e.g., replacing press-technique by T1 results in over-generalization with respect to 'acq.' Because of this, 'acq' had to be dropped from RA II. See Chapter 5.

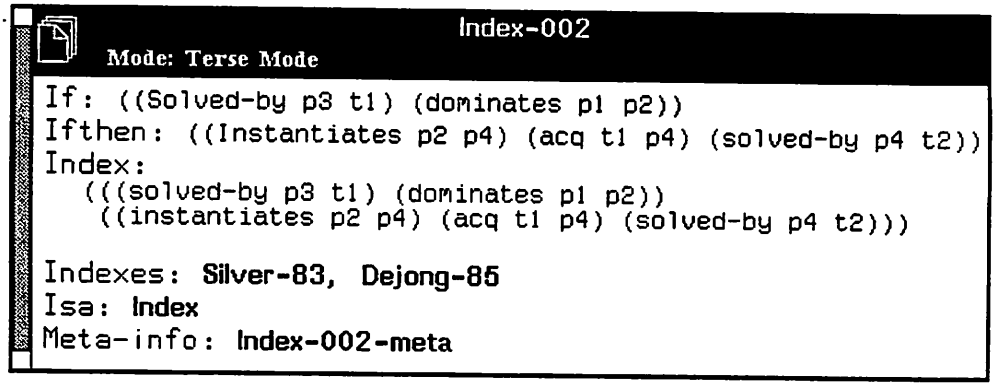


Figure 2.12: Frame for Index Index-002

ref: {(dominates learning-problem concept-learning-problem)
(dominates learning-problem control-learning-problem)
(instantiates concept-learning-problem schema-acquisition-problem)
(instantiates control-learning-problem LP-control-learning-problem)
(solves explanatory-schema-acquisition schema-acquisition-problem)
(solves LP-learning-technique LP-control-learning-problem)}

def: {(solves EBL learning-problem)
(instantiates EBL explanatory-schema-acquisition)
(instantiates EBL LP-learning-technique)}

This schema corresponds to the following abstract:

> [Mitchell et al 86]: Learning problem consists of two kinds, concept-learning-problem and control-learning-problem [God 08]. Schema-acquisition-problem is an instance of concept-learning-problem and is solved by a technique called explanatory-schema-acquisition [DeJong & Mooney 85]. LP-control-learning-problem is an instance of control-learning-problem and is solved by a technique called LP-learning-technique [Silver 83]. In this paper we propose: EBL is a general technique for solving the learning-problem. Explanatory-schema-acquisition and LP-learning-technique may be seen as instances of EBL.

This abstract corresponds to the following skeletal schema:

ref: {(dominates P1 P2)
(dominates P1 P3)
(instantiates P2 P4)
(instantiates P3 P5)
(solves T1 P4)
(solves T2 P5)}

def: {(solves T3 P1)
(instantiates T3 T1)
(instantiates T3 T2)}

This skeletal schema is shown in Figure 2.13. It may be seen as the following research heuristic:

> If there is a problem P1 that consists of two kinds of problems P2 and P3 whose instances P4 and P5 are solved using techniques T1 and T2, then suggest, "You could provide a general technique T3 to solve P1, and show that T1 and T2 are instances of T3."

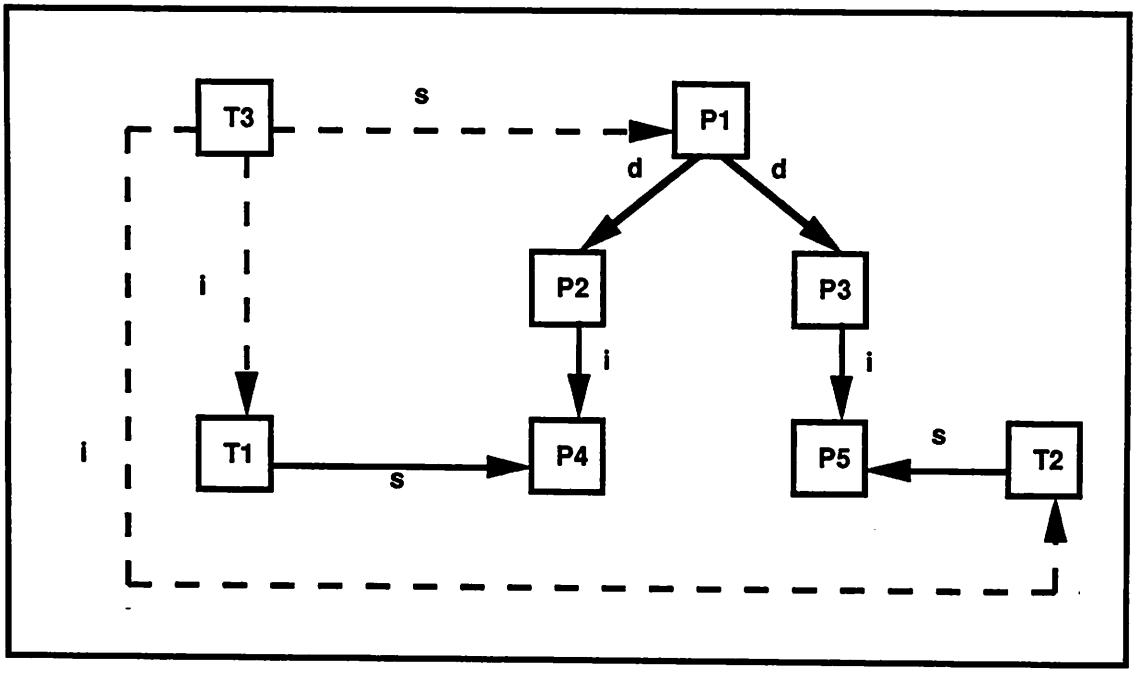


Figure 2.13: Skeletal Schema of [Mitchell et al 86]

It is now time for some disclaimers about my use of EBL. The EBL details here and in the rest of the dissertation are not grossly in violation of reality, but there is some misrepresentation in how I interpret some papers and how I use the historical record. For example, [Mitchell et al 86] may be captured in terms of a number of different schemas, and I chose a convenient one here. The paper [DeJong & Mooney 85] is not the first paper by DeJong on explanatory schema acquisition (the first one is perhaps the one published in IJCAI 81, see reference [DeJong 81]). [DeJong & Mooney 85] served my purposes better, so I used it. In general, RA is not the time capsule of EBL, but may be seen as 'EBL-motivated'. Therefore I will use EBL for concrete examples when

convenient, and twist EBL to suit my convenience when needed. I do this without a great deal of apology or remorse because the idea here is to describe knowledge evolution and the RA program, not to build an accurate model of EBL.

In summary, RA's memory may be seen as consisting of (1) S-knowledge entities organized in a semantic net, and (2) research schemas that are structured sets of pointers to the S-knowledge relations. Papers which have identical skeletal schemas are indexed using the same index-frame; the index frame also contains the 'if-then' version of the skeletal schema which is used as a research heuristic. Through concrete examples, I showed how the memory grows as a series of papers are added to it. Table 2.2 contains a synopsis of RA's memory organization.

- RA's subject knowledge, i.e., knowledge of EBL, is called S-knowledge.
- S-knowledge is represented in terms of objects and relations.
- S-knowledge relations state the relationship between two S-knowledge objects, such as (solves EBL learning-problem).
- RA's evolutionary knowledge, i.e., knowledge of the evolution of EBL, is called E-knowledge.
- E-knowledge is represented in terms of instantiated research schemas.
- Instantiated research schemas contain two sets of pointers, **ref** and **def**. The pointers in **ref** point to S-knowledge relations that a paper references, and pointers in **def** point to the S-knowledge relations that a paper defines.
- Uninstantiated or skeletal research schemas are structural patterns of research. These are used as memory indices. A memory index contains pointers to all instantiated schemas that use the same structural pattern (or strategy) of research.
- Skeletal schemas are also used as if-then heuristics where the **ref** is used as a condition and the **def** is used as a suggestion.

Table 2.2: Synopsis of RA's Memory Organization

2.2 Representational Primitives

The RA system represents research papers from the field of Explanation-Based Learning (EBL). RA's representation has two significant features, itemized below:

- Papers are viewed from a high level of abstraction to capture the structural relations among problems, techniques etc.
- Papers are viewed as research events through which the knowledge of the field evolves.

The previous section was concerned with the representation of papers in terms of research episodes (i.e., as research schemas) through which RA's memory evolves. This section is concerned with the S-knowledge primitives from which the research schemas are built.

2.2.1 Primitives

From an implementation point of view, RA's representation scheme may be seen as a frame-based language with the usual sorts of slots and links[9]. In addition to the primitives, RA has a number of other ad-hoc types of objects, strange relations with ill-specified semantics etc. — the kinds of things you need to get an AI system to work. Fortunately, most of these 'extraneous' beasts are not part of RA's representation for research schemas. In this section, I will describe only those primitives that will be used in schemas. The other objects will be introduced when the need arises.

The representational language defines four types for objects. Objects — also called nodes, entities, EBL entities or subject entities — are represented as nodes in a network, and are shown as symbols in sans-serif font. Graphically, objects are depicted by a symbol, enclosed in a box. The 'type' of an object should be obvious from its name and the context in which it appears. Itemized below are the four types of objects and their variable prefixes (the prefixes are used to depict typed variables in skeletal schemas and heuristics):

1. Problem, depicted as P.
2. Technique, depicted as T.
3. Concept, depicted as C.
4. Property, depicted as Pr.

[9] Btw RA is constructed on top of a generic 'Frame System' (FS) built by Mike Greenberg of the Experimental Knowledge Systems Laboratory at the University of Massachusetts.

Before getting to their 'semantics,' let me itemize the set of relations that are defined for these objects. Relations — also called links, predicates, or assertions — are written as Lisp expressions in running text, as in (dominates learning-problem concept-learning-problem). Graphically, relations are depicted as labelled arrows between nodes. Following normal AI conventions, I use the term relation in two different ways: sometimes the term refers only to the predicate symbol, e.g., 'dominates'. Sometimes it refers to an entire assertion, e.g., (dominates learning-problem concept-learning-problem). This is usually very clear from context.

In all, nine relations are defined for the four types of objects above. The relations and their graphical notation are shown below:

1. Dominates, depicted as $\xrightarrow{d}$.
2. Instantiates, depicted as $\xrightarrow{i}$.
3. Encapsulates, depicted as $\xrightarrow{n}$.
4. Solves, depicted as $\xrightarrow{s}$.
5. Entails, depicted as $\xrightarrow{e}$.
6. Exhibits, depicted as $\xrightarrow{x}$.
7. Involves, depicted as $\xrightarrow{v}$.
8. Acq, depicted as $\xrightarrow{acq}$.
9. R, depicted as $\xrightarrow{R}$.

In the next subsection, I describe the semantics of these objects and relations. The notion of 'semantics' is somewhat complicated in RA's case and my use of the term is clarified in Section 2.3.1.

2.2.2 Semantics of the Primitives

Research schemas are constructed from 4 types of objects and 9 relations. This section describes what these objects and relations are supposed to model, and illustrates them with examples. For ease of understanding, the graphical notation introduced in the previous section will be used throughout.

2.2.2.1 Semantics of Objects

In RA's terms, a problem is anything that can be solved, and a technique is something that solves a problem. While the two terms 'problem' and 'technique' suggest an engineering domain, there is nothing essentially engineering about the theory of research being proposed here. These two objects simply recognize that research is a purposeful activity: 'problem' denotes the object of inquiry and 'technique' denotes the result of the inquiry. With respect to different fields of research, these objects may take on different meanings. For a mathematical domain, 'problem' may correspond to a theorem, and 'technique' may correspond to a proof. For a psychological domain, 'problem' may correspond to a hypothesis and 'technique' may correspond to an experiment to verify the hypothesis. So long as the use of these terms is consistent within a domain, I believe that these two notions capture the purposive nature of a research field. However, this belief remains to be validated (see Chapter 7).

A more interesting question with the notions of problem and technique is not whether they are general across different research disciplines, but whether they can characterize a large domain consistently. Given a single paper to transcribe, one can always decide what is the problem being solved and what is the solution being proposed. As you expand from that paper to transcribe an entire field, you notice that your notions of problem and technique have undergone a 'concept drift,' i.e., what you meant by a problem in some neighborhood of the research domain is somewhat different from what you mean by the same term in another neighborhood[10]. Let's consider some examples: Winston's work on learning [Winston 71] may be seen as proposing a technique to solve the problem of learning the structural description of 'arches' (as opposed to cantilever beams); it may also be seen as proposing a technique to solve the problem of concept-learning. Samuel's work on learning [Samuel 59] may be seen proposing a technique to solve the problem of learning to play checkers; it may also be seen as proposing techniques to solve the credit assignment problem. Utgoff's work on bias adjustment showed how his STABB system [Utgoff 84], in the context of learning heuristics for solving symbolic integration problems, learns the concept 'odd number' automatically. Thus the problem being solved may be called the 'odd-number-learning-problem.' A more generic notion of this problem is the 'bias-adjustment-problem.' An even more generic notion of this problem is the 'new-term-learning-problem.' As you look at all papers from a domain, the problems addressed by different papers can be characterized at different levels of genericity.

The simple solution adopted in RA was as follows: A paper was represented at the level of genericity at which other papers referred to the work. For example, Winston's work was represented as proposing a technique to solve the concept learning problem,

[10] BIW See Cynthia Loiselle's forthcoming dissertation on 'ontology maintenance,' i.e., techniques to identify and possibly correct concept drift in large knowledge bases. For a brief overview of this approach, See [Loiselle & Cohen 89].

because most other papers (for e.g., [Vere 75] and [Mitchell 78]) that refer to this work see it as solving the concept-learning problem, not as solving the arch-learning problem[11]. Samuel's paper [Samuel 59] is represented as solving the 'checker-learning-problem' because the only other paper in RA that refers to this paper is [Samuel 67] which is best seen as solving the checker-learning-problem by extending the original technique by a new idea called signature tables. Finally, Utgoff's work [Utgoff 84] is represented as solving the problem of 'fixed-representational-bias' since this is how the field as a whole characterizes this work. Hence the notion of 'problem' refers to problems at different levels of genericity in different neighborhoods of the knowledge base. This suggests an important problem for future research discussed in Chapters 5 and 7.

There is another dimension to the 'concept drift' problem as it applies to research domains. This has to do with the fact that research domains are not homogeneous entities. This means that, within the boundaries of the discipline, some researchers address what might be called the 'first order' problems and others address 'second and higher order' problems. For example, the first order problem in EBL has to do with designing learning techniques to solve various kinds of learning problems. While this might be the major research mode of most of the EBL researchers, a small number of researchers engage in second order problems such as proving that EBL is same as program transformation, EBL will not work, EBL will work, and so on. It is hard to capture, for instance, Searle's paper on the Chinese room argument [Searle 80] and a paper that proposes a faster version of the constraint backpropagation algorithm, in the same representation without a change in your notion of problems and techniques. The two other types of objects in RA, 'Property' and 'Concept' are defined as a simple way of relating the first order objects, i.e., problems and techniques, to higher and lower order objects.

To understand the motivation behind the notion of 'property,' let us consider an example. Let's assume that our focus of interest is the domain of algorithm design. Let's say that, in this domain, problems correspond to specific problems for which we seek computational methods or algorithms. Hence, 'Traveling-salesman-problem' is a problem and 'Dijkstra's-algorithm' is a technique to solve that problem. In the same vein, 'the halting problem of the Turing machine' is a legal problem for which we can seek algorithms, and any algorithm that exists is a technique to solve 'the halting problem of the Turing machine.' However, at this level of representation, the fact that 'the halting problem of the Turing machine is undecidable' will be represented as an uninterpreted assertion, i.e., a property, pertaining to the 'halting problem of the Turing machine.'

[11]Hypothetically, let's assume that architects had gotten interested in this work, and had written papers such as the following: "Winston's technique solves the problem of learning the structural description of arches with *two* standing supports [Winston 71]. In this paper, I generalize Winston's technique to solve the problem of learning the structural description of arches with any number of standing supports." In this case, Winston's work would be represented as proposing a technique to solve the 'arch-learning' problem.

This is not to say that our representation cannot capture theoretical disciplines. If our focus of interest — i.e., our first order concern — is the theory of computability, then a problem in this domain would be 'to prove that the halting problem is undecidable,' and any proof for this statement would be a technique to solve the problem.

While the notion 'property' is used to denote second order objects of a domain, the notion 'concept' stands for zeroth order objects. For example, 'to prove that the halting problem of the Turing machine is undecidable' is a first order problem in theory of computability. How is this problem related to 'the halting problem of the Turing machine?' We will say that the latter is a 'concept' that is involved in the former. In general, a concept is any object that is not a direct object or result of inquiry in a piece of research but is part of the problem or technique being investigated.

Table 2.3 contains a summary of what RA's objects are supposed to model. In my effort to concentrate on the memory organization and the integration of S-knowledge and E-knowledge, I have attempted to keep the set of representational primitives minimal. 'Property' and 'Concept' are rather simple-minded (zeroth order?) solutions to handle the multiple levels of abstractions involved in conceptualizing any complex piece of knowledge. As a representation for scientific knowledge, RA's vocabulary is rather impoverished. See the discussion on 'content theories' (Sections 6.1.2 and 6.4.1) for some perspective on why RA's vocabulary was kept minimal. See Chapter 7 on suggestions for future work. Table 2.3 lists the four types of objects in RA.

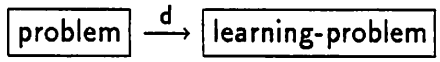
Object	Purpose
Problem	Stands for an object of inquiry in research.
Technique	Stands for the result of inquiry in research.
Property	Stands for a statement about a problem or technique.
Concept	Some idea that is involved in a problem or technique.

Table 2.3: RA's Objects

2.2.2.2 Semantics of Relations

This section describes the semantics of the nine relations defined in RA. Three of these are well-known epistemological relations that stand for subsumption, instantiation, and encapsulation. Out of the other six, five are specific to research domains; and the sixth is specific to machine learning. In RA's case, the terms 'domain' and 'domain-dependent' are somewhat misleading. I will reserve the use of the term 'domain' to refer to the domain of research and use the term 'subject' to refer to machine learning and EBL. Hence three of RA's relations are domain-independent, five are domain-dependent, and one is subject-dependent. These relations are described below:

Dominates: This relation asserts a type/subtype relationship between two objects. For example, the assertion '(dominates problem learning-problem)' means that learning-problem is a subtype of problem. This is depicted as

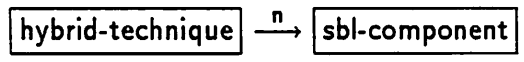


This relation is essentially the same as the dominate relation used in Kodiak [Wilensky 87]. It should be contrasted from the infamous 'isa' relation — 'isa' has been traditionally used to depict two conceptually different relations: type/token and type/individual. In RA, as in Kodiak, dominates denotes the former and instantiates denotes the latter.

Instantiates: This relation asserts a type/individual relation between two objects. For example, the assertion '(instantiates learning-technique version-spaces)' means that version-spaces is a specific individual of the type learning-technique. This will be depicted as

This relation is more or less the same as the instantiate relation in Kodiak and the individuates relation in Klone [Brachman 79]. In Brachman's terminology, instantiation is a relation in the real world, and individuation is a *representation* of the real-world relation. For example, the object Eiffel Tower (the one in Paris) is an instance of the concept 'tower.' The token eiffel-tower *denotes* the object Eiffel Tower in a representational language. The token eiffel-tower is said to *individuate* the concept 'tower,' whereas the object Eiffel Tower is said to *instantiate* the concept 'tower.' Brachman's terminology has never quite caught on, and for simplicity, I will use the term 'instantiate' in lieu of 'individuate'[12].

Encapsulates: This relation asserts a part/whole relationship between two objects. For example, the assertion '(encapsulates hybrid-technique sbl-component)' means that sbl-component is a sub-part of hybrid-technique. This is depicted as



Languages such as Klone and Kodiak use a relation called 'Role' that has far more structure than 'encapsulates.' In Kodiak, for example, one could define a concept called commercial-transaction, and then say that this concept has the roles transacter, transactee, and transacted-object. Further, the first two roles may be constrained to take only humans as their value and the third role constrained to take only a non-human as

[12] Btw Note that the direction of the 'instantiate' relation as used in RA is opposite the direction as used by Brachman. In my use of the relation, a concept instantiates an individual, while in Brachman's terminology, the individual instantiates the concept.

its value[13]. When you define a new-concept called buying-event as a specialization of commercial-transaction, Kodiak will automatically inherit the roles and role constraints from the parent concept, commercial-transaction. This kind of inheritance is called 'structured inheritance' [Brachman & Schmolze 85]. The 'encapsulates' relation in RA states only that some object is encapsulated within another, without specifying any role names. Property inheritance does not play a role in RA since almost all EBL concepts are polymorphic, i.e., different instances of a concept have different properties. For example, few EBL systems have all the properties of an ideal system; Keller's LEXCOP system [Keller 87] does not require an input example; those that use Schank's notion of 'explanation' [Schank 86] do not have a clearly defined generalization phase, and so on. Thus a polymorphic concept has several disjunctive components and there is little at the core of the concept that is common to *all* of its members[14]. In the context of polymorphic concepts, property inheritance becomes a nuisance than a help.

The above three relations are RA's epistemological relations. In Brachman's view, epistemological relations are those that form the 'conceptual coat-rack' from which other domain-dependent entities are hung. Five of the other six relations in RA are domain-dependent and one, 'acq,' is subject-dependent.

Solves: This relation asserts that a technique solves a problem. For example, the assertion '(solves version-spaces concept-learning-problem)' means that version-spaces is a technique that solves concept-learning-problem. This is depicted as

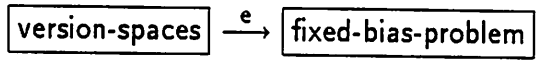
The important thing to note about this relation is that its meaning depends entirely on the semantics of the objects 'problem' and 'technique.' If we use the term problem to stand for a theorem and the term technique to stand for a proof, then this relation denotes what we normally mean by 'proves.' If we use the term problem to denote a hypothesis and the term technique to denote an experiment, then this relation denotes

[13] Btw Wendy Lehnert uses sentences such as "The sheik bought Mary from John," to show that hard slot constraints are a problem in sentence processing. Her latest parser, CIRCUS, uses 'connectionist-motivated' techniques to implement soft constraints [Lehnert 88b].

[14] Btw This is not the same as 'disjunctive concepts' that several machine learning researchers have been pre-occupied with. Disjunctive concepts, while abundant in toy domains such as boolean functions, are probably quite rare in the real world. Briefly, the difference between polymorphic and disjunctive concepts is this: in a polymorphic concept, there is a significant intersection (known as 'family resemblance' [Rosch & Mervis 75]) between any two instances. When you take all the instances of the concept, however, there are few features that they all have in common. Whereas a disjunctive concept is characterized by a union of several equivalence classes — two instances either have all features in common or none at all. In some circles [Smith & Medin 81], it is believed that the only place for disjunctive concepts is at a level above the 'basic level' [Rosch et al 76]. For example, a concept like 'clothing' which is a super-ordinate category (See Chapter 6) may be seen as a disjunction of basic-level concepts such as 'pants,' 'shirts,' 'jackets' etc. I would argue that disjunctive concepts are not particularly important concepts in the real world: often, tribal languages [Berlin 78] and the American Sign Language (ASL) [Newport & Bellugi 78] do not have superordinate concepts.

what we normally mean by 'verifies.' 'Solves' is the primary relation connecting problems and techniques.

Entails: This relation asserts that a technique has an emergent problem or a deficiency. For example, the assertion '(entails version-spaces fixed-bias-problem)' means that version-spaces has an emergent problem called fixed-bias-problem. This is depicted as



It is not always easy to tell what an emergent problem is. A general heuristic was used in deciding what should be called an emergent problem. If more than one paper refers to a technique T1 as having a deficiency, then a separate token P1 was created and marked as entailed by T1[15]. Under this heuristic, fixed-bias-problem is an emergent problem of version-spaces since several researchers have written about it. Any problem that fails this heuristic is not explicitly represented. For example, Samuel's original technique to play checkers [Samuel 59] had several small deficiencies; these were fixed by Samuel using a technique called 'signature tables' [Samuel 67]. It appears that nobody else pursued these deficiencies to propose solutions for them. Hence these would not be accorded a separate 'emergent-problem' status. Instead, we will say that signature tables simply *extend* Samuel's original technique (Also see the discussion under the relation R). The relationship between solves and entails may be understood as follows: problems give rise to techniques — solves denotes the relationship between them. In turn, techniques give rise to new problems — entails denotes this relationship.

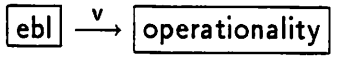
Exhibits: This relation asserts that some object exhibits some property. For example, the assertion '(exhibits sbl search-property)' means that similarity based learning (sbl) has a property called search-property[16]. This is depicted as

This is the primary relation linking problems, techniques, and concepts to properties — i.e., the relation linking zeroth- or first-order objects to second-order objects. The corresponding relation that links second- or first-order objects to zeroth order objects is the 'involves' relation described below.

Involves: This relation asserts that some object involves some concept. For example, the assertion '(involves ebl operationality)' means that operationality is a concept that is somehow involved in the definition of EBL. This is depicted as

[15]This is similar to Lenat and Guha's heuristics for creating concepts in CYC [Lenat & Guha 89]. In CYC, something is accorded concept status if one has enough things to say about it. In RA, an emergent problem is explicitly represented as a token if there is an expectation that several papers will refer to that token in the future as a problem to be solved.

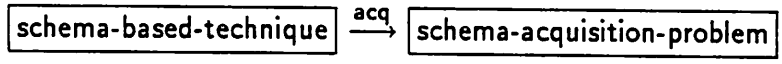
[16]ML This property is the statement that all inductive learning methods may be viewed as a search in a hypothesis space [Mitchell 81].



This relation links first or second order objects to the zeroth order objects of type concept. This relation is used rarely in research schemas.

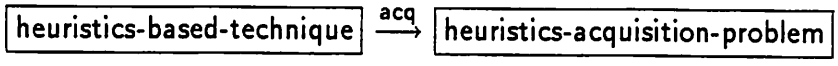
R: This relation asserts that some unknown relationship holds between two objects. Unlike the above relations, R is a relation schema than a single unitary relation. It stands for any relation that is not expressible in RA's language. For example, Samuel's signature tables [Samuel 67] extends his original technique to play checkers [Samuel 59]; Rosenbloom's property [Rosenbloom 88] generalizes Mitchell's search property [Mitchell 81]; Mahadevan et al's framework for learning control knowledge [Mahadevan et al 88] is analogous to Valiant's framework for learning concept descriptions [Valiant 84]. Rather than define several such relations, RA uses a single relation R to depict arbitrary domain-specific relationships. As a convenience, RA uses an extra slot with all R's to denote what that R stands for. When it is clear from context, I will use this slot name rather than R in depicting domain-specific relationships. Hence, when I want to say that Rosenbloom-88-property R (generalizes) search-property (see Chapter 1, Section 1.1.4), I will say that Rosenbloom88-property generalizes search-property and depict this as

Acq: While all the above relations are independent of machine learning and EBL, this relation is specific to the learning domain. It is used to state how a learning problem is related to the rest of the world. An example best illustrates its use. In his doctoral work, Gerald DeJong built a system called FRUMP that uses a technique based on 'sketchy scripts' (a kind of schemas) to understand newspaper stories [DeJong 79]. Let's call FRUMP's technique schema-based-technique. This technique solves the story-understanding-problem. After completing his doctoral work, DeJong realized that the success of FRUMP hinged on its access to hundreds of schemas that were handcoded into the system. His research focused on how a system could automatically acquire these schemas. Let's call this the schema-acquisition-problem. DeJong developed his explanatory-schema-acquisition technique to learn schemas like the ones he used in FRUMP. The relation acq denotes the relationship between schema-based-technique and schema-acquisition-problem. This will be depicted as



To illustrate this relation better, let me mention two other examples. AM is a discovery system developed by Douglas Lenat as part of his doctoral work [Lenat 76]. Starting from 100-odd 'pre-numerical' concepts, AM used 200-odd discovery heuristics to discover several well-known mathematical concepts such as prime numbers and unique factorization. Realizing that AM's success was due to the power of its heuristics (and its deficiency due to its inability to learn new heuristics), Lenat built another

system called Eurisko to learn heuristics automatically [Lenat 83]. Let's say that AM's heuristics-based-technique solves the discovery-problem. Let us further say that Eurisko's heuristics-acquisition-technique solves the heuristics-acquisition-problem. 'acq' denotes the relationship between AM's heuristics-based-technique and Eurisko's heuristics-acquisition-problem. This can be depicted as



One more example that's hard to resist: RA I is a system developed by Kishore Swaminathan as part of his doctoral work. RA I performs several tasks in the domain of research using a single representational mechanism called 'research schemas.' Since RA I depended so heavily on the notion of research schemas, he built another system called RA II to learn these schemas automatically. Let's say that RA I's research-schema-technique solves the problem-of-research. Let's say that RA II's research-schema-acquisition-technique solves the research-schema-acquisition-problem. 'acq' denotes the relationship between research-schema-technique and research-schema-acquisition-problem[17]. This can be depicted as

Table 2.4 contains a summary of what RA's relations are supposed to model. The 4 object types and the 9 relations constitute the representational primitives for representing RA's S-knowledge. With 4 object types and 9 relations, there are 144 possible combinations involving two types and a relation. Of these, the 'acq' relation can only state a relation between a 'performance-technique' and a 'learning-problem.' 'R' is allowed to state any relation between any two objects of any type resulting in 16 possible relations involving R. When the type constraints are imposed, the other seven relations can partake in only 20 out of a possible 112 relations. These are shown in Table 2.5. The table contains the seven relations as row labels. Each column contains a pair of objects one below the other. A tick mark denotes that the relation in its row is a legal relation from the first object in its column to the second object in the same column. All illegal combinations are left as blanks.

Some readers may have noticed that this section used the term 'semantics' mostly in a model-theoretic sense: we talked at length about what the objects and the relations are supposed to model (i.e., their denotation) rather than what the system does with these objects. Section 2.3.1 discusses the notion of semantics as it applies to RA.

[17] BTW The relation acq was originally defined to state precisely the relationship between RA I and RA II. Ponder that one! As irony would have it, acq had to be dropped from RA II because it does not satisfy the properties of associativity and discriminability required by RA II's learning strategy. See Chapter 5.

Relation	Purpose
Dominates	To state that one class subsumes another class.
Instantiates	To state that a class has an instance.
Encapsulates	To state that one object is part of another object.
Solves	To state that a technique solves a problem.
Entails	To state that a technique has an emergent-problem.
Exhibits	To state that an object exhibits a property.
Involves	To state that an object involves the use of a concept.
R	To state that an object is somehow related to another object.
Acq	To state that a learning problem stems from some technique.

Table 2.4: RA's Relations

	P P	T T	P T	T P	P C	T C	C C	T Pr	P Pr	C Pr	Pr C	Pr Pr
Dominates	✓	✓					✓					✓
Instantiates	✓	✓					✓					✓
Encapsulates	✓	✓					✓					✓
Solves				✓								
Entails				✓								
Exhibits								✓	✓	✓		
Involves					✓	✓					✓	

Table 2.5: Relations and Their Type Constraints

2.3 *Discussion

This section discusses some advanced topics related to RA's representation. While these are important to understand the research contributions of this dissertation, they may be skipped on a first reading without loss of continuity.

In Section 2.3.1, I discuss the notion of 'semantics' and how it applies to RA. Section 2.3.2 discusses the notion of 'structural level' and how RA's primitives may be seen as characterizing research literature at this level of abstraction. Section 2.3.3 is a philosophical discussion of science and knowledge evolution.

2.3.1 Semantics

When confronted with a 'knowledge representation' scheme, it is fair to ask "But what is your semantics?" The answer to this question is somewhat complicated in RA's case. This section is aimed at clarifying the notion of 'semantics' as it applies to RA.

Typically, the discussion of a 'knowledge representation' scheme is cast in terms of what a computer system does with the representation. For example, the semantics of the 'isa' link can be defined in terms of the inheritance properties that an inference mechanism attributes to that link. Suppose we say

and that the token cat has properties arms, legs, eyes, sleepiness and so on; then the semantics of the isa link may be defined as follows: when the inference mechanism is asked "Does Winnie have legs?," it should say yes because cat has legs and winnie isa cat[18]. Representational languages such as Kodiak and Klone define a set of so-called 'epistemological' relations whose semantics are precisely defined by the domain-independent inference mechanism.

For the sorts of everyday objects such as cats, dogs, and of course, penguins, we generally assume that we know what we mean (in a model-theoretic sense) when we say '(isa winnie cat)'. The question of semantics focuses exclusively on *what the system knows* when it has the representation above. In other words, Kodiak and Klone do not propose a theory of the world, but provide a neutral mechanism to represent any such theory.

In contrast, let's take a representation such a Schank's 'Conceptual Dependency' (CD) theory [Schank 73]. The CD theory is more a theory of the ontology of the world, rather than a theory-neutral representational mechanism. The notion of semantics with such a theory is primarily denotational or model-theoretic. This means, "Do we, humans, understand what we mean when we say (atrans john book mary)?" We need to specify

[18] Biw See McDermott [McDermott 76] for several amusing misuses of the 'isa' relation.

what 'atrans' means first; the symbol-level semantics is secondary. The following is Newell's [Newell 82] characterization of the CD theory:

> Providing a simple model such as this [CDs] constitutes a contribution at the knowledge level — to how to encode knowledge of the world in a representation ... For many of us, the meaning of conceptual dependency seemed undefined without a process that created conceptual dependency structures from sentences. Yet, when this was finally forthcoming (in Margie), there was nothing there except a large AI program containing the usual sorts of things, e.g., various ad hoc mechanisms within a familiar framework (a parser, etc). What was missed was that the program was simply the *implementation* of the model in the obvious, i.e., straightforward, way. The program was not supposed to add significantly to the specification of the mapping. There would have been trouble if additions had been required, just as a computer program for partial differential equations is not supposed to add to mathematics. (Newell, p. 120)

Newell's characterization of CD's is precisely on the mark. When a representational scheme purports to encode some piece of world knowledge in some new way, the first question of semantics should be one of 'denotation': What real world objects do the symbols denote? This tack is also seen in Lenat and Guha's monograph on CYC's representation [Lenat & Guha 89]. This 150+ page monograph focuses almost entirely on how to carve up the world into concepts; from this, they go on to discuss what real world objects their symbols stand for. The symbol-level semantics, i.e., what CYC does with the representation (called CYCling), is a secondary consideration.

In Chapter 6, I will develop a framework for classifying representational theories. This framework distinguishes between ontological and epistemological theories of knowledge and representation. Under this framework, RA's contributions can be seen as an ontological theory in that it identifies the kinds of knowledge involved in some intelligent behavior; the RA program is simply an implementation of the ontological theory in "the obvious, i.e., the straightforward way." Chapter 3 describes the RA program and explains what RA does with the representation.

2.3.2 Structure

I have stated earlier that RA's representation views research papers at a very high level of abstraction to capture the *structural* relations among problems, techniques, properties and concepts. The notion of 'structure' is not easy to define; this section is aimed at clarifying rather than defining my use of the term. For a

related discussion on the differences between a structural and an intensional representation, see Section 6.3 which compares RA's representation to that of the AM system [Lenat 76].

Structural levels of abstraction may be distinguished from *deep-semantic levels* of abstraction[19]. To understand this distinction, let's see how you might read a research paper:

- At the deep semantic level, your concern is with the internal semantics of the various entities in the paper. With a paper in Explanation-Based Learning, you want to understand the details of the learning problem, the domain theory that is used, the internals of the explanation and the generalization mechanisms, and so on.

- At the structural level, your concern is with how the entities in a paper are related to each other and other known entities in the field. In other words, you are mostly concerned with the relationship among entities rather than their internal semantics.

There are two important differences between representations that emphasize deep-semantics and those that emphasize structure: (1) Deep-semantic representations involve objects at multiple levels of abstraction whereas structural representations view all objects from the same level of abstraction. (2) Deep-semantic representations use type hierarchies to relate objects at different abstraction levels; in a structural representation, even if there is a type hierarchy, the underlying representation language does not attribute any special semantics (e.g., inheritance) to the relations that assert hierarchical relationships.

This notion is not entirely new in AI and I will give two examples below. Structural relations with the above two properties occur in Lehnert's work on narrative summarization [Lehnert 81]. In this work, Lehnert views a narrative as consisting of events that affect the characters in the narrative in one of three ways: they cause a character to have a negative, positive, or neutral 'affect state.' For example, assume that John wins two million dollars in a lottery, and then loses it all in an AI company that promises to commercialize RA. We have two events here: 'John winning the lottery' and 'John losing his investment.' The first event causes John to have a positive affect state, and the second causes him to have a negative affect state. These two states are linked with a relation called 'terminates' to denote that the first (positive) state is terminated by the second (negative) state. Proceeding this way, Lehnert constructs an affect state map of an entire narrative. This map is then converted into a 'plot-unit graph' that identifies large structures of stereo-typical affect state configurations. Finally, a summary of the narrative is constructed in terms of the plot-unit that has the highest connectivity.

[19] BTW The terms 'deep semantics' and 'deep semantic processing' (DSP) were first used by Eugene Charniak in his doctoral dissertation [Charniak 72] in order to distinguish between processing a sentence out of context (called 'internal translation') and in context (called DSP). While I have borrowed the term from Charniak because of its aptness here, my use of this term is completely different from his. In fact, one could say that my use is exactly opposite to that intended by Charniak: I use DSP to refer to the internal semantics (irrespective of context) of a research paper, and the term 'structure' to refer to how a paper fits into the overall context of research in its field.

Let us focus on affect state maps. All the defined types — positive, negative, and neutral affect states — are at the same level of abstraction; i.e., there is no type hierarchy among them. The set of relations (termination, actualization, motivation, and equivalence) are once again at the same level of abstraction; there is no hierarchy among them. An affect state map may be seen as a network of structural relations among objects belonging to a single abstraction level[20].

The notion of structure is rampant in theories of analogical mappings: Let's consider the work of Gentner [Gentner 83]. In Gentner's structure mapping theory, a piece of knowledge, say k_b, from a base domain, is mapped to another piece of knowledge, say k_t, in a target domain, in order to find an analogy between the two. This theory maintains that the mapping process preserves higher order structure in the base-domain when k_b is mapped on to k_t. To illustrate structure mapping and to distinguish structural inference from deep semantic inference, let us consider an example from Schank's *Dynamic Memory* [Schank 82]. A concept such as 'visit to a health professional' subsumes the concepts 'visit to a dentist' and 'visit to a psychiatrist' — i.e., the former is more abstract than the latter two. These three concepts form a type hierarchy with the 'health professional visit' on top and the 'dentist visit' and the 'psychiatrist visit' at the bottom. With a type hierarchy like this, one expects to make inferences through inheritance: you probably need an appointment to visit a health professional, therefore you'll need one to visit a dentist; you normally wait in a waiting room (probably with plants and a fish tank) before seeing a health professional, therefore you could expect to do the same before seeing a psychiatrist. Further, the subtypes may add additional semantics specific to the subtypes. For example, what distinguishes a dentist visit from anything else is that you have work done on your teeth. Deep semantic inferences deal with type hierarchies where a concept inherits properties from higher-level concepts. To the inherited properties, a concept adds new ones specific to itself.

In contrast, the structure mapping theory allows structural inferences as follows: a dentist visit is analogous to a psychiatrist visit because they are both subsumed by the health professional visit[21]. There is a subtle but important distinction between deep semantic inference and structural inference: even if there is a type hierarchy due to the subsumption relation, structural inferences do not attribute any 'subsumption' semantics

[20] BTW The same cannot be said about the relationship between affect state maps and plot units, nor about the relationship between two plot-units. A plot-unit (which is some stereo-typical configuration of affect state units) essentially provides a higher-level vocabulary for intentional situations in narratives. In Lehnert's work, the mapping from affect state maps to plot-units is mostly magic. There is no formal relation to state the relationship between an affect state map and its corresponding plot unit. Similarly, different plot-units may be related to each other in deep-semantic ways: for example, a plot-unit such as 'request honored with conditional request' may be a supertype of the plot-unit 'honored request.' Once again, Lehnert provides no mechanism to state such deep-semantic relationships.

[21] BTW This description doesn't do full justice to Gentner's work. There are several other aspects to her work such as the 'order' of the relations and the 'systematicity' principle. They are not discussed here because they do not contribute to our current discussion on the notion of structure.

to this relation. With respect to structure, 'health professional visit,' 'dentist visit,' and 'psychiatrist visit' are all objects at the same level of abstraction. Each of the last two objects (dentist visit and psychiatrist visit) is related to the first one by a relation labelled 'subsumption.' The structural inference about the subsumption relation is no different from a structural inference about any other relation. For example, assume that there is another object called 'an-arm-and-a-leg' and that there is a relation called 'costs' that links it to 'dentist-visit' and 'psychiatrist-visit.' A structural inference mechanism would find the following analogy between 'dentist-visit' and 'psychiatrist-visit': A dentist visit is analogous to a psychiatrist visit because they both cost an-arm-and-a-leg. The point of this long example is that the main difference between structural representations and deep-semantic representations is that the former views all objects from a single level of abstraction focusing on the relationships among them, whereas the latter has a type hierarchy and therefore multiple levels of abstraction. This statement is true even if the representational language has a 'subsumption' relation to construct a type hierarchy. An inference mechanism that does not attribute any subsumption semantics to a subsumption relation treats a subsumption relation like any other structural relation.

RA's representation has a subsumption relation (dominate) as well as an instantiation relation. Only toward the end did I realize that these relations were really no different from any other relation in the system; their usual semantics (such as inheritance and the type/individual distinction) played no part with respect to most of RA's functionalities. To illustrate this, I'll show how the suggestion component of RA uses a research schema to suggest a research direction. Assume that RA's knowledge-base consists of the following relations:

```
{(dominates learning-problem concept-learning-problem)
(dominates learning-problem control-learning-problem)
(dominates learning-problem discovery-problem)
(solves john-83-technique concept-learning-problem)
(instantiates EBL john-83-technique)}
```

Now assume that a new paper, [Mary 84], comes along with the following schema:

```
ref: {(dominates learning-problem concept-learning-problem)
      (dominates learning-problem control-learning-problem)
      (instantiates EBL john-83-technique)
      (solves john-83-technique concept-learning-problem)}

def: {(solves mary-84-technique control-learning-problem)
      (instantiates EBL mary-84-technique)}
```

This schema corresponds to the following abstract:

> [Mary 84]: Learning problem consists of two kinds: concept-learning-problem and control-learning-problem [God 08]. John has shown that an EBL technique, John-83-technique, solves the concept-learning-problem [John 83]. In this paper, I propose an EBL technique called Mary-84-technique to solve the control-learning-problem.

The above abstract corresponds to the following skeletal schema:

ref: {(dominates P1 P2)
(dominates P1 P3)
(solves T1 T2)
(solves T2 P2)}

def: {(solves T3 P3)
(instantiates T1 T3)}

The skeletal schema may be seen as the following heuristic:

> If there is a problem P1 that dominates two other problems P2 and P3, and one of them, P2, is solved by a technique T2, which is an instantiation of a technique T1, then suggest, "Perhaps you can try to solve P3 with a technique that is an instantiation of T1[22]."

After acquiring this schema (and hence its corresponding heuristic), the system might make a future suggestion: "Perhaps you could try to solve 'discovery-problem' with a technique that is an instantiation of EBL." Even though this heuristic has the relations 'instantiates' and 'dominates', nowhere do we use these any differently from any other relations. All objects are viewed from a single level of abstraction and the relations act as no more than labelled arcs. The 'if' part seeks a structure of nodes that are related to each other by a set of specific arc labels; when one is found the 'then' part suggests how this structure can be grown.

In summary, the difference between structural representations and deep semantic representations is that the former views all objects from a single level of abstraction; the relations denote structural connections among these objects. Whereas deep-semantic representations involve objects at multiple levels of abstractions — i.e., there is a type hierarchy and each level of the hierarchy has its own specific semantics. A structural representation enables you to represent the relationships among entities belonging to the same level of abstraction. A deep semantic representation enables you to represent the internal semantics of various entities by means of a type hierarchy. For some more perspective on this distinction, see Section 6.3.1 for a comparison of RA with the AM system.

[22]Informally, this heuristic means that if you have one kind of technique to solve a problem, then suggest that the same kind of technique might be used solve a sibling problem.

2.3.3 Science and Knowledge Evolution

One common charaterization of textbooks in general is that they describe the 'knowledge' of the field without any reference to the process through which the knowledge accrued. This has been voiced by researchers belonging to several traditions, e.g., Kuhn [Kuhn 70], Polya [Polya 57], and a little closer to home, Lenat [Lenat 76]. In Kuhn's view, the emergence of textbooks signals the onset of a 'paradigm.' This means that the field has gathered a critical mass of non-controversial knowledge; the textbook writers can write about the knowledge, suppressing the process through which this knowledge accrued. The purpose of a textbook is to organize the knowledge as a coherent whole and relate different bodies of knowledge while downplaying the dizzying back-and-forth scientific rhetoric in which scientists engage before agreeing on a scientific principle[23]. In a nutshell, the criticism against textbooks is that they teach the student science, but not the scientific process or the scientific method; the textbook squeezes all sense of time and context out of science and presents only the 'facts'; a great deal of science goes unappreciated when robbed of this historical context[24].

One perspective on RA's S-knowledge is that it stores the scientific 'facts' similar to a textbook. The E-knowledge of RA, represented in terms of research schemas, reflects the scientific process and method. It probably didn't make your heart skip a beat when I slipped in 'process' and 'method' in the same sentence. The connection between them is not all that obvious. For example, Kuhn focuses on the process almost to the exclusion of the methods; Polya and Lenat focus on the methods to the exclusion of the process. With RA, it took me about two years to realize that they may be viewed as the same — 'research schemas' that were used as a representation for the process of science could also be used as intentional rules of action, i.e., rules reflecting the methods of science. The combination of RA's S- and E-knowledge, therefore, captures science as (1) as a set of facts about the world, (2) as a process, and (3) as a set of methods. RA's S-knowledge pertains to (1) and research schemas pertain to (2) and (3).

Interestingly, the view of science as a process and the view of science as a set of methods are in conflict with each other. Which view is appropriate for a given theory of science depends on the level of abstraction from which that theory views science. Higher the level of abstraction, the more apt the 'process' view; lower the level of abstraction, the more apt the 'method' view. Let me explain: Kuhn used a very high level of abstraction

[23] Btw Two classic examples of scientific rhetoric come to mind: Imre Lakatos' 'Proofs and Refutations' and Smith and Medin's 'Categories and Concepts.' Both these books treat their subject material in terms of arguments and counter-arguments.

[24] Btw Imagine an AI textbook a hundred years from now. How would this book present the work of Terry Winograd? [Winograd 72] It is hard to appreciate Winograd's work in a world where the controversy between procedural and declarative knowledge is long forgotten. As another example, you are invited to read Dijkstra's famous letter to CACM titled "Goto considered harmful," [Dijkstra 68] which essentially ushered in structured programming. To a student who learned programming under the new dogma, Dijkstra's letter is almost trite.

and saw science as moving from one world view (a set of theories and methodologies) to another. To put words into Kuhn's mouth, the scientist is simply driven by the currents of his time and his paradigm; a scientist doesn't voluntarily say, "OK, let me end the current paradigm and start a new one." A scientist simply takes part in the process and an observer (like a philosopher of science) attributes methods to the scientist's madness.

Kuhn's level of abstraction can be contrasted from Lenat's level of abstraction in AM. Lenat uses very low level primitives such as functions, relations, their domains, ranges and so on. His heuristics are mutation operators that are more appropriately seen as methods for doing science (in Lenat's case, elementary arithmetic) rather than a description of the process of science. This level of abstraction can pretty much ignore the larger processes of science (such as paradigm shifts) within which a given heuristic will be embedded.

To understand this better, let me draw an analogy with evolution. 'Speciation' and 'extinction' are process-level descriptions. Individual members of a species have little control over the process. A species doesn't use a 'method' to survive or die; any methods that may be perceived (for example, the mass suicide rituals of lemmings) mere attributions. In contrast, the theory of plans and goals advanced by Schank and Abelson [Schank & Abelson 77] uses a much lower level of abstraction. Plans and goals are properly seen as descriptions of the intentional methods that individuals use. They would not be the appropriate primitives on which to base a theory like natural selection[25].

I believe that the level of abstraction used in RA is a happy medium between the process view and the method view of science. Note that this is a belief, rather than a claim. Even though RA uses research schemas in both ways (i.e., to represent the process of science, and as methods when it uses them as heuristics), this dissertation offers no hard argument why RA's level of abstraction (in terms of problems and techniques) is appropriate for describing science as a process as well as describing science as a set of methods. I have not quite been able to come up with a good argument for (or against) research schemas being able to do double duty as a description of research papers in addition to being able to act as intentional rules for action.

2.4 Summary

In Section 2.1, I used a series of examples to illustrate how RA may be seen as a model for knowledge evolution. At one-level there is nothing new about the model: any semantic memory grows as you add more relations to it. The fifteen or so pages of this section were meant to drive home the point that while the S-knowledge grows, side-by-side RA's E-knowledge also grows. Each research schema in the E-knowledge is a characterization

[25] BTW Who knows? If Michelin guides stop publication and the human species becomes extinct due to starvation, perhaps plans and goals would be the right theory to to describe the extinction process. Ponder that one! If you didn't follow that bit about Michelin guides, never mind.

of a knowledge evolution event, a bridge between an old ontology and a new one. These events relate the new knowledge added by each paper to a small slice of the old ontology in order to include the context for the new information.

Section 2.2 discussed the representational primitives of RA and the motivation behind their choice. During the design of the system, I was frequently hampered by the sparsity of the representation and craved for a richer set of primitives. However, the rather small representational language was retained to concentrate on the major concerns of this dissertation.

In Section 2.3, I discussed several assorted topics related to the notion of semantics and structure, and the two viewpoints that characterize science. While these topics are integral to understanding the contribution of this dissertation, they are somewhat tangential to the research reported here. For those readers who skipped this section, an important point of Section 2.3.1 is worth summarizing here: this research is best understood as a 'knowledge-level' theory of how to organize an evolving memory. The RA program is simply an implementation of our model in the most straight-forward way.

Chapter 3

How RA Works

It was still cloudy when day dawned, our third day on board.

—Thor Heyerdahl, The RA Expeditions.

RA's memory organization in terms of research schemas supports all of RA's capabilities. While the algorithms used to achieve these capabilities are quite simple, almost trivial, they illustrate the strong interplay between RA's knowledge of EBL (S-knowledge) and RA's knowledge of how it evolved (E-knowledge).

This chapter describes the four kinds of tasks performed by RA: retrieval, research suggestions, chronological summarization and analogical summarization. Of these, the last three may be seen as three separate components, with the retrieval task distributed among these three.

The four tasks performed by RA can be grouped in various wasy. The suggestion and retrieval tasks are primarily S-knowledge tasks, and the chronological and analogical summarization tasks are primarily E-knowledge tasks. Seen from how these tasks use research schemas, the retrieval and analogical summarization tasks use them as indices, the chronological summarization task uses them as a representation, and the suggestion task uses them as heuristic rules. Finally, chronological and analogical summarization tasks deal with instantiated schemas and suggestion and retrieval tasks deal with skeletal schemas. Table 3.1 contains a guide to Chapter 3.

3.1 RA as a Computer-Aided Research System

The uninitiated could not hope to produce a boat in this particular shape by simply lashing papyrus reeds together on impulse.

—Thor Heyerdahl, The RA Expeditions.

Section	On first reading	Description
1	skim	Describes RA as a computer-aided research system. This is a novel idea.
2	read	Introduces a small knowledge base for use in all subsequent examples. Try to understand Figure 3.11.
3	read	Describes various kinds of retrieval in RA. Shows the strong interplay between S- and E-Knowledge.
4	skim	Describes the suggestion component. Read Section 3.4.2, and skim through the rest.
5	skim	Describes the chronological summarization task. If you understand Figure 3.11, this is straightforward.
6	skim	Describes the analogical summarization task. If you understand Figure 3.11, this is straightforward.
7	read	Describes some refinements. Some simple additions to RA's tasks can result in a powerful system.
8	read	Summary.

Table 3.1: Guide to Chapter 3

Since the primary thrust of this dissertation is the memory organization of RA, the various capabilities of RA were implemented only to show that RA's memory organization can and does support these capabilities. At every stage, I decided to use the simplest possible algorithms without concern for efficiency or elegance. Hence this dissertation makes no specific claims about any of the individual functionalities beyond the fact that RA's memory organization supports all of RA's tasks in a very natural way. However, since the idea of 'Computer-aided Research' is a new one, considerable effort went into the design of RA's user interface to illustrate that a computer-aided research system might in fact be a useful and practical idea. This chapter is written in that same spirit: I have tried to discuss novel ideas such as RA's user interface at some length even though this may not be of great interest to some readers, and glossed over well-understood ideas such as traversal of RA's memory.

In this section, I gather several aspects of RA's user interface in one place. If you are interested in RA as a Computer-Aided Research system, then this section may be of considerable interest to you. You may safely skip this section if you are bored with mice and menus[1].

[1] [Btw] RA is implemented on a TI Explorer II. It is implemented on top of a Frame system designed and written by Mike Greenberg at the University of Massachusetts. RA also uses several Common-Lisp and Zetalisp facilities and the Explorer window system software.

RA's user interface includes a hypertext component with hooks into RA's various functionalities. Figure 3.1 shows a screen of RA as seen by the user. This screen consists of four windows, labelled A, B, C, and D. These windows are described below:

- **A:** This window (labelled Session) is the main window on which RA displays most of its output. For example, short summaries generated by the hypertext system plus the output of the chronological and analogical summarization components are displayed here.

- **B:** This is a command window. User commands to RA are typed here. Since most of the interactions with RA are menu driven, this window is used sparingly.

- **C:** This window displays a permanent menu. This menu lists all the nodes in RA's S-knowledge store plus the names of all papers in RA's memory. Clicking-left on any item in this menu generates a command to the hypertext component: "I am interested in this item. What can you tell me about it?" Clicking-right on any item in this menu generates a command to the Suggestion component: "I am interested in this item. Can you suggest some research directions involving this item?"

- **D:** This window (labelled Text) is used by the hypertext component to display the text of papers.

In addition to the permanent menu, RA also uses several temporary (or pop-up) menus to query the user for various kinds of information.

When you click-right on a menu item, RA starts a Research-Session. The item on which you clicked is called an *anchor* node, and RA generates research suggestions involving that node. The research session runs as a separate process, overlaying its own window on top of window A. Figure 3.2 depicts the RA screen with a research-session window overlaid. All research suggestions generated by RA are displayed on this window. Typically, a suggestion refers to various nodes and papers in RA's memory. After seeing a suggestion, you can go back to the hypertext interface shown in Figure 3.1 to look at the papers or nodes referred to in the suggestion. Since the research session runs as a separate process, you can go back to this session from the hypertext interface at any point. If you accept a suggestion, then RA lets you create new nodes and (optionally) attach text to these nodes — i.e., so far as RA is concerned, you have just written a new paper based on RA's suggestion. If you reject the suggestion, RA comes up with more suggestions until there are no more heuristics that apply at the anchor node or until you choose a different anchor node[2].

[2]In principle, there is no reason why one cannot keep multiple research sessions (based on different anchor nodes) active at the same time, but the current implementation of RA allows you to maintain only one research session at one time.

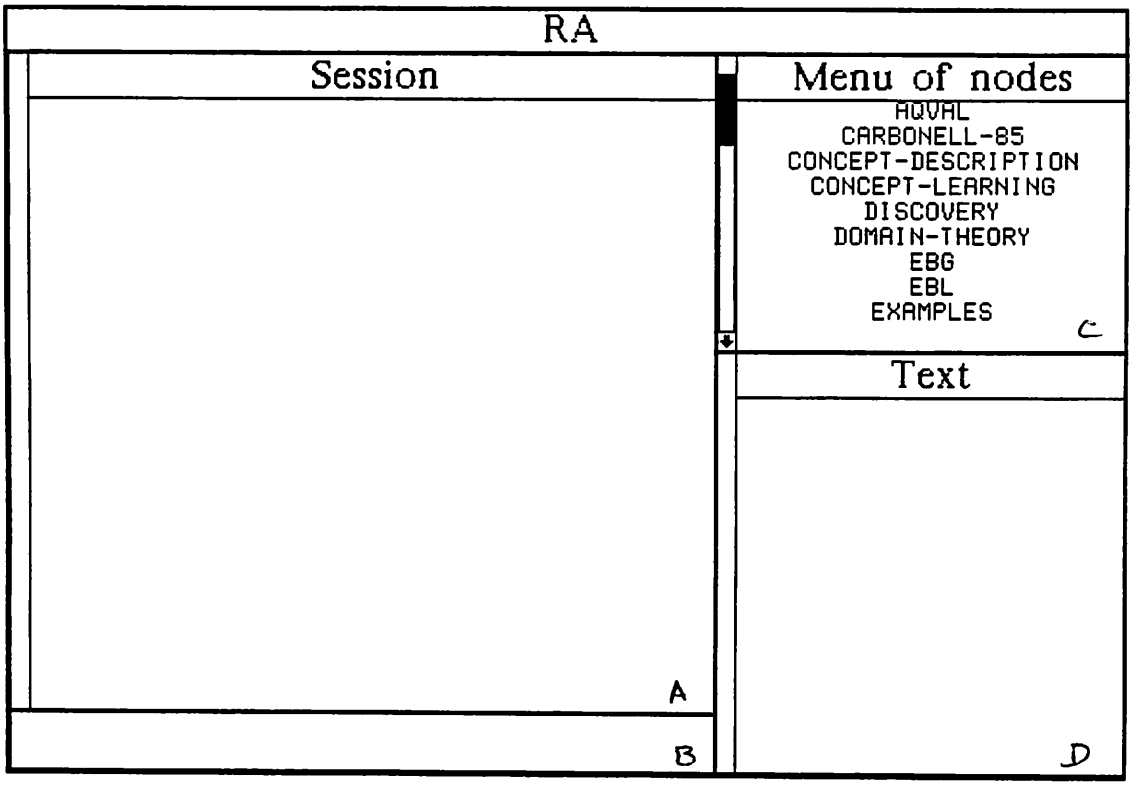


Figure 3.1: RA Screen

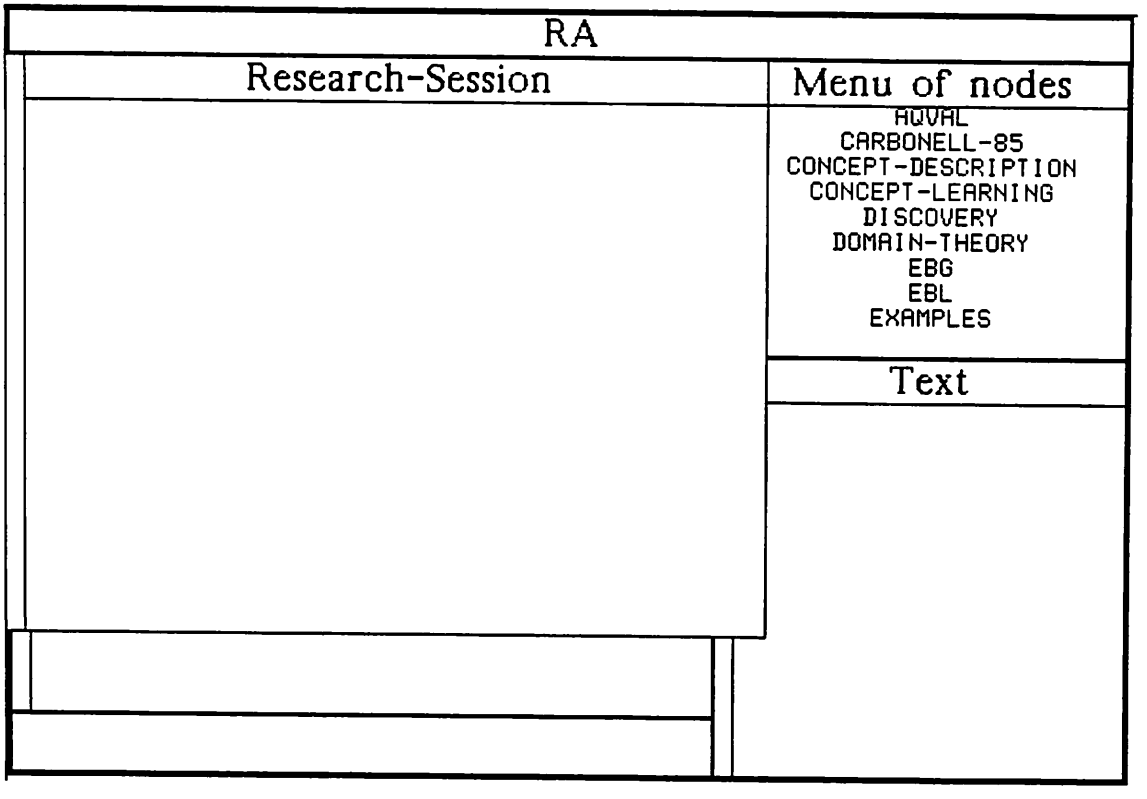


Figure 3.2: RA Screen with Research Session Overlaid

3.1.1 Dialog Revisited

In Chapter 1, I showed an example dialog between RA and the user. That dialog was doctored to give you a quick introduction to RA. This section revisits the same dialog to show how the interaction between RA and the user actually proceeds. Here I consider only the user interface aspects. Sections 3.3 to 3.6 describe each of the functionalities of RA.

Let's assume that you start with the screen as shown in Figure 3.1. You right-click on the EBL node to say: "I am interested in EBL. Can you suggest some research directions involving EBL?" This initiates a research session with EBL as the anchor node. Now RA overlays a research-session window on top of the regular RA window to display its suggestion. Figure 3.3 shows some of the initial suggestions generated by RA. The research session starts with the statement as to where the anchor node is described (in this case, EBL is described in [Mitchell-86]). At this point, you leave the research session, go back to the hypertext screen and find that the items EBL and Mitchell-86 have been placed at the end of the menu for quick access. When you click left on Mitchell-86, RA generates the following summary from the schema of this paper:

> [Mitchell-86]: CONCEPT-LEARNING-PROBLEM
> instantiates SCHEMA-ACQUISITION-PROBLEM.
> EXPLANATORY-SCHEMA-ACQUISITION solves
> SCHEMA-ACQUISITION-PROBLEM [DEJONG-85].
> CONTROL-LEARNING-PROBLEM instantiates
> LP-LEARNING-PROBLEM.
> LP-LEARNING-TECHNIQUE solves
> LP-LEARNING-PROBLEM [SILVER-83].
> In this paper, I show: EBL solves LEARNING-PROBLEM.
> EBL instantiates EXPLANATORY-SCHEMA-ACQUISITION
> and LP-LEARNING-TECHNIQUE.

In addition to generating this summary, RA also retrieves the text of the paper Mitchell-86.

Let's say that after reading the paper, you get back to the research session by clicking right on EBL again. RA has generated a simple suggestion that you could do a performance evaluation of EBL, and suggests that you look at '[Fisher 86]' for an example. You decline this suggestion by responding "No" to RA's query, "Do you want to continue this research direction?" RA goes through several other suggestions that you decline, until you hit the suggestion shown below:

> One way to look for limitations of EBL is to
> think if EBL really solves the class of
> problems LEARNING-PROBLEM. Can you identify
> subclasses of LEARNING-PROBLEM that EBL does

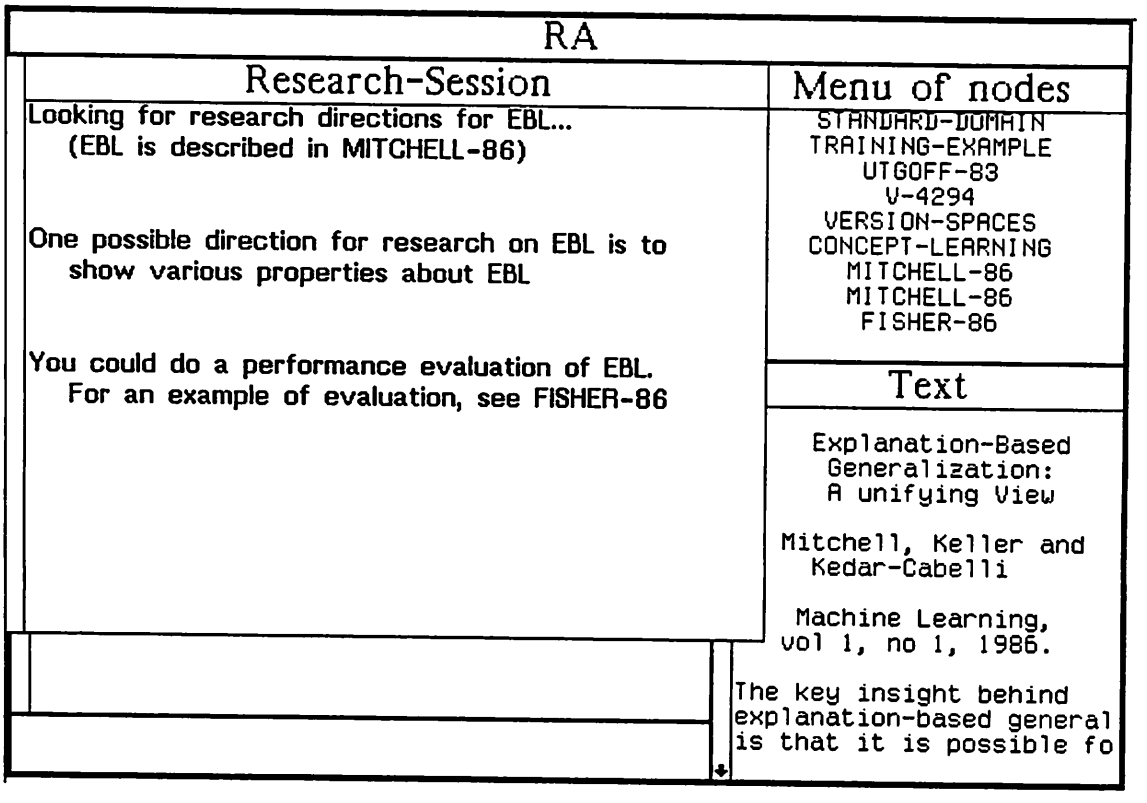


Figure 3.3: Some Initial Suggestions

not solve? LEARNING-PROBLEM is described in
CARBONELL-85. For an example of this
strategy, see MINSKY-69.

Assume that you think hard about this suggestion, and then decide to accept it. Now RA lets you create the new terms corresponding to this suggestion. This is depicted in Figure 3.4. Here you create a term imperfect-domain-problem as the class of problems not solved by EBL, and perfect-domain-problem as the class of problems that are solved by EBL. In RA's view, you have just written a new paper with the following schema:

ref: {(solves EBL learning-problem)}

def: {(dominates learning-problem perfect-domain-problem)
(dominates learning-problem imperfect-domain-problem)
(solves EBL perfect-domain-problem)
(not-solves EBL imperfect-domain-problem)
(entails EBL perfect-domain-problem)
(not-solves EBL learning-problem)}

The suggestion above was generated using the skeletal schema of [Minsky & Papert 69], hence when you accepted the suggestion, RA generates a schema for your "paper" that is derived from this skeletal schema. Further, RA also indexes the schema of your paper on the skeletal schema of [Minsky & Papert 69][3]. At this point, you could (optionally) attach text to this schema if you like.

Let's now assume that you choose the item imperfect-domain-problem as your new anchor node, directing RA to suggest research directions involving that node. One of these suggestions asks you to combine EBL with an SBL technique to solve an emergent problem of EBL[4]. Now you want to think about this suggestion seriously, so you go back to the hypertext screen and click on the SBL node. You see that there are several instantiations of SBL techniques, and you want to find out about one of them in particular: version-spaces. To get an idea of the research trends that led to version spaces, you type the following in the command window: (csummary version-spaces). RA generates the following chronological summary:

WINSTONS-TECHNIQUE solves CONCEPT-LEARNING.
[VERE-75] showed: WINSTONS-TECHNIQUE entails BACKTRACKING1.
VERES-TECHNIQUE solves CONCEPT-LEARNING.
VERES-TECHNIQUE not-entails BACKTRACKING1.

[3]See Section 4.4.3 for a full description of this schema.

[4]This heuristic is the Hegelian heuristic. If a technique does not completely solve a problem, then try solving it with a synthetic technique that combines that technique with another technique. Some other examples of such syntheses in AI include the combination of Rule-based reasoning and Case-based Reasoning (see, for e.g., [Skalak & Rissland 90]), and the combination of symbolic and connectionist techniques (see, for e.g., [Sumida & Dyer 89] [Lehnert 88b]).

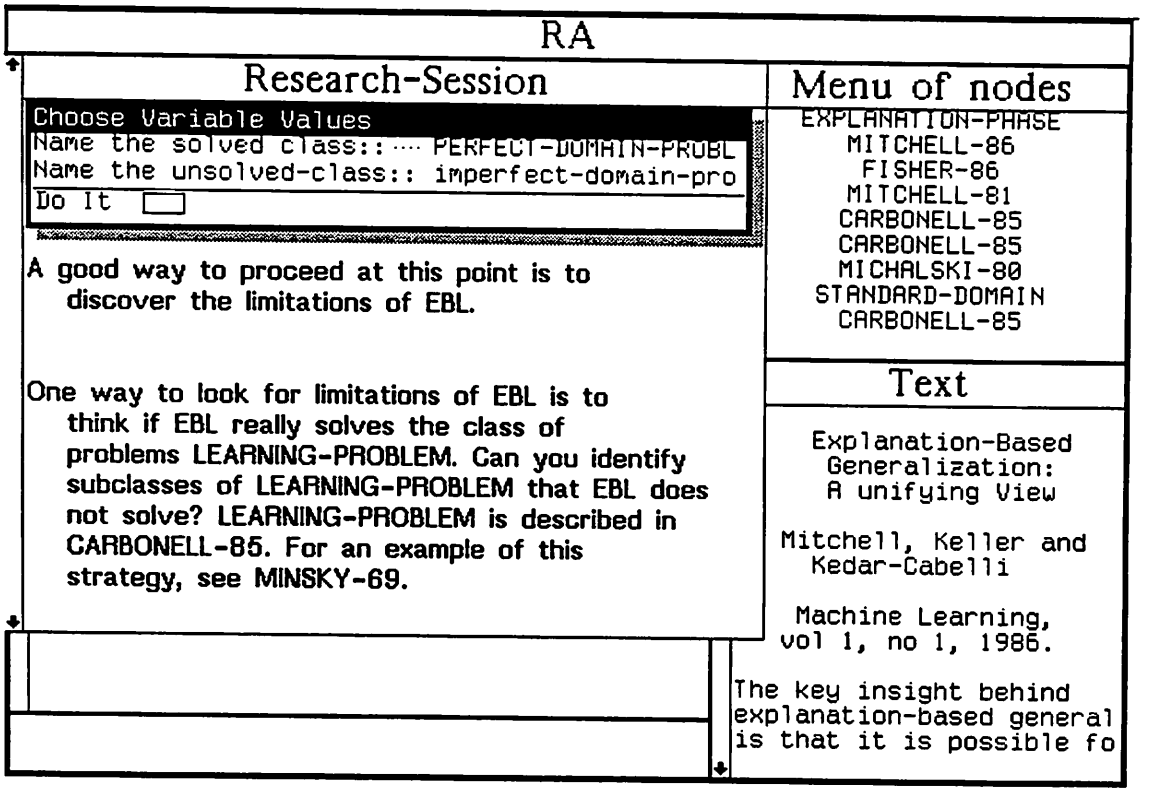


Figure 3.4: Accepting A Suggestion

[Mitchell 78] showed: VERES-TECHNIQUE entails BACKTRACKING2.
VERSION-SPACES solves CONCEPT-LEARNING.
VERSION-SPACES not-entails BACKTRACKING1.

See Figure 3.11 to understand the above summary. Suppose you now wonder if version-spaces itself has any emergent problems, and type '(entails version-spaces)', RA traces entails links from version-spaces and comes up with the following:

VERSION-SPACES entails FIXED-BIAS.
BIAS-ADJUSTMENT solves FIXED-BIAS [UTGOFF-84].

After reading all these papers and having understood all about SBL, let's assume that you come up with a hybrid technique super-duper-hybrid-technique to solve imperfect-domain-problem. You now go back to the research session, and accept RA's suggestion. RA now lets you create a new node (Figure 3.5), and attributes the following schema to your "paper":

ref: {(dominates learning-problem imperfect-domain-problem)
(solves SBL learning-problem)
(entails EBL imperfect-domain-problem)}

def: {(solves super-duper-hybrid-technique imperfect-domain-problem)
(encapsulates super-duper-hybrid-technique component1)
(encapsulates super-duper-hybrid-technique component2)
(instantiates EBL component1)
(instantiates SBL component2)[5]}

Let's now assume that the new term that you created, super-duper-hybrid-technique, is made the anchor node. One of RA's suggestions for this anchor node is shown below (also see Figure 3.6):

SUPER-DUPER-HYBRID-TECHNIQUE combines EBL and
SBL. You could see how properties of SBL
apply to SUPER-DUPER-HYBRID-TECHNIQUE. A
known property of SBL is "Inductive learning
is search" (described in MITCHELL-81).

This section illustrated in some detail the idea of Computer-Aided Research implemented in RA. As I said in Chapter 1, this idea is based on the interaction between a typical research advisor and a student, with RA attempting to duplicate some of the capabilities of a research advisor. Table 3.2 summarizes the user interface aspects of RA. The next several sections describe how RA's capabilities are achieved.

[5]This complicated looking schema states: given that imperfect-domain-problem is an emergent problem of EBL, and the parent problem, learning-problem is solved by SBL, then you propose a new technique that combines EBL and SBL to solve imperfect-domain-problem. The stylistic way to denote that a technique hybridizes two techniques A and B is to say that the technique encapsulates two components C and D which are instantiations of A and B.

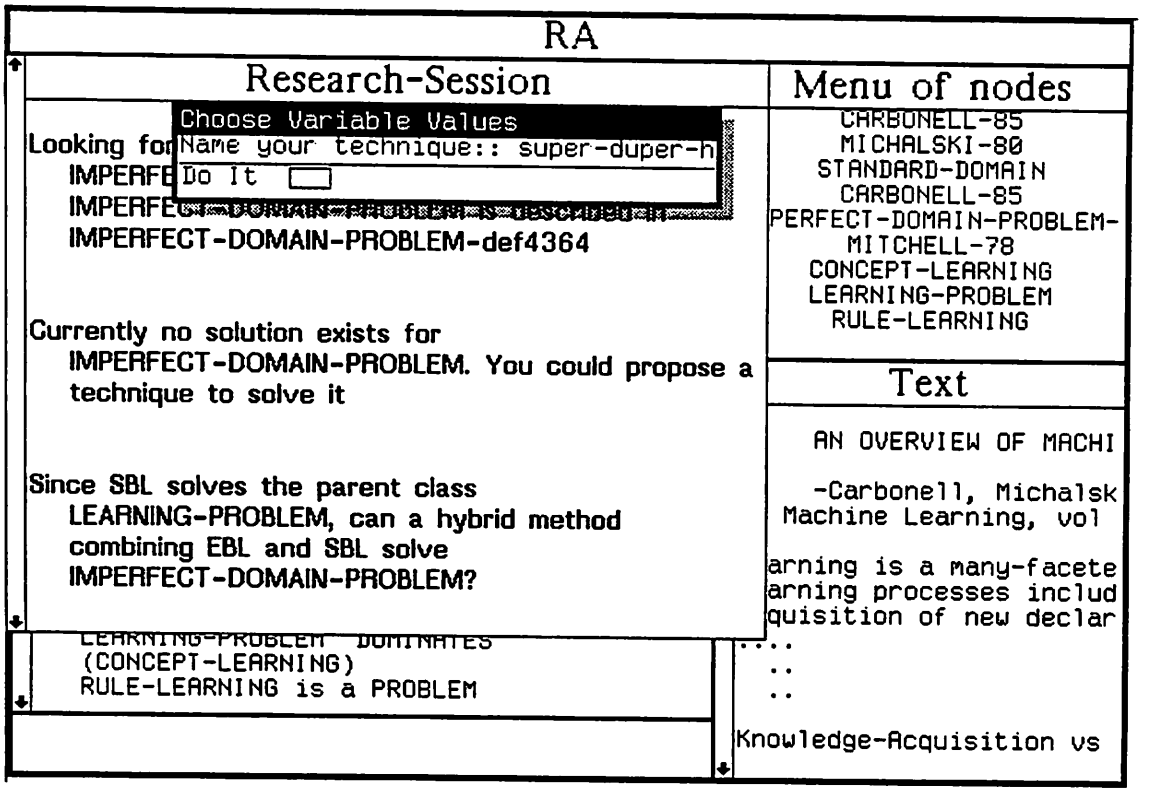


Figure 3.5: Suggesting A Hybrid Technique

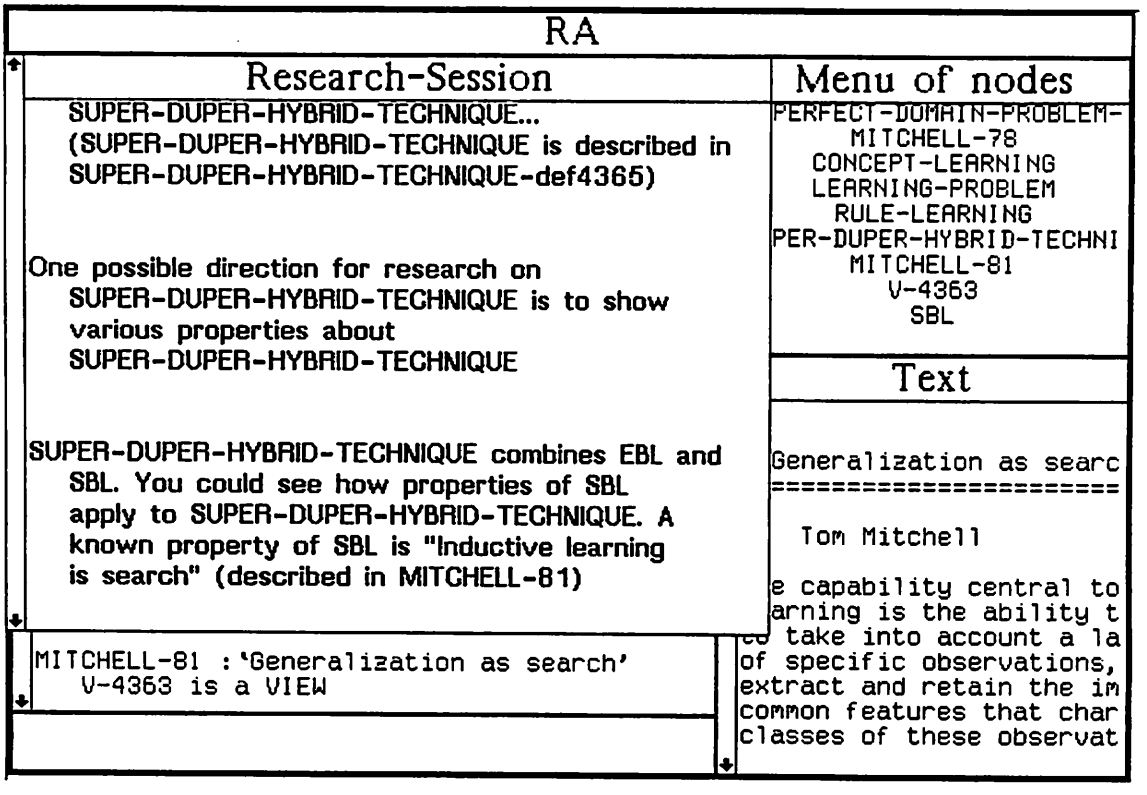


Figure 3.6: Another RA Suggestion

- Most of the capabilities of RA are accessed from its hypertext interface. See Figure 3.1.
- The Research Session runs as a separate process; the user can go back and forth between this process and the hypertext process. See Figure 3.2.
- Clicking left on a menu item generates a command, "I am interested in this item, what can you tell me about it?" This results in a short description plus the display of the text of the relevant papers.
- Clicking right on a menu item generates a command, "I am interested in this item, can you suggest some research directions involving this item?"
- When RA makes a suggestion, you can accept it or decline it. If you accept it, in RA's eyes you've just written a paper. RA already knows the paper's **ref**, and asks you to name the unknown objects in the **def**.

Table 3.2: Synopsis of RA's User Interface

3.2 Example: A Small Knowledge Base

This section introduces a small knowledge base of four papers. This knowledge base will be used by the next four sections to illustrate the various functionalities of RA. These four papers were chosen in order to provide good but simple examples for all of RA's functionalities. Appendix B gives a more complicated example to illustrate some of the advanced features of chronological summarization component.

The knowledge-base shown here consists of four papers, [Winston 71], [Vere 75], [Mitchell 78], and [Utgoff 84]. After Winston proposed a technique to solve the concept-learning problem, Vere showed that Winston's technique had an emergent problem of backtracking. Vere proposed a technique to solve the concept-learning problem, while avoiding the backtracking problem. Mitchell then showed that Vere had overlooked another kind of backtracking problem. Mitchell proposed his version-spaces technique to solve the concept-learning problem, while avoiding this problem. The schemas of these four papers are shown below:

[Winston 71]

ref: {}

def: {(solves winstons-technique concept-learning)}

[Vere 75]

ref: {(solves winstons-technique concept-learning)}

def: {(entails winstons-technique backtracking1)
(solves veres-technique concept-learning)
(not-entails veres-technique backtracking1)}

[Mitchell 78]

ref: {(solves veres-technique concept-learning)}

def: {(entails veres-technique backtracking2)
(solves version-spaces concept-learning)
(not-entails version-spaces backtracking2)}

[Utgoff 84]

ref: {(solves version-spaces concept-learning)}

def: {(entails version-spaces fixed-bias)
(solves fixed-bias bias-adjustment)}

Figures 3.7 to 3.10 depict the gradual addition of knowledge to RA through these four schemas. In each figure, the right side of the figure shows the S-knowledge of a paper, and the left side shows the E-knowledge in terms of research schemas. The **ref** and **def** links link the right side to the left side. Figure 3.11 shows how the skeletal schemas and the text are related to the rest of the memory. This small memory will be used by the next four sections to illustrate RA's functionalities.

3.3 Retrieval

The retrieval functionality is not a single component but is distributed across all the other components of RA. The various kinds of retrieval performed by RA are described in this section. They are depicted pictorially in Figure 3.12; Table 3.3 contains a synopsis. As I said earlier, although the retrieval algorithms are quite simple in nature, they illustrate the strong interplay between RA's subject knowledge (S-knowledge) and its evolutionary knowledge (E-knowledge).

3.3.1 Retrieval of a Paper on a Semantic Index

Suppose you select an item belonging to RA's subject knowledge (by clicking-left with the mouse). RA generates a short description of that item. This description depends on the type of the item as follows:

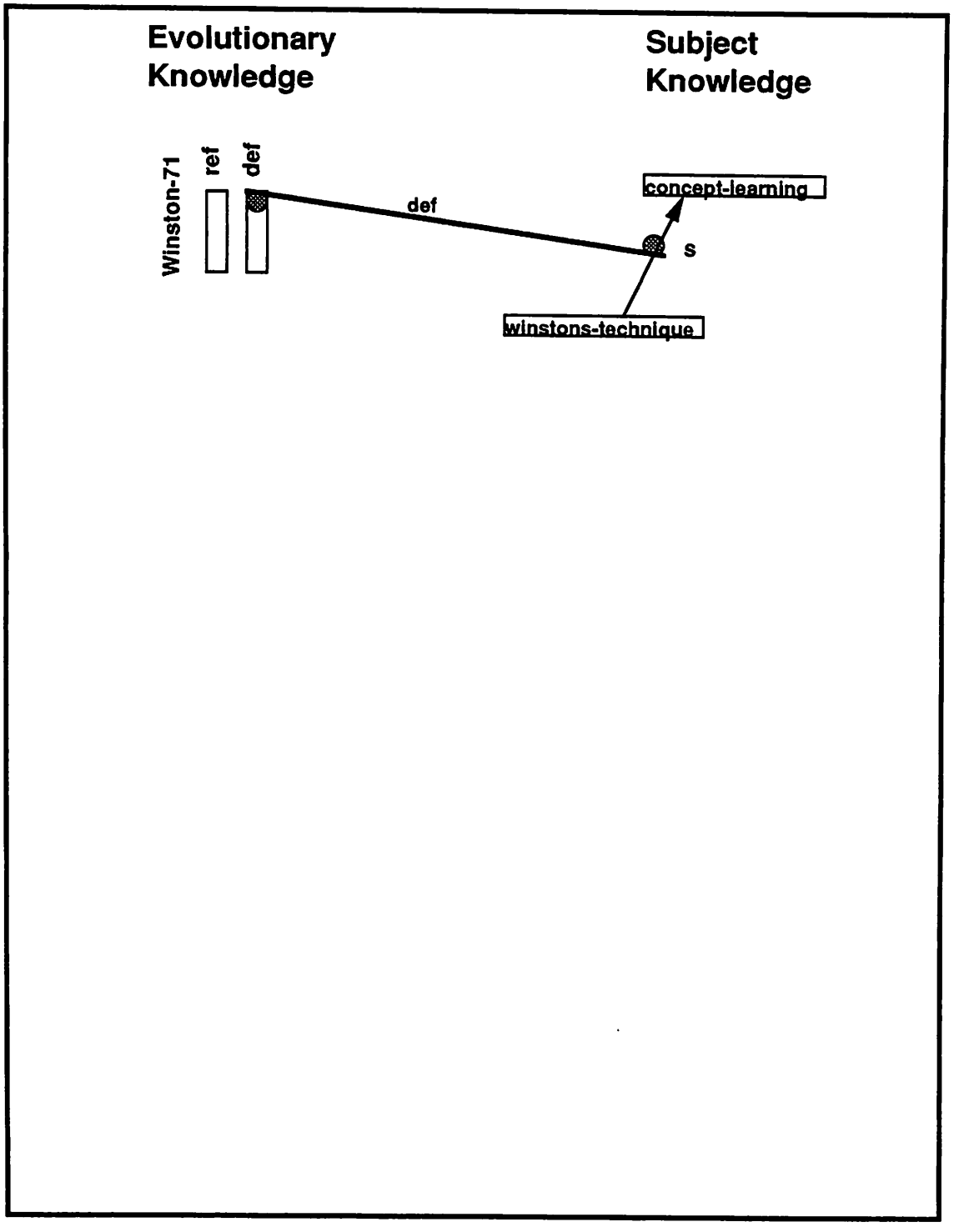


Figure 3.7: Memory after [Winston 71]

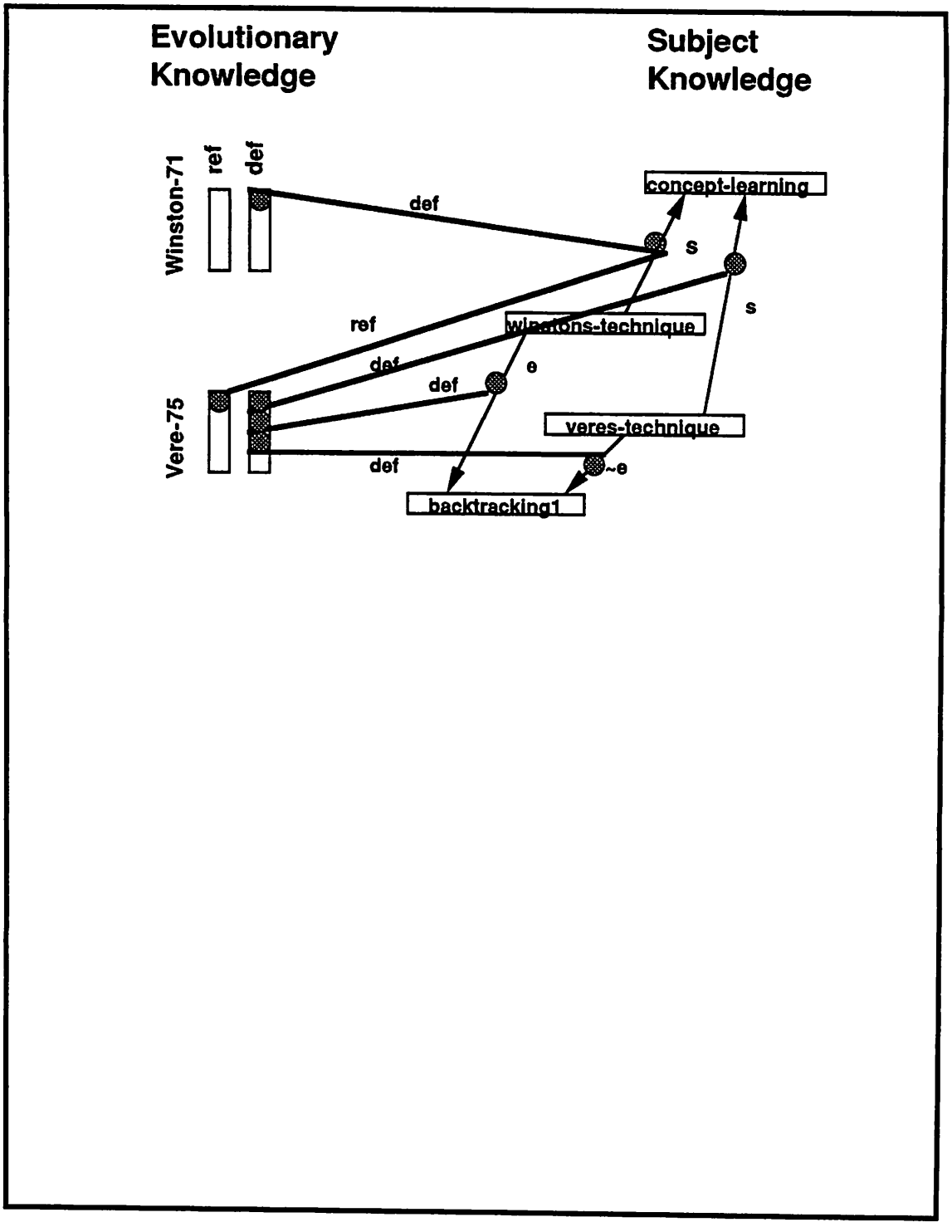


Figure 3.8: Memory after [Vere 75]

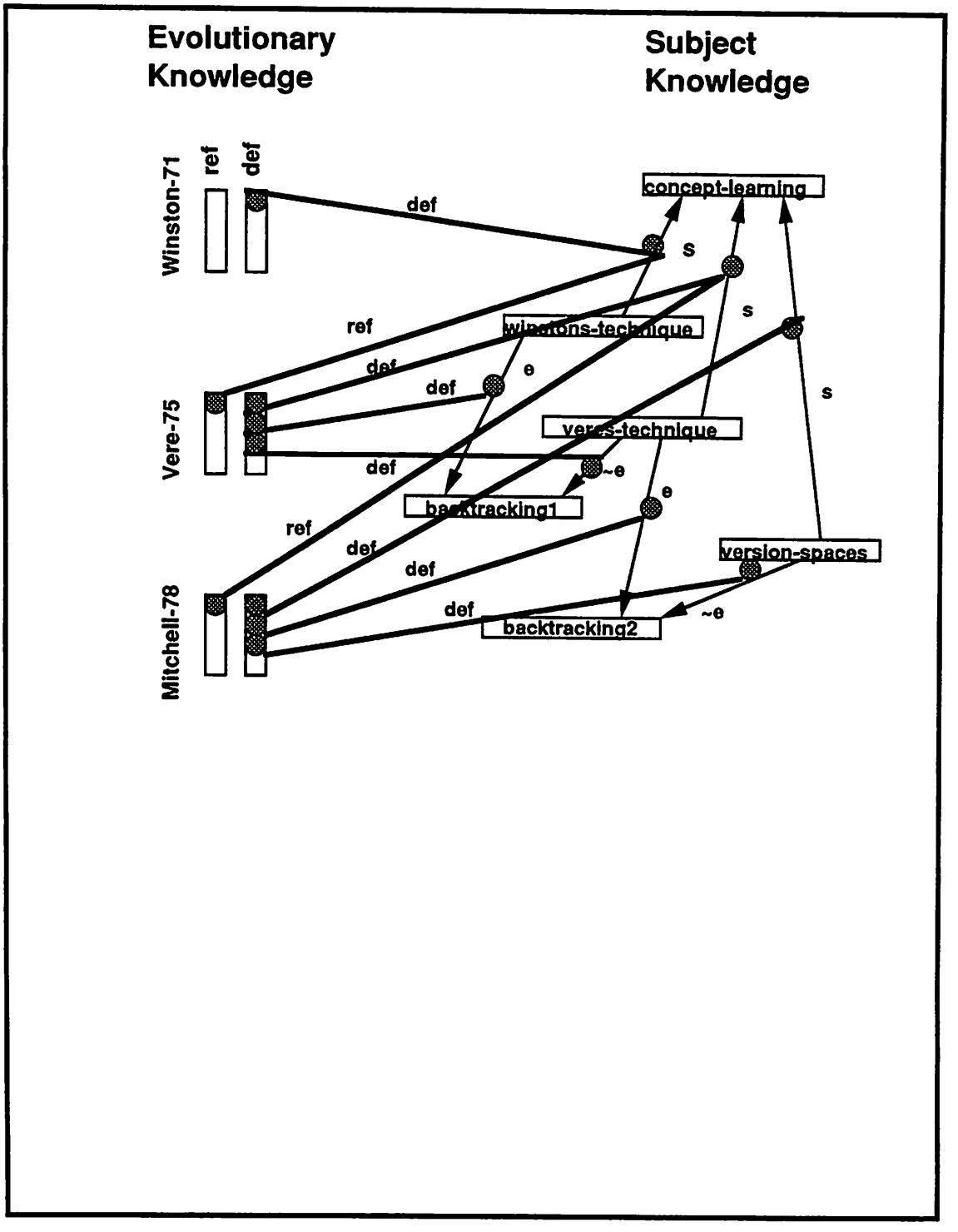


Figure 3.9: Memory after [Mitchell 78]

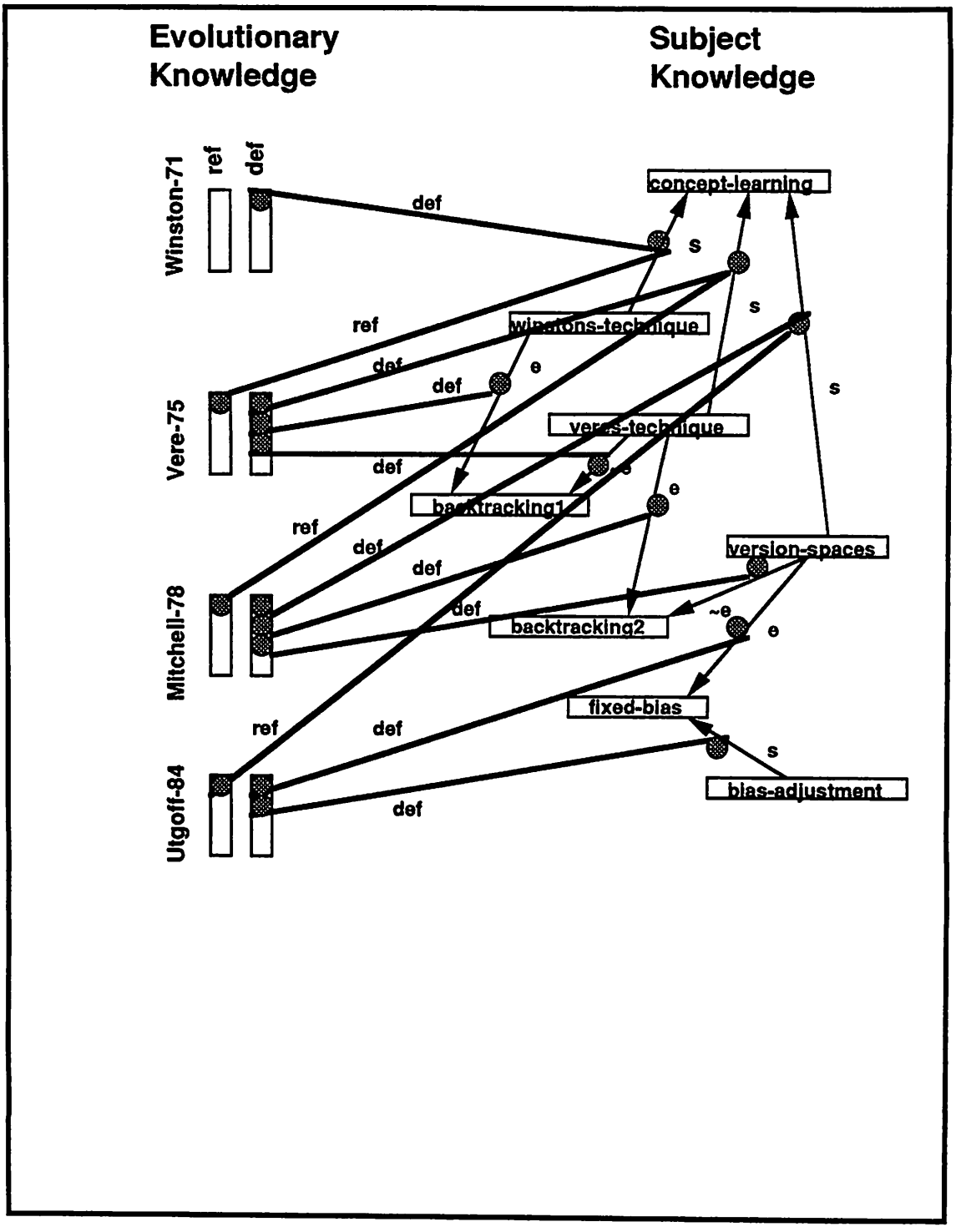


Figure 3.10: Memory after [Utgoff 84]

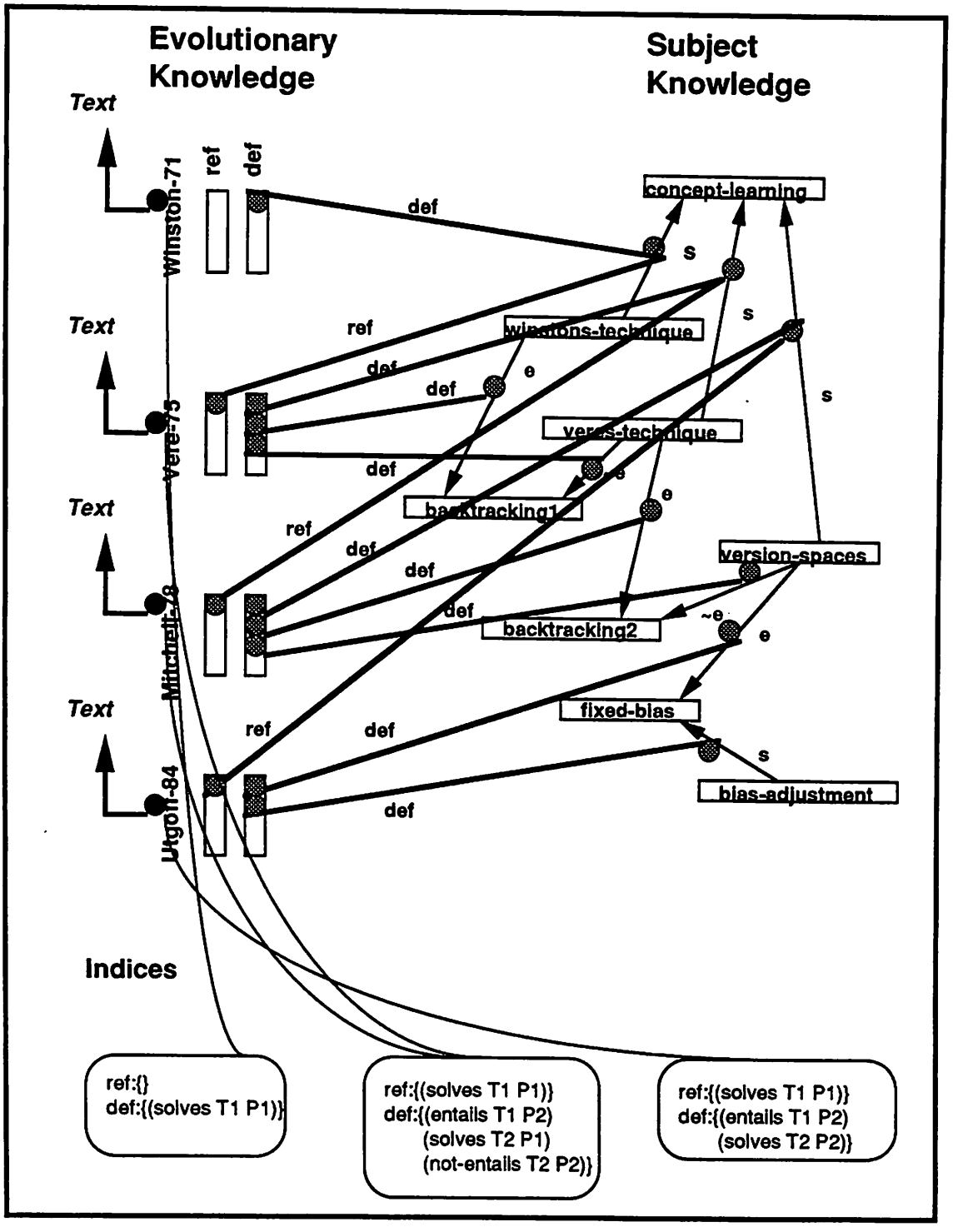


Figure 3.11: Overall Memory with Indices

- **Problem:** If the item is of type problem, then RA states where that problem came from, and lists any technique that solves it.
- **Technique:** If the item is of type technique, then RA states what problem that technique solves.
- **Concept:** If the item is of type concept, RA states what item's definition involves that concept.
- **Property:** If the item is of type property, then RA states what item exhibits that property.

In addition, RA lists the paper that first introduced that item into RA's memory. For example, version-spaces was introduced into RA's memory by the **def** of [Mitchell 78], and fixed-bias-problem was introduced into the memory by the **def** of [Utgoff 84]. Thus, selecting version-spaces 'retrieves' the episode called [Mitchell 78], and selecting fixed-bias-problem 'retrieves' the episode called [Utgoff 84]. The following are some examples of this kind of retrieval:

Select VERSION-SPACES
RA: VERSION-SPACES solves CONCEPT-LEARNING [MITCHELL-78].

Select FIXED-BIAS
RA: VERSION-SPACES entails FIXED-BIAS.
BIAS-ADJUSTMENT solves FIXED-BIAS [UTGOFF-84].

Select SEARCH-PROPERTY
RA: SBL exhibits SEARCH-PROPERTY [MITCHELL-81].
SEARCH-PROPERTY is "Inductive learning is search".

Select OPERATIONALITY
RA: EBL involves OPERATIONALITY [MITCHELL-86].

3.3.2 Retrieving the Semantic Relations from a Schema

When you select a paper (by clicking on an item such as Mitchell-78), RA generates a description of the schema of the paper. To generate this description, RA traverses the **ref** and **def** links from the schema in order to obtain the S-knowledge relations belonging to the schema. For example, selecting Mitchell-78 generates the following description:

[MITCHELL-78]: VERES-TECHNIQUE solves
CONCEPT-LEARNING [VERE-75].
In this paper, I show: VERES-TECHNIQUE entails BACKTRACKING2.
VERSION-SPACES solves CONCEPT-LEARNING. VERSION-SPACES
not-entails BACKTRACKING2.

In addition to this description, RA also retrieves the text attached to the instantiated schema of the paper and displays it on the text window (window *D* in Figure 3.1).

3.3.3 Retrieving a Paper on a Structural Index

This kind of retrieval is performed by the suggestions and the analogical summarization components. These components access the skeletal schemas which constitute the *structural* indices for papers. When a structural index is used, these components retrieve the papers that are indexed on that index. For example, when the suggestion component interprets a structural index as a heuristic rule, it retrieves all papers indexed on that index as examples of where such a research strategy has been used before. The retrieval of [Minksy & Papert 69] in Figure 3.4 illustrates this kind of retrieval.

In summary, the retrieval of papers (research episodes) based on semantic indices proceeds from the S-knowledge relations over **def** links to the instantiated schema of a paper. The retrieval of the S-knowledge relations from the instantiated schema of a paper proceeds from the schema through the **ref** and **def** links to the S-knowledge relations. Finally, the retrieval of papers on structural indices proceeds from the skeletal schema to the instantiated schema of the paper. Figure 3.12 illustrates the three kinds of retrieval. Table 3.3 contains a synopsis of the of the RA's retrieval capabilities.

3.4 Suggestions

When a user right-clicks on a node, this generates a command to the suggestion component: "I am interested in this item. Can you suggest any research directions involving this item?" The node selected by the user is called an *anchor* node. Given the anchor node, the Suggestion component generates a series of research suggestions involving that node. As we saw in the example dialog above, you may either accept or decline a suggestion. When you decline a suggestion, RA simply proceeds to the next suggestion. If you accept it, so far as RA is concerned, you have just written a paper. The **ref** of your paper is automatically obtained from the 'if' part of the heuristic rule (that was used to generate the suggestion), and RA prompts you for the new objects in the **def** of the paper. This section describes these operations in detail. Table 3.4 contains a synopsis of the suggestion component.

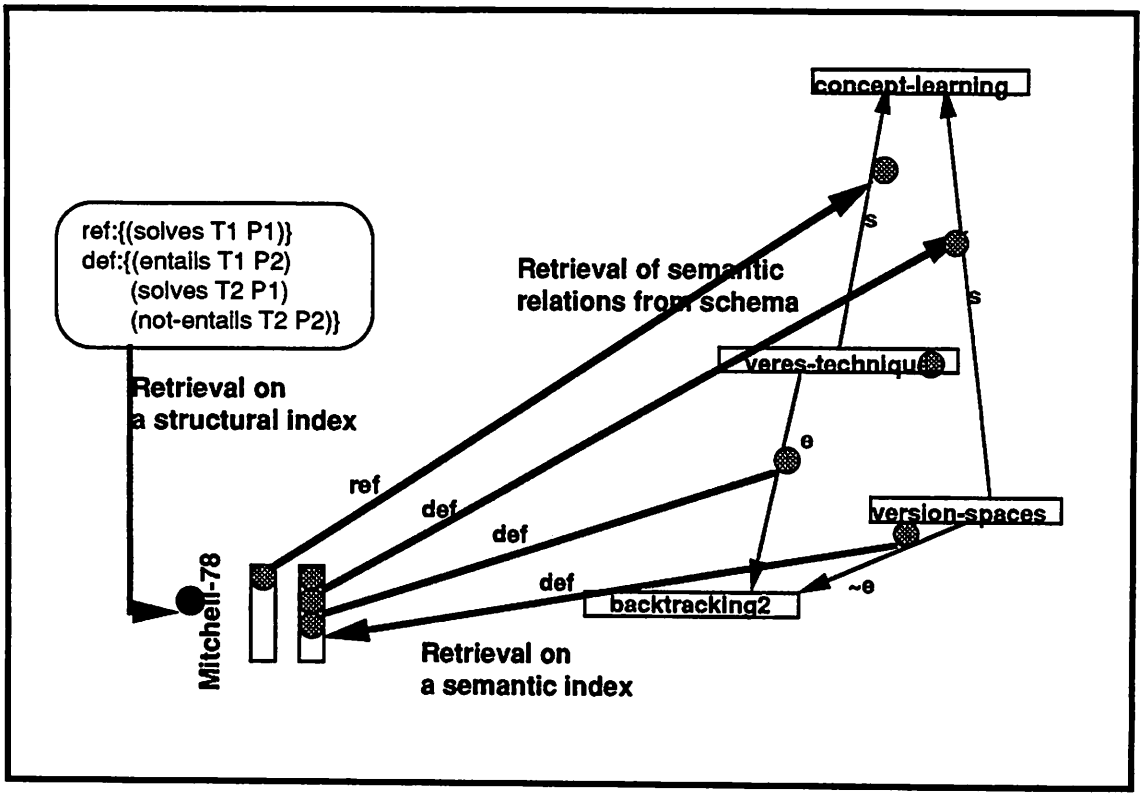


Figure 3.12: The Three Kinds of Retrieval in RA

- Retrieval on a semantic index: RA retrieves the research schema of a paper based on the subject knowledge defined by the paper, i.e., it uses S-knowledge to retrieve E-knowledge. This process traverses a **def** link from an S-knowledge relation to a research schema.
- Retrieving the S-knowledge: RA retrieves the subject knowledge defined by a paper based on its research schema, i.e., it uses E-knowledge to retrieve S-knowledge. This process traverses both **ref** and **def** links from a research schema to S-knowledge relations.
- Retrieval on a structural index: RA retrieves the instantiated research schema of a paper based on its research strategy, i.e., it uses structural indices to retrieve E-knowledge. This process traverses a link from the skeletal schema to the instantiated schema of the paper.

Table 3.3: Synopsis of RA's Retrieval Capabilities

3.4.1 Accessing Heuristics

As shown in Figure 3.11, the skeletal schemas are used as structural memory indices for papers. When RA is given an anchor node, RA accesses all these structural indices and begins to interpret them as heuristic rules, where the **ref** corresponds to the 'if' part of the rule, and **def** corresponds to the 'then' part of the rule. The 'then' part is used to generate a suggestion. After generating the suggestion, RA retrieves all the papers that are indexed on that structural index as examples of where that particular research strategy has been used before. In anthropomorphic terms, when you ask RA for a suggestion, RA tries to 'remember' the various research strategies it knows. When a particular strategy is applicable for your situation, it gives you a suggestion. Further, RA is also 'reminded' of the other papers that have the same strategy. These papers are thus 'retrieved' based on their research strategy.

3.4.2 Matching a Heuristic

The structural indices or heuristics are skeletal research schemas and hence have a **ref** part and a **def** part. RA interprets the **ref** part as an 'if' condition. Without loss of generality, assume that the anchor node is of type technique. The matcher collects all the variables of type technique in the **ref**. For each such variable, the matching process tries substituting the anchor node for that variable. Under this substitution, the matcher

checks if all the other variables in the **ref** are instantiable. If so, the heuristic is said to be 'applicable' for that substitution. In graph-theoretic terms, RA's knowledge-base as well as the **ref** of a heuristic are labelled graphs. The matcher attempts to find a subgraph in the knowledge-base that is isomorphic to the **ref** with the additional constraint that one of the nodes of this subgraph should be the anchor node[6].

As an example, let's first consider the skeletal schema of [Minsky & Papert 69] that was used to generate the suggestion in Figure 3.4 above. This schema is shown below:

ref: {(solves T1 P1)}

def: {(dominates P1 P2)
(dominates P1 P3)
(solves T1 P2)
(not-solves T1 P3)
(entails T1 P3)
(not-solves T1 P1)}

The anchor node chosen was EBL. In seeing if this rule is applicable, the matcher finds that there is only one variable of type technique in the **ref**. It tries a substitution of EBL for this variable, obtaining (solves EBL P1). To find instantiations for the other variables in the **ref**, it follows the links from the EBL node. In this case, there is a solves relation that links EBL to learning-problem, so RA substitutes learning-problem for EBL. Since all the variables in the **ref** are now instantiated, this rule is applicable for the following substitution: {(T1/EBL) (P1/learning-problem)}.

Let's now consider a slightly more complicated example that led to the suggestion in Figure 3.6. This suggestion was generated from the following heuristic[7]:

ref: {(encapsulates T1 T2)
(instantiates T3 T2)
(exhibits T3 Pr1)}

def: {(R T1 Pr1)}

Referring to the dialog above (Section 3.1.1), the object super-duper-hybrid-technique was defined as an encapsulation of two components component1 and component2, which were defined (respectively) as instantiations of EBL and SBL. With super-duper-hybrid-technique as the anchor node, the matching process proceeds as follows. There are three

[6]In fact the anchor node is called the anchor node because we can visualize this matching process as follows: The graph corresponding to a **ref** is overlaid on top of the graph that corresponds to RA's S-knowledge. Suppose the anchor node is of type *technique*. Then each variable in the **ref** of type technique is (in turn) *anchored* on the anchor node and the **ref** graph is moved around until it is isomorphic to some subgraph of the S-knowledge graph. If such a subgraph is found, then the heuristic is applicable.

[7]This is the schema of [Hirsh 88]. See Section 4.4.5.

technique nodes in **ref**: T1, T2 and T3. Let's consider a matching of the anchor node against T3. With a substitution of super-duper-hybrid-technique for T3, the matcher tries to find a way to satisfy the relation '(exhibits super-duper-hybrid-technique Pr1)'. This fails because there is no exhibits relation emanating out of that node. However, the substitution of the anchor node for T1 succeeds: (encapsulates super-duper-hybrid-technique T2) obtains a substitution of component2 for T2; (instantiates T3 component2) obtains a substitution of SBL for T3; (exhibits SBL Pr1) obtains a substitution of search-property for Pr1. Hence this heuristic is applicable for the following substitution: {(T1/super-duper-hybrid-technique) (T2/component1) (T3/SBL) (Pr1/search-property)}. Figure 3.13 illustrates this matching process.

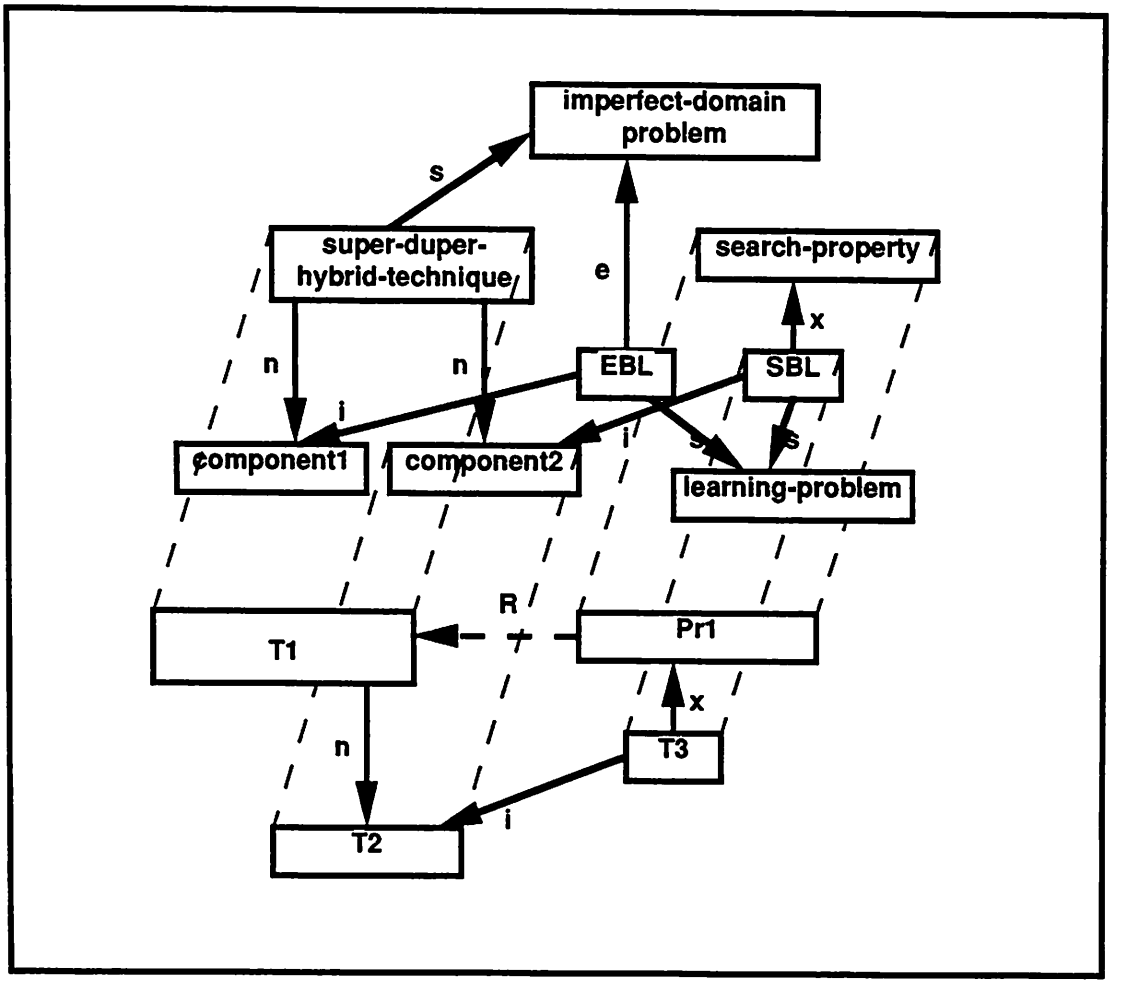


Figure 3.13: Matching A Heuristic

3.4.3 Generating a Suggestion

As we saw above, a heuristic is applicable if all the variables in the **ref** are instantiable in the neighborhood of the anchor node (i.e., with one of the variables in the **ref** instantiated to the anchor node). When a heuristic is applicable, then the substitution that satisfies the **ref** of the heuristic is first carried over into the **def**. As an example, let's consider the heuristic of [Minsky & Papert 69] above. The substitution {(EBL/T1) (learning-problem/P1)} is carried into the **def** to obtain the following set of relations for the **def**:

{(dominates learning-problem P2)
(dominates learning-problem P3)
(solves EBL P2)
(not-solves EBL P3)
(entails EBL P3)
(not-solves EBL learning-problem)}

These relations are used to generate a suggestion. RA has a rudimentary ability for language generation using 'canned' sentences. For example, the above set of relations will be used to generate the following suggestion:

Can you find P2, P3 such that: LEARNING-PROBLEM dominates P2,
LEARNING-PROBLEM dominates P3, EBL solves P2,
EBL not-solves P3.

However, several of RA's schemas have been 'demo-ized' and have been provided with a touched-up English suggestions. For the above schema, RA generates the following suggestion:

One way to look for limitations of EBL is to
think if EBL really solves the class of
problems LEARNING-PROBLEM. Can you identify
subclasses of LEARNING-PROBLEM that EBL does
not solve?

3.4.4 Accepting a Suggestion

After displaying a suggestion, RA gives you the option of either accepting or declining the suggestion. If you decline the suggestion, RA discards that suggestion and tries other substitutions for the same heuristic or moves to the next heuristic to generate another suggestion. If you accept a suggestion, in RA's eyes, you've just written a new paper. For this paper, RA already knows the **ref**: i.e., the set of substitutions that satisfied the **ref** of the skeletal schema is assumed to be the **ref** of your new paper. For any uninstantiated variables in the **def**, RA asks you to create new objects to which these variables can be

instantiated. In the example shown above, P2 and P3 are the uninstantiated variables for which RA asks the user for new values (Figure 3.4). With these new values, RA creates a schema for your paper and indexes this schema under the same skeletal schema from which the suggestion was generated. For example, in the suggestion above, we provided RA with perfect-domain-problem for P2 and imperfect-domain-problem for P3. There are no more variables to be instantiated, so RA constructs the following schema for the new paper:

ref: {(solves EBL learning-problem)}

def: {(dominates learning-problem perfect-domain-problem)
(dominates learning-problem imperfect-domain-problem)
(solves EBL perfect-domain-problem)
(not-solves EBL imperfect-domain-problem)
(entails EBL imperfect-domain-problem)
(not-solves EBL learning-problem)}

This schema is now indexed on the same skeletal schema as [Minsky & Papert 69].

In summary, the Suggestion component accesses the skeletal schemas (i.e., structural indices) in memory and interprets them as heuristic rules. The **ref** part of these schemas is treated as the 'if' condition and the **def** part is used to generate a suggestion. A heuristic is applicable if RA can find a set of instantiations for all the variables in the 'if' part of the rule, with one of these variables instantiated to the anchor node. When such an instantiation is found, RA carries these instantiation into the 'then' part of the rule and generates a suggestion. If the user accepts the suggestion, then RA prompts the user to create new objects as instantiations for the uninstantiated variables in the 'then' part of the rule. This results in a new schema being created for the user's new 'paper'; this schema is now indexed in memory on the same skeletal schema from which the suggestion was generated. Table 3.4 contains a synopsis of the suggestion component.

3.5 Chronological Summary

The two functionalities, chronological and analogical summarization, were motivated by the phrase 'research trend.' The Webster's dictionary defines 'trend' as follows: (1) a line of general direction or movement, and (2) a prevailing tendency or inclination. Normally both these senses of the word 'trend' apply to the phrase 'research trend.' We can talk of the research trends that led to EBL as well as the current trends in EBL. In the former case, we are referring to a chronological progression of papers that led to EBL, and in the latter case, we are referring to the analogical relationship among a set of papers pertaining to EBL. It was noticed that the representation of research papers in terms of **ref** and **def** was ideally suited for generating summaries of papers in both chronological

- The user selects an *anchor* node to ask RA to generate research suggestions involving that node.
- RA accesses the structural memory indices, which are also skeletal research schemas, and interprets them as 'if-then' heuristics. The **ref** part of the skeletal schema is used as an 'if' condition and the **def** part is used for generating a suggestion.
- A heuristic is applicable if the anchor node can be substituted for some variable in the **ref**, and under this substitution, there is some substitution for every other variable in the **ref**.
- When a heuristic is applicable, the substitutions of the **ref** are carried into the **def**. The relations in the **def** now constitute a suggestion.
- If the user accepts the suggestion, RA asks the user to name the uninstantiated variables in the **def**.

Table 3.4: Synopsis of the Suggestion Component

and analogical terms. This section describes the chronological summarization component, and the next section describes the analogical summarization component. Table 3.5 contains a synopsis of the chronological summarization component.

The chronological summarization component is accessed from the hypertext interface (Figure 3.1) by typing a command in the command window (window B). The command can take one of the following forms:

1. (csummary paper): The events that led to the paper are described.
2. (csummary paper1 paper2): The chronological relationship between the two papers is described.
3. (csummary node): First the paper that introduced that node in its **def** is accessed. Then the events that led to the paper are described as in (1) above.
4. (csummary node1 node2): First the papers that introduced these nodes in their **def** are accessed. Then the chronological relationship between these two papers is described as in (2) above.

Since (3) and (4) are trivial extensions of (1) and (2), I will describe (1) and (2) below.

1. (csummary paper): When asked to provide a chronological summary of a paper, RA accesses the schema of the paper. This schema is added to a summarization list called C-List. Then RA traces through the **ref** of the paper to locate the S-knowledge relations that are linked to the **ref**. From these S-knowledge relations, RA traverses the **def** links to locate the schemas that introduced each of these relations into RA's memory. These schemas are added to the beginning of the C-List. There are two possible cases: in the simple case, each schema in the list refers to exactly one paper prior to it, and the schemas form a linear chain. In this case, the number of generations of schemas added to C-List is arbitrarily set to 2. Consider the examples shown in Figure 3.11. Suppose the user wanted a 'csummary' of [Utgoff-84]. First, this schema is added to C-List. This paper has only one **ref** link, and traversing this link takes us to the relation (solves version-spaces concept-learning). From this relation, we traverse a **def** link and locate the schema of [Mitchell 78] in order to identify the schema (episode) that introduced this relation into RA's memory. Thus the schema of [Mitchell 78] is added to the beginning of the C-list. Now we traverse the **ref** link of this schema to reach the relation (solves veres-technique concept-learning). Traversing the **def** link from this relation, we reach the schema of [Vere 75]. This schema is added to the beginning of the C-List. Since each paper in the C-List added exactly one paper from a previous generation, we have a linear list of papers, and the search is cut off at this point since we have reached a limit of two generations.

Once the traversal process is completed, RA generates a description of the schemas in the C-List. For instance, the above example results in the following summary:

WINSTONS-TECHNIQUE solves CONCEPT-LEARNING.
[VERE-75] showed: WINSTONS-TECHNIQUE entails BACKTRACKING1.
VERES-TECHNIQUE solves CONCEPT-LEARNING.
VERES-TECHNIQUE not-entails BACKTRACKING1.
[MITCHELL-78] showed: VERES-TECHNIQUE entails
BACKTRACKING2. VERSION-SPACES solves CONCEPT-LEARNING.
VERSION-SPACES not-entails BACKTRACKING2.
[UTGOFF-84] showed: VERSION-SPACES entails FIXED-BIAS.
BIAS-ADJUSTMENT solves FIXED-BIAS.

The more complicated case occurs if a schema in C-List refers to more than one paper. Without loss of generality assume that a schema refers to two different papers: i.e., a schema has two **ref** links to two S-knowledge relations, and following the **def** links from these relations takes us to two different schemas. In this case, each of these two schemas is traversed separately (adding all the encountered schemas to the C-List) until both paths converge at some schema. At this point, the traversal is terminated, and a description of the schemas in the C-List constitutes a chronological summary. See Appendix B for a more involved example.

2. (csummary paper1 paper2): When given two papers, RA attempts to find a chronological relationship between the two. This proceeds as follows: Without loss of generality, assume that paper1 is the chronologically earlier one, and paper2 is the later one. RA starts with the schema of the later paper, paper2, and conducts a depth-first search through the **ref** of the paper until the schema of the earlier paper, paper1, is reached. First, the S-knowledge relations in the **ref** of paper2 are collected. For the first relation in the **ref**, RA traverses a **def** link to locate the schema (episode) that introduced that relation; for the first relation in the **ref** of this latter schema, RA locates the schema that introduced that relation and so on. If this results in arriving at the schema of paper1, then all the schemas encountered during the process are added to a C-List, and summarized as above. However, if a predefined depth limit of 5 is reached, then RA backtracks and tries a depth-first search on the subsequent S-knowledge relations until either a path connecting the two schemas is found, or there is none within a depth-limit of five[8].

In terms of the examples in Figure 3.11, each paper refers to only one previous paper, and hence this search is trivial. When given the command (csummary Winston-71 Utgoff-84) RA starts with the schema of [Utgoff 84] searching backward through the **ref** links until it reaches the schema of [Winston 71]. The same summary is generated as with the previous case. Appendix B describes a more involved example.

[8] [Biw] The reason why RA performs a uni-directional search from the schema of paper2 rather than a bi-directional search from the schemas of both paper1 and paper2 is as follows: Each S-knowledge relation has exactly one **def** link linking it to the schema that introduced it, whereas a relation can have several **ref** links to all the schemas that reference it. Thus the branching factor is much smaller in the backward direction than in the forward direction. Hence the choice of a unidirectional search backward in time rather than a search forward or a bi-directional search.

In summary, RA's representation of research papers in terms of **ref**, **def** pairs supports the chronological summarization task in a natural way. Even though this task is accomplished by traversal algorithms that are quite trivial in nature, these algorithms illustrate the strong interplay between RA's subject knowledge (i.e., its knowledge of machine learning) and RA's evolutionary knowledge (i.e., its knowledge about the schemas of the papers). Table 3.5 contains a synopsis of the chronological summarization component.

- When asked to state the chronological 'trends' that led to some item, RA finds the paper that defined that item; for example, RA finds [Mitchell 78] as the paper that defined version-spaces. It goes two generations into the past by tracing through **ref** links. All the papers visited in this process are summarized.
- When asked to find the chronological connection between two papers, it conducts a depth-first search through **ref** links from the later paper until the earlier paper is reached. All the papers visited in this process are summarized.

Table 3.5: Synopsis of the Chronological Summarization Component

3.6 Analogical Summary

Two kinds of analogical summaries are considered in this work. The first kind of summary may be called *structural* analogy, and the second kind may be called *semantic* analogy. When given two papers and asked to find an analogy, RA first tries to find a structural analogy. If that fails, it tries a semantic analogy. Table 3.6 contains a synopsis of the analogical summarization component.

These two kinds are illustrated with the examples from Figure 3.11 above. Suppose RA is asked to find an analogy between the papers [Vere 75] and [Mitchell 78]: (asummary Vere-75 Mitchell-78). First RA accesses the schemas of these two papers. From the schemas, it traverses back to find their structural index. In this case, both these papers are indexed on the same structural index, i.e., they have the same skeletal schema shown below:

ref: {(solves T1 P1)}

def: {(entails T1 P2)
(solves T2 P1)
(not-entails T2 P2)}

Since these papers are structurally identical, a structural analogy exists between them. Now RA attempts to find a maximal instantiation of this skeletal schema until it fails to describe either paper. For example, the instantiated schema of [Vere 75] and [Mitchell 78] are shown below:

ref: {(solves winstons-technique concept-learning)}

def: {(entails winstons-technique backtrcking1)
(solves veres-technique concept-learning)
(not-entails veres-technique backtracking1)}

ref: {(solves veres-technique concept-learning)}

def: {(entails veres-technique backtracking2)
(solves veres-technique concept-learning)
(not-entails veres-technique backtrcking2)}

To find a maximal instantiation of the skeletal schema to reflect the analogy between the two papers, RA considers each variable in the skeletal schema and sees if the corresponding constant in both the instantiated schemas is the same. If so, the variable is replaced by that constant. In the example above, only P1 is instantiated to the same constant (i.e., concept-learning) in both the schemas, while the others are instantiated to different constants in the two schemas. Hence RA has found the following analogy between the two papers:

ref: {(solves T1 concept-learning)}

def: {(entails T1 P2)
(solves T2 concept-learning)
(not-entails T2 P2)}

The above analogy is described as follows:

> The analogy between [VERE-75] and [MITCHELL-78] is:
> Given T1 solves CONCEPT-LEARNING, both papers show: T1 entails P2.
> T2 solves CONCEPT-LEARNING. T2 not-entails P2.

In English, this summary is equivalent to the following: Given a technique to solve the concept learning problem, both papers show that their respective techniques have each an emergent problem, and propose a new technique to solve the concept learning problem, while avoiding their respective emergent problems.

A weaker form of structural analogy is attempted even if the two papers don't have the same structural index (i.e., skeletal schema) but the skeletal schemas have the same **ref** and have at least one common relation in the **def** that shares a variable with the **ref**. For example, if RA is asked to provide an analogy between [Mitchell 78] and [Utgoff 84], RA retrieves their skeletal schemas shown below:

ref: {(solves T1 P1)}

def: {(entails T1 P2)
(solves T2 P1)
(not-entails T2 P2)}

ref: {(solves T1 P1)}

def: {(entails T1 P2)
(solves T2 P2)}

The **ref** of these two skeletal schemas is identical, and there is one relation common to the two **def**s that shares a variable (T1) with the **ref**[9]. As before, RA tries to find a maximal instantiation for the common parts of the two schemas and finds that P1 is instantiated to concept-learning in both schemas. After this instantiation, we have a common part between the two skeletal schemas and a part that is different. The common part is the **ref** plus the one common relation in the **def**. The part that is different in [Mitchell 78] is the following: {(solves T2 concept-learning) (not-entails T2 P2)}. The part that is different in [Utgoff 84] is the following: {(solves T2 P2)}. These are used to generate the following summary:

> The analogy between [MITCHELL-78] and [UTGOFF-84] is:
> Given T1 solves CONCEPT-LEARNING, both papers show: T1 entails P2.
> [MITCHELL-78] shows: T2 solves CONCEPT-LEARNING.
> T2 not-entails P2.
> [UTGOFF-84] shows: T2 solves P2.

In English, this summary is equivalent to the following: Given a technique to solve the concept learning problem, both papers show that their respective techniques have

[9]This restriction is to ensure that the common relation in the **def** is not any arbitrary one, but is at least connected to the **ref**.

each an emergent problem. [Mitchell 78] proposes a technique to solve the original problem while avoiding the emergent problem, whereas [Utgoff 84] proposes a technique to solve the emergent problem.

If RA fails to find this kind of analogy as well, then it resorts to a weaker form of semantic analogy. The instantiated schemas of the two papers are retrieved and compared to see if they have any constants in common. Hypothetically, assume that RA failed to find the above structural analogy between [Mitchell 78] and [Utgoff 84]. When RA attempts a semantic analogy between the two, it finds that they have two constants in common among all the relations in their **refs** and **defs**, namely, concept-learning and version-spaces. So RA generates the following (rather wimpy?) summary:

> The analogy between [MITCHELL-78] and [UTGOFF-84] is:
> Both papers are concerned with: CONCEPT-LEARNING, VERSION-SPACES.

Similarly, if RA is asked to find an analogy between [Winston 71] and [Utgoff 84], it finds that they both have one constant, concept-learning in common. It generates the following summary:

> The analogy between [WINSTON-71] and [UTGOFF-84] is: Both papers are concerned with: CONCEPT-LEARNING.

In Summary, the analogical summarization component attempts to find something in common between the two papers it is asked to summarize. If they have the same skeletal schema (the strongest analogy) RA finds a maximal instantiation of this schema that captures both the papers. If not, RA looks for a weaker form of structural analogy, i.e., if their **refs** are (structurally) similar, and if they have at least one common relation in **def** that is connected to the **ref**. If that also fails, RA sees if there is at least a semantic similarity, i.e., whether the two schemas have any constants in common. Table 3.6 contains a synopsis of the analogical summarization component.

3.7 Some Refinements

This section discusses several refinements to the basic mechanisms described in the previous sections. These refinements were planned, but never implemented. Since these refinements will be both important and useful in a real and practical Computer-Aided Research systems, they are discussed here.

Discourse History: The current implementation of RA has no sense of discourse. By maintaining a discourse history, i.e., a history of the papers that the user has read, RA can refine several of its functionalities:

1. When RA is asked to generate a chronological summary to describe a paper, currently the summarization component goes two generations into the past. Suppose

- When two papers have the same skeletal schema, RA finds a maximal instantiation of their skeletal schema as the analogy between the two papers.
- When two papers don't have the same skeletal schemas, if their skeletal schemas have the same **ref** plus one common relation in **def** that shares a variable with **ref**, RA finds a maximal instantiation of the two skeletal schemas as the analogy between the two papers.
- If neither of the above holds, RA looks for common constants in the instantiated schemas of the two papers. The term analogy is somewhat stretched in this case.

Table 3.6: Synopsis of the Analogical Summarization Component

the discourse history shows that the user has recently accessed a set of related papers. Using this information, RA can generate a summary (going back as many generations as needed) until it touches upon the set of papers that the user has recently accessed.

2. Suppose a user selects some paper from the hypertext interface (for a quick description and text). Suppose this paper has the same skeletal schema as one of the papers that the user had recently seen. RA can come up with, "By the way, there is an analogy between this paper and the one you recently saw."

3. Suppose the discourse history shows that the user has accessed a set of papers that have strong chronological connections (i.e., a sequence of papers that form a linear chronological chain) in a hap-hazard manner by seeing several unrelated papers in the middle. RA can generate a spontaneous chronological summary to say, "You may not have realized it, but these papers are all strongly connected to each other."

To support the above refinements, all that RA needs in the form of a discourse history is a simple linear list of the papers and nodes visited by the user.

Summarizing a corpus: The summarization components take two papers and attempt to find chronological or analogical relationships between the two. In principle, there is no reason why this cannot be extended to a corpus of papers. Providing some trivial corpus selection mechanism can make these components look quite impressive. Suppose we can select a corpus of papers on one of several criteria, such as all papers by

a particular author (e.g., Gerry DeJong), all papers published in a particular conference (e.g., AAAI-88) or journal, all papers on a particular system (e.g., SOAR) and so on. Depending on the constitution of the corpus, the system can automatically choose the kind of summary to generate. Presumably, a chronological summary best captures the set of papers published by an author, whereas an analogical summary best captures the set of papers in a conference. For example, a general command like (summary (corpus-select DeJong)) should provide to the summarization component a list of all papers ever written by DeJong, so that RA can summarize the entire research history of an author!

Organizing the order of suggestions: In building RA, I was most interested in showing that a single memory organization can support a number of different functionalities. While the level of abstraction used in RA is able to support its various functionalities, it is inadequate to capture the inherent 'interestingness' of the various papers and their research strategies (more on this in Chapter 7). Hence, the current implementation has no notion of interestingness and generates all possible suggestions in an essentially arbitrary order. While no serious solution to this problem is believed possible at RA's level of abstraction (without infusing RA with lot more deep-semantic knowledge, See Chapter 7), some simple-minded solutions can be used to organize the order in which RA generates its suggestions. A schema that has several relations in its **ref** has a stringent applicability condition: such a schema tends to be applicable only in a small number of cases, but when it does, it tends to give a suggestion encompassing objects that are widely separated in the memory (the schema of [Hirsh 88] in Section 4.4.5 is an example). On the other hand, a schema that has several relations in its **def** tends to give a very specific suggestion (the schema of [Minsky & Papert 69] is an example). Organizing the order of the suggestions based on the specificity of the schema's applicability condition as well as the specificity of its suggestion will enable RA to generate those suggestions that are likely to be more interesting before those that are likely to be less interesting.

In Summary, some simple refinements to the various capabilities can provide a powerful Computer-Aided Research system. In such a system, each of the functionalities, in isolation, will be quite simple, but the combination of the various functionalities in a single system can provide a powerful tool.

"Mucho trabajo," they all responded in chorus. "A lot of work."

—Thor Heyerdahl, The RA Expeditions.

3.8 Summary

RA provides several capabilities that together constitute a Computer-Aided Research system. All of these capabilities are supported by a single unified memory organization

for research papers in terms of research schemas. A single user interface allows access to all of RA's functionalities.

In this chapter, I first discussed the novel idea of Computer-Aided Research and described how it is realized in RA. This was illustrated by revisiting the example dialog introduced in Chapter 1. The next several sections described each of RA's functionalities. Although the algorithms used to achieve these functionalities are quite simple, they illustrate the strong interplay between RA's subject knowledge (i.e., knowledge of machine learning) and its evolutionary knowledge (i.e., knowledge of the papers of episodes through which the subject knowledge accrued). Finally, I discussed some possible refinements to RA's functionalities on the road to implementing practical computer-aided research systems.

> *"The rubber raft is meant to give everyone a feeling of security. This is nothing but a scientific experiment."*
> *"Come on, where's the saw? What's the good of something we will never use?" Santiago insisted provocatively.*
>
> —Thor Heyerdahl, The RA Expeditions.

Chapter 4

RA II: Acquisition of Research Schemas

I wanted to find out if a reed boat would be able to sail even farther, even from one continent to another.

—Thor Heyerdahl, The RA Expeditions.

The previous chapters described RA I, the first version of the RA program. The research schemas of RA I were handcoded into the system. In order to acquire RA I's research schemas automatically, a second version of the system, called RA II, was built. Several results have emerged from this effort.

Not only can RA II acquire research schemas, it can do so without extra representational baggage. Put another way, research schemas are an adequate representation with respect to yet another functionality, the acquisition of new research schemas.

Despite certain similarities, RA II's learning is quite different from the explanation-based learning (EBL). In EBL, an explanation is constructed by using general (uninstantiated) knowledge of the domain. In RA II, the **ref** of a schema is obtained as the most specific (instantiated) knowledge in memory that connects the various objects in the **def**. Hence, RA II's learning may be seen as a memory based learning technique as opposed to EBL which may be seen as a knowledge-based learning technique.

RA II's generalization strategy is also different. Since RA II does not use domain theoretic rules to construct an explanation, it would appear that it has no basis for generalization. In Chapter 5, I will justify RA II's generalization strategy in terms of basic-level theories of categorization. Table 4.1 contains a guide to Chapter 4.

Section	On first reading	Description
1	read	Motivation. Describes why RA II was built.
2	read	Introduces terminology. Using a set of common sense examples, describes what RA II learns.
3	skip	Describes how RA II works. The mechanisms are simple. Table 4.4 contains a synopsis.
4	skim	Describes several examples. Read through a couple. Particularly the example in 4.4.3.
5	skim	Discusses two known deficiencies of RA's learning strategy. Read 4.5.2.
6	skim	Summary

Table 4.1: Guide to Chapter 4

4.1 RA II: Motivation

Why a second Ra? Why was I beginning a thick expedition diary from page one again? Could I answer?

—Thor Heyerdahl, The RA Expeditions.

RA I proposes a memory organization that integrates the subject and evolutionary knowledge of a scientific domain. This memory organization hinges on the crucial notion of research schemas. In RA I, research schemas act (1) as a descriptive representation for papers, (2) as intentional rules of action, and (3) as memory indices to index papers with similar research strategies. The functionalities of RA I were implemented in order to illustrate these different uses of research schemas.

Then a question arose: if this memory model is so general, can it support other capabilities not originally planned for? In exploring this question, two additional tasks were proposed. One task was to use research schemas as a mechanism for processing the text of research papers. I never did complete this task, although some progress was made. The idea of conceptual references that came out of that work is described briefly in Chapter 1 (Section 1.3.2) and also in Chapter 7 (Section 7.2). The second task was the automatic acquisition of research schemas. The RA II system was built for that purpose. This chapter describes RA II's learning strategy, and the next chapter contains an analysis of this strategy.

4.2 What Does RA II Learn?

Given the **def** of a paper, RA infers its **ref**[1]. This **ref**, **def** pair constitutes the paper's research schema. This schema is then turned into a skeletal schema, which is used both as a research heuristic and as a memory index. More abstractly, when RA is given some new information, it infers how this information relates to the current ontology. The new information and its relation to the current ontology characterize the knowledge-evolution event. This event is then generalized (skeletized); the generalized event is used as a heuristic rule and as a memory index.

Since research schemas are used in three different ways (as a representation, as research heuristics, and as memory indices), there are three different perspectives on what RA learns. In what follows, I will describe the learning strategy mostly from the perspective of research schemas as a representation for research papers, and slide to the other perspectives when necessary.

4.2.1 Why Call It Learning?

Given the **def** of a paper, RA infers its **ref**, and thus 'learns' the schema of the paper. On a first glance, you may wonder why the process of finding the **ref** of a paper is characterized as learning. To understand this, refer to Figure 4.1. The input to RA is a sequence of papers: each paper is a set of semantic relations that reflect the new knowledge that is contributed by the paper, with no reference to research schemas. As expected, RA would construct a semantic memory that links all these input relations into a single network. This is shown on the right hand side of the figure. However, in addition to the semantic memory (which is also RA's S-knowledge), RA also acquires the E-knowledge store which sees each paper as a **ref**, **def** pair. The generalization of these **ref**, **def** pairs results in the papers' research strategies. Thus the input to RA is simply the contribution of each paper, with no reference to its research strategy. From this input, RA acquires the paper's research strategy. Therefore RA automatically *learns* the research strategies in the domain given only the contributions of a sequence of papers.

4.2.2 Research Schemas: Conventions and Terminology

All research schemas obey the following two conventions: (1) Both **ref** and **def** contain only relations and not objects, and (2) There is always at least one relation in **ref**[2]. The first convention is used for convenience at the implementation level — a research schema

[1] In the rest of Chapter 4, RA = RA II.

[2] Except those papers that are the first in getting a knowledge base off the ground, such as [God 08] and [Winston 71].

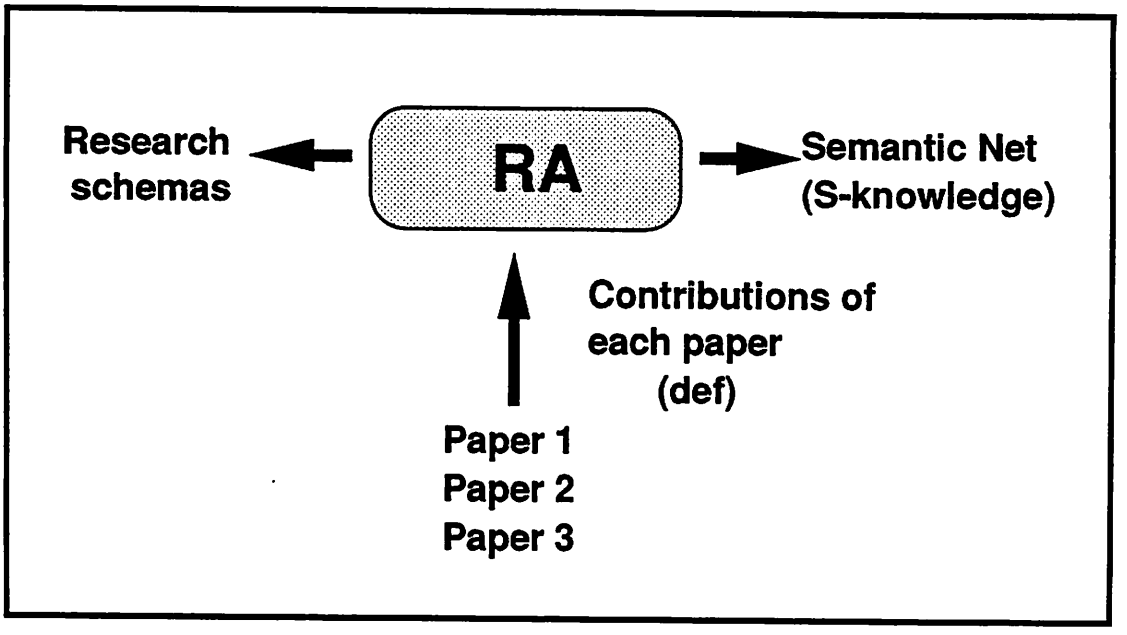


Figure 4.1: Why Is This Learning?

is simply a set of pointers to S-knowledge relations. The second assumption ensures that every research schema has an attachment point to the rest of the knowledge base, which in turn ensures that every heuristic rule has an applicability condition.

To make our discussion easier, Table 4.2 introduces some terminology. Since these new terms are important for following the rest of the discussion, let me solidify them through an example. Figure 4.2 shows part of the knowledge-base before the assimilation of the paper '[Rajamoney 88].' What is shown is therefore the pre-ontology of this paper. This paper has the following 'canonical abstract':

> EBL has two emergent problems: incomplete-theory-problem [Mitchell et al 86] and incorrect-theory-problem [Mitchell et al 86]. In this paper, I propose a new problem called Rajamoney-88-problem that has both an incomplete and incorrect theory. I propose a technique called Theory Revision to solve the Rajamoney-88-problem.

This abstract corresponds to the following research schema:

ref: {(entails EBL incomplete-theory-problem)
(entails EBL incorrect-theory-problem)}

def: {(instantiates incomplete-theory-problem rajamoney-88-problem)
(instantiates incorrect-theory-problem rajamoney-88-problem)
(solves theory-revision rajamoney-88-problem)}

- RA's S-knowledge is called RA's world *ontology*.
- The *pre-ontology* of a paper is the ontology before the assimilation of a paper.
- The *post-ontology* of a paper is the ontology after the assimilation of a paper. Thus a paper takes a pre-ontology and converts it into a post-ontology.
- *Pre-relations* and *pre-objects*, with respect to a particular paper, are relations and objects belonging to that paper's pre-ontology.
- The **ref** of a paper contains only pre-relations among pre-objects.
- *New-relations* and *new-objects* are new relations and objects that the **def** of a paper introduces into the pre-ontology to obtain the post-ontology.
- The **def** of a paper contains only new-relations, but these relations may involve two new-objects, one new-object and one pre-object or two pre-objects.
- In research schemas, pre-objects and pre-relations are depicted with solid lines, and new-objects and new-relations are depicted with dashed lines.
- A paper is assimilated into the ontology at the point where the pre- and new-ontologies begin to diverge, i.e., at the *pre-objects* among the *new-relations* in **def**.

Table 4.2: Some Terminology for Chapter 4

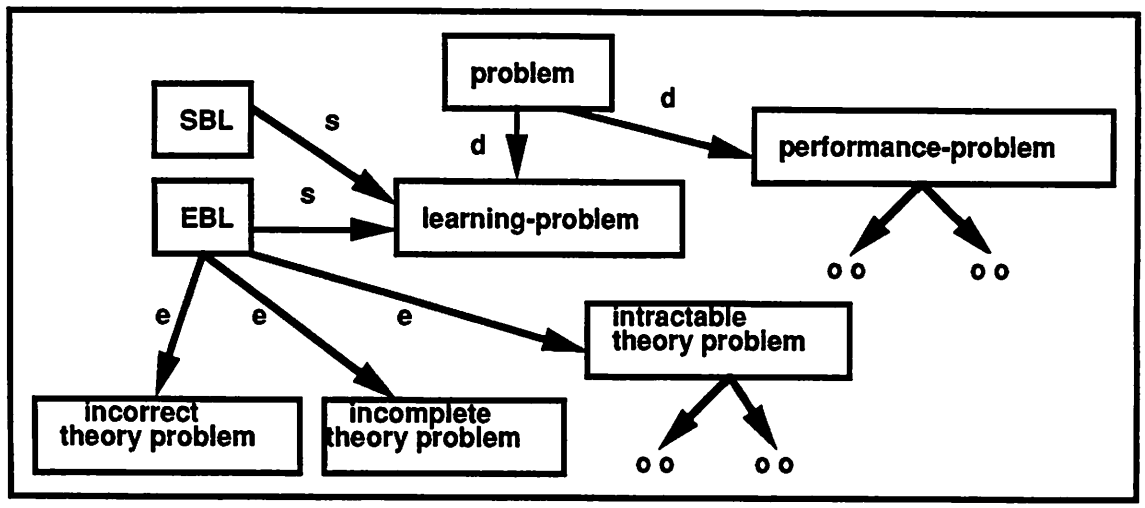


Figure 4.2: The Pre-Ontology of [Rajamoney 88]

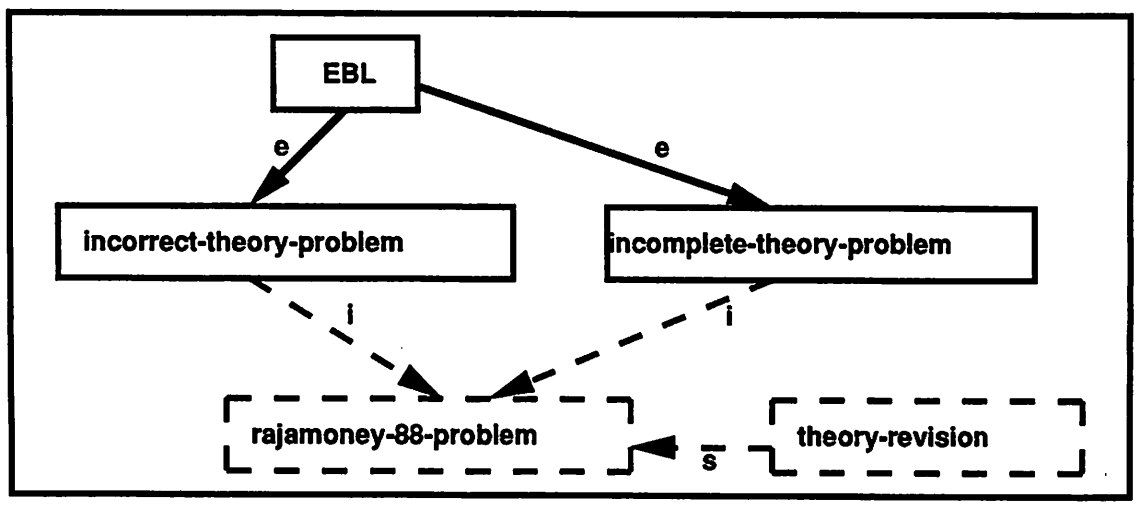


Figure 4.3: The Schema of [Rajamoney 88]

This schema is depicted in Figure 4.3. The two relations in **ref** are already known, i.e., they are pre-relations in the pre-ontology (see Figure 4.15). They state relationships among three pre-objects, namely, EBL, incomplete-theory-problem and incorrect-theory-problem. Hence these relations and objects in **def** are all shown in solid lines.

The relations shown in dashed lines are all new-relations belonging to **def**. The three new-relations involve two new-objects, rajamoney-88-problem and theory-revision, and two pre-objects, incorrect-theory-problem and incomplete-theory-problem. The first two relations each asserts a relationship between a pre-object and a new-object, while the third asserts a relationship between two new-objects.

Figure 4.4 shows the post-ontology of [Rajamoney 88]. Notice that this paper is assimilated into the post-ontology exactly where the pre-ontology and the post-ontology begin to diverge, i.e., at the pre-objects in the **def** of the paper. In simpler terms, when given a new set of information that involves some known objects, the known objects are the transition points between what you knew and what is new[3]. RA's learning strategy starts with the pre-objects belonging to a paper's **def** in order to find the relations belonging to its **ref**.

4.2.3 A Common Sense Example

In this section, I introduce an extended common-sense example in order to illustrate the assumptions behind RA's learning scheme. These assumptions are summarized in Table 4.2.3. Let's assume that Bob tells you, "You know, Paul likes Mary and John hates Paul." To put this in our framework, Bob's utterance may be seen as the **def** of a paper, say, [Bob 90]. The two relations in the **def** are shown below:

{(likes Paul Mary)
(hates John Paul)}

If you knew Bob to be rational — through somewhat prone to gossip — you will assume that there is probably some connection between these two relations since they were stated as part of one discourse event (one paper). With this assumption, the next few pages consider how you might understand [Bob 90] under different pre-ontologies.

Case 1: You've never heard of John, Paul or Mary. Hence, both the relations in **def** are relations among new-objects. There is no way for you to relate this information to anything else you know. At best you can memorize these as isolated facts.

Case 2: Assume that your ontology contains the following two relations: (friend-of me John) and (friend-of me Mary). This means that John and Mary are your friends, but

[3] BTW Originally, I used the terms **pre** and **post** to refer to what are now called **ref** and **def**. If for any reason you decide to look at the Lisp code of RA, in most places you will find **ref** and **def** denoted by the symbols pre and post. The terms **ref** and **def** were inspired by the cartoon characters Mutt and Jeff!

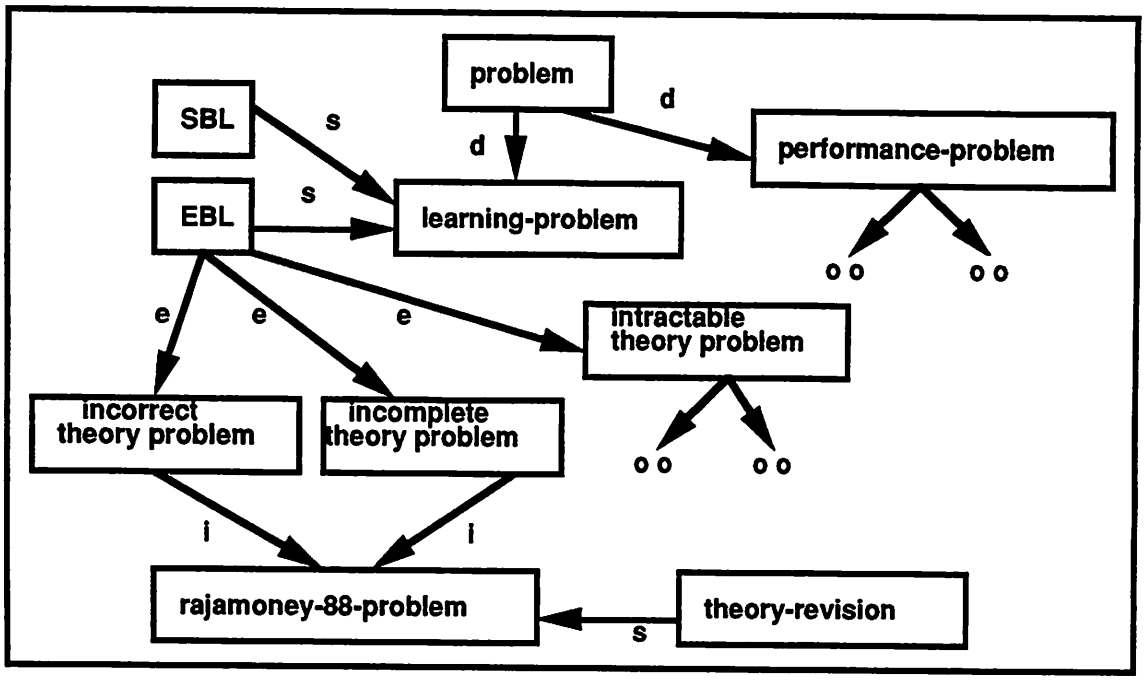


Figure 4.4: The Post-Ontology of [Rajamoney 88]

you've never heard of Paul before. Hence John and Mary are pre-objects (pre-people?) and Paul is a new-man. Now the **def** of [Bob 90] makes a little more sense. You have a way to file away the new information: "John and Mary are my friends. It appears that somebody named Paul likes Mary, and John hates this Paul." What you have done is find some attachment points in your ontology for the **def**. Thus the schema of [Bob 90] is the following:

ref: {(friend-of me John)
(friend-of me Mary)}

def: {(likes Paul Mary)
(hates John Paul)}

This schema is shown in Figure 4.5. It doesn't make very much sense, but one point is worth noting. You understand the new information in terms of how it relates to other things you know: to do that, you start with the known objects in the new information. Hence the first property of **ref** is that, in order to connect the new information to your old ontology, the **ref** includes a set of pre-relations involving the pre-objects in **def**.

Case 3: Suppose Mary is your friend, but John is your husband's friend. Paul is still a new-man. In other words, you have the following relations in your ontology:

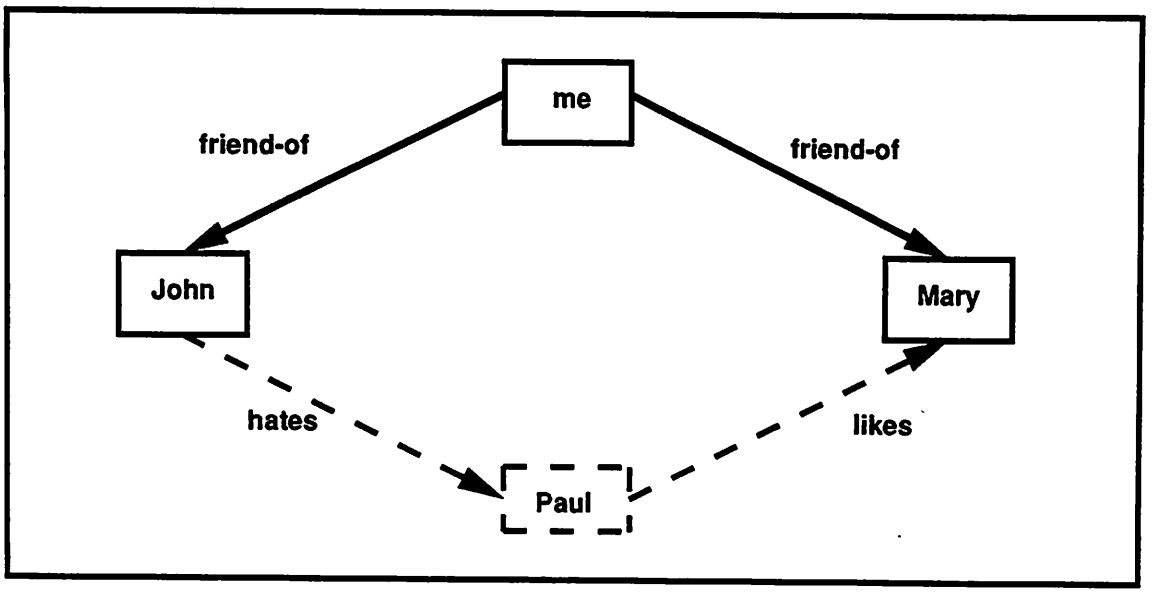


Figure 4.5: The Schema for Case 2

{(friend-of me Mary)
(friend-of Jeff John)
(married-to me Jeff)}

How would you understand Bob's utterance in this case? The first two relations above will connect the **def** of [Bob 90] to your ontology. If you take these two as your **ref**, you get with the following schema:

ref: {(friend-of me Mary)
(friend-of Jeff John)}

def: {(likes Paul Mary)
(hates John Paul)}

This schema is shown in Figure 4.6. Even though the two relations in **ref** connect Bob's utterance to the rest of your ontology, there is something missing in this schema. To see this, imagine yourself repeating these four relations to a total stranger you meet at a party: "You know, Mary is my friend and John is Jeff's friend. Some guy called Paul likes Mary and John hates Paul." This stranger will probably ask "So, who is Jeff?" Only when you tell the stranger that Jeff is your husband, can he understand your utterance in a coherent way. The schema for [Bob 90] is more likely the following:

ref: {(married-to me Jeff)
(friend-of me Mary)
(friend-of Jeff John)}

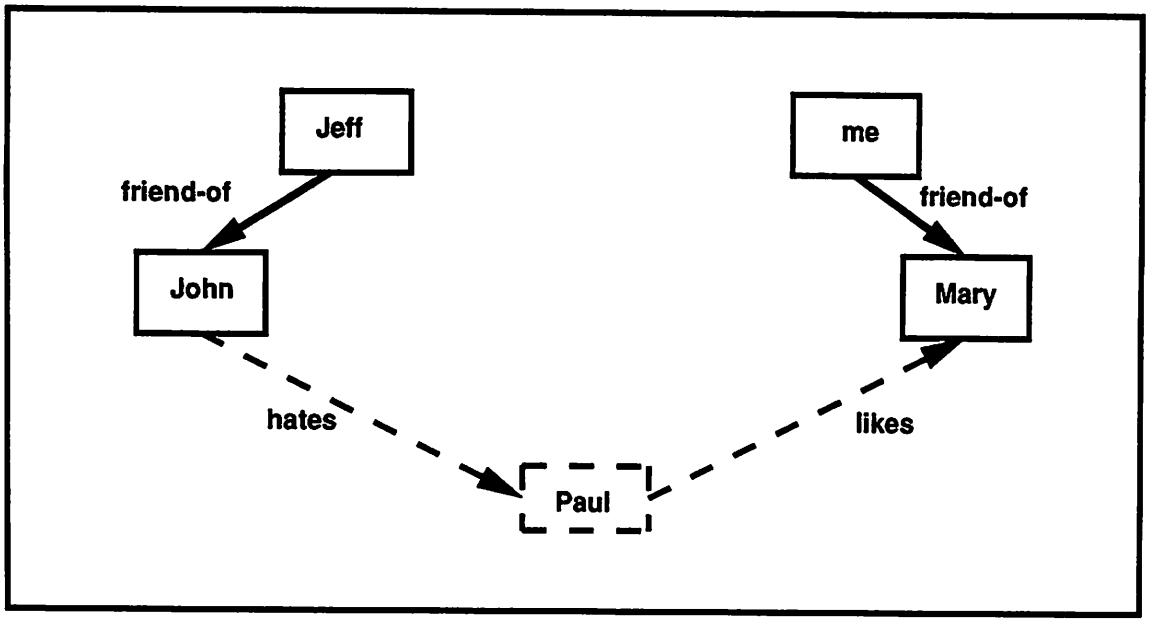


Figure 4.6: An Invalid Schema for Case 3

def: {(likes Paul Mary)
(hates John Paul)}

This schema is shown in Figure 4.7. Thus the purpose of **ref** is not only to connect the pre-objects of **def** to the ontology, but also to connect the pre-objects to each other. Syntactically, **ref** is a tree connecting all the pre-objects in **def**.

Case 4: In addition to the pre-ontology shown in Case 2, suppose you also know that John is married to Mary, i.e., your ontology includes another relation (married-to John Mary). Now what Bob told you makes even more sense. Not only are John and Mary your friends, there is an even more specific relationship between them: they are married to each other. Presumably, you'll attribute the following schema to [Bob 90]:

ref: {(married-to John Mary)}

def: {(likes Paul Mary)
(hates John Paul)}

This schema corresponds to the following abstract for [Bob 90]:

> John is married to Mary [ref: town-registry]. In this paper I say that Paul likes Mary and John hates Paul.

This schema is shown in Figure 4.8. From this schema, you could probably learn the following general rule: If x is married to y, then if z likes y, x will hate z. The point of

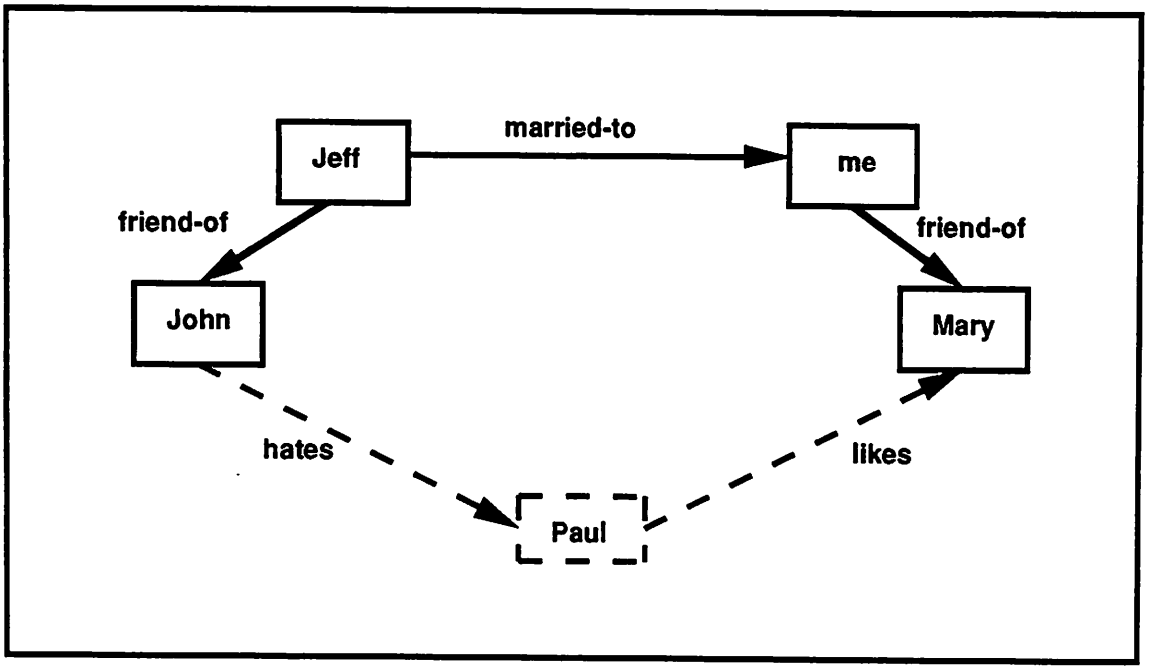


Figure 4.7: The Schema for Case 3

this example is that not only does **ref** connect the pre-objects in **def**, it connects them through the *most specific* relations among them in the pre-ontology. Syntactically, **ref** is a *minimal* tree connecting the pre-objects of **def**[4].

Case 5: Now I will show that **ref** should in fact be a tree, not any arbitrary graph. This restriction is best understood from the perspective of research schemas as heuristic rules. Let's consider the same example as in Case 2. Assume that you attribute the following schema to [Bob 90]:

ref: {(friend-of me Mary)
(friend-of me John)
(married-to John Mary)}

def: {(likes Paul Mary)
(hates John Paul)}

This schema is shown in Figure 4.9. As a description of what Bob told you, this schema is fine. You would understand this as "John is married to Mary; they are both my friends. Apparently Paul likes Mary and John hates Paul." However, as a heuristic rule, this schema is over-specific. This schema corresponds to the following rule: "If x and

[4]If there are several such minimal trees, RA prefers one involving domain-dependent relations over one involving epistemological relations.

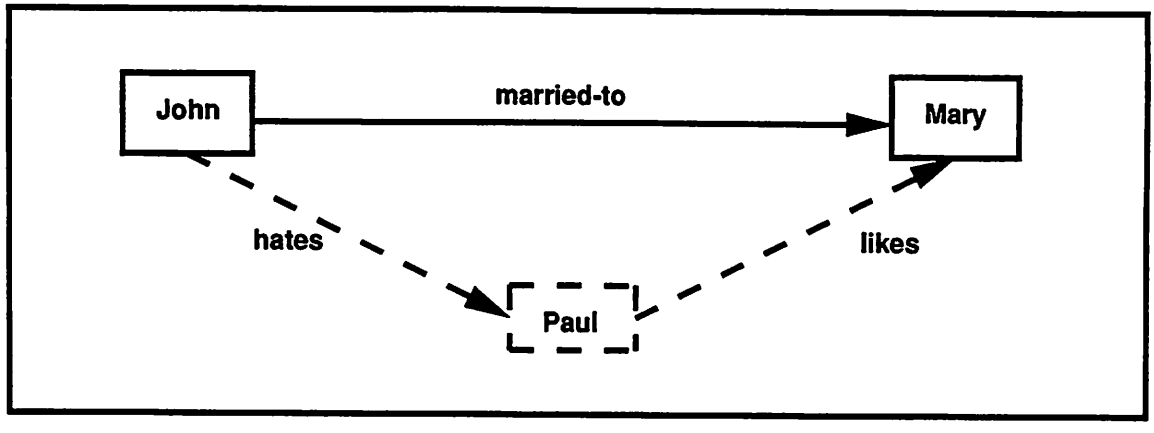


Figure 4.8: The Schema for Case 4

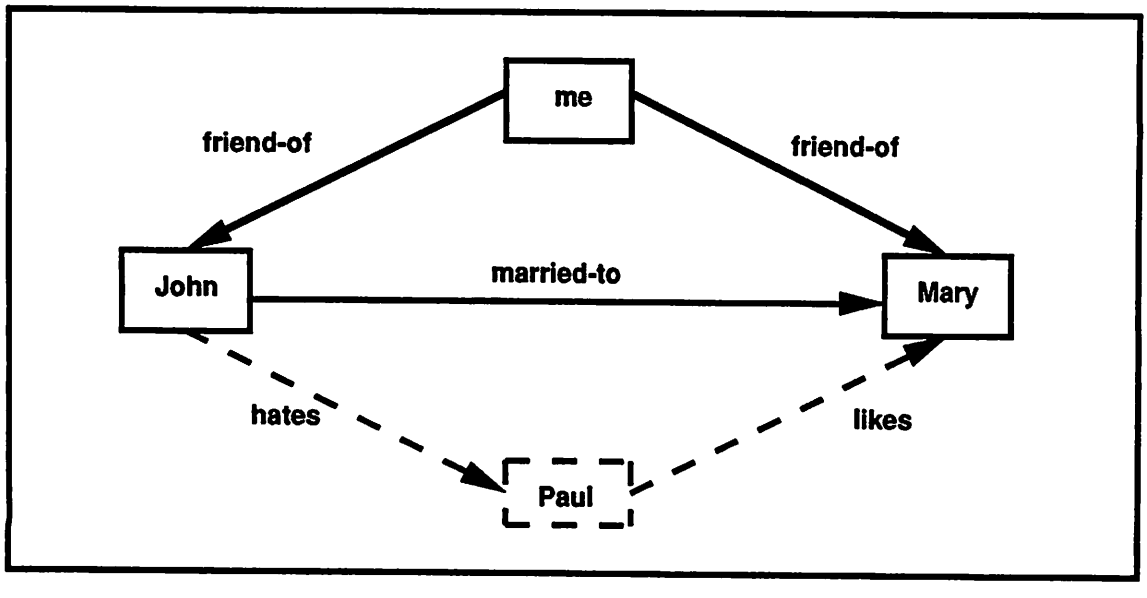


Figure 4.9: The Schema for Case 5

y are my friends and x is married to y, then if z likes y, x will hate z." Presumably, jealousy is a universal trait not confined to just your friends. As a heuristic rule, the only applicability condition that matters is that x and y be married, not that they be your friends[5]. Thus the **ref** is a set of relations that constitute a single shortest path, i.e., a tree, among the pre-objects in **def**.

These properties of **ref** are tantamount to the following two assumptions: (1) If you don't have a causal theory about a domain, assume that a set of new-relations among some known objects is not random. The new relations are probably due to some already existing relationships among these objects. (2) In addition, the more direct these existing relations, the more relevant they are likely to be (e.g., the relation that John is married to Mary). Relations farther removed may be purely coincidental (e.g., the relations that John and Mary are your friends). These two assumptions enable RA to infer a paper's **ref** from its **def**. Table 4.2.3 summarizes the properties of **ref** and the assumptions behind the learning scheme. See Chapter 5 for an in-depth analysis of RA's learning strategy.

4.3 RA II: How Does It Work?

RA II's learning technique consists of two phases: the *assimilation* phase and the *generalization* phase. During the assimilation phase, RA infers the **ref** of a paper from its **def**. This **ref**, **def** pair constitutes the paper's instantiated research schema. During the generalization phase, RA converts the instantiated schema into a skeletal schema. The two subsections of this section describe the two phases of RA's learning technique. Table 4.4 contains a synopsis of the learning technique.

Before we proceed, some disclaimers are in order. In spite of the lofty statements and quotations, RA II is an extremely simple system. It implements two trivial mechanisms well-known among Lisp programmers: an intersection search and a binding list. The intersection search is used during the assimilation phase to find a path among the pre-objects in the **def**. The binding list is used during the generalization phase to find the appropriate variable substitutions for the constants in the instantiated schema.

I should further mention that the intersection search procedure was hacked together to handle a set of several examples without worrying about its efficiency, generality and other such noble goals. Thus the search procedure itself is of little theoretical significance. This spirit pervades Section 4.3.1 where I sketch how the search procedure

[5]One way to understand this is in terms of the following two scenarios: (1) It is possible that John and Mary are married to each other because they are both your friends — perhaps you set them up on a blind date; (2) It is possible that John hates Paul because John is married to Mary and Paul likes Mary. In isolation it makes some sense to attribute these causalities. However, situation (2) is independent of situation (1). By insisting that **ref** should be a tree we include only one level of 'causality' in heuristic rules.

Properties of ref

- The **ref** includes relations that involve the pre-objects among the new-relations in **def**. This connects the new information in **def** to the rest of the ontology.
- The **ref** is a path the connects *all* the pre-objects in **def**. This ensures that all the relevant information is included.
- The **ref** is the *most specific path* that connects all the pre-objects in **def**. This ensures that the most specific and relevant information is included in **ref**.
- The **ref** is a tree, not an arbitrary graph. For heuristic rules, this ensures that the **ref** includes only one level of causality.

Assumptions behind the learning scheme

- In the absence of a causal theory of the domain, assume that some new information pertaining to some known entities is not random. Assume that one needs to know the existing relationship among these entities in order to assimilate the new information.
- In addition, assume that the most specific relationship among these entities is the most relevant. Relations farther removed may be purely coincidental.

Table 4.3: Properties of ref and the Assumptions in Learning

works. However, this statement should not deter you from appreciating the importance of the ideas behind the learning strategy: assimilating a new piece of information involves the process of finding the most specific information that relates the new information to the one's current ontology. This point holds irrespective of the generality or efficiency of the symbol-level procedures.

4.3.1 Assimilation Phase

During the assimilation phase, RA infers the **ref** of a paper from its **def**. To do this, first the pre-objects in the **def** are identified and isolated[6]. Now there are two possible cases to consider:

Case 1: If there is only one pre-object in **def**, then we just need to find a pre-relation involving this object. This relation will connect the pre-object to the rest of the ontology. If there are several such pre-relations to choose from, the predicate-preference rules (see Table 4.4) are used to choose one.

Case 2: If there is more than one pre-object, then **ref** should include a set of pre-relations that constitute the most-specific path connecting all the pre-objects. The most-specific path is the the shortest path among the pre-objects; if there is more than one shortest path, then the predicate-preference rules are used to choose the best one.

The predicate preference rules induct a weak sense of causality into the learning scheme: Suppose the **def** of a paper says that it proposes an emergent problem for some technique T1. To understand this paper, it is probably more important to understand what problem T1 solves rather than what other emergent problems T1 entails. The predicate preference rules order the relational predicates of the representational language in terms of their importance. The preference ordering used in the RA system is shown in Table 4.4. In general, domain-dependent predicates are preferred over epistemological ones.[7,8]

The intersection search procedure works as follows: It starts a search simultaneously from each of the pre-objects. For each pre-object p, one preferred relation (according to the order above) involving that object is added to the tentative **ref**. Without loss of generality, assume that this relation has p as its primary, i.e, this relation has the form (predicate p q). This **ref** is examined to see if any of the pre-objects have been connected, i.e., if q has turned out to be one of the other pre-objects. If so, this object, q, is eliminated from the search. If not, the search is continued from q to include another preferred relation of the form (predicate q r). This cycle is continued until either all the pre-objects have been connected or until some path length exceeds a constant, say 4. In

[6]This is accomplished through a call to the predicate frame-exists-p (object).

[7] [Btw] At the time of this writing, I know of one schema for which no total or partial order will work. This suggests some directions for future research. See Chapter 7.

[8]RA II cannot learn schemas involving the ninth predicate acq. See Chapter 5.

the latter case, we retract the last leg of this path and consider the next preferred relation involving q. This process is continued until all the pre-objects have been connected[9]. The set of relations found by this process to connect the pre-objects is taken to be the paper's **ref**. The generalization phase is now invoked to convert this **ref**, **def** pair (the instantiated schema) into a skeletal schema. We will see several examples of the assimilation process later in Section 4.4.

- The learning technique consists of two phases: *assimilation* phase and *generalization* phase.
- The assimilation phase uses an intersection search to find the shortest path that connects the pre-objects in the **def**. If more than one shortest path exists, the predicate preference ordering (below) determines the conceptually shortest path. The shortest path is taken to be the **ref** of the paper. This **ref**, **def** pair constitutes the instantiated schema of the paper.
- The generalization phase uses a binding list to convert the instantiated schema into a skeletal schema. The constants in the instantiated schema are generalized by replacing them with *typed* variables.
- The Predicate Preference ordering used in RA: solves, entails, exhibits, involves, R, encapsulates, instantiates, dominates. Domain dependent predicates are preferred over epistemological predicates.

Table 4.4: Synopsis of RA's Learning Technique

4.3.2 Generalization Phase

The generalization phase converts an instantiated research schema into a skeletal schema by replacing the constants in the instantiated schema by typed variables. The generalization process itself is extremely trivial, but the reason why it works is not. This section describes the generalization process. The analysis of why it works is postponed until Chapter 5.

[9] Btw You may have noticed that this algorithm, in general, may not find the shortest path if the shortest path consists entirely of predicates that are very low in the preference order. For the examples considered in RA II, this has not been a problem. A more general strategy would be to construct all paths, and then use a scoring scheme to prefer the best one: this may involve attributing a score, i.e., a conceptual distance measure, to each predicate: e.g., solve=1, entail=1.1,..., dominates=1.7 and so on. The path that has the lowest total score would be conceptually the shortest.

The generalization algorithm considers the relations in a schema one by one. Each relation states a relationship between two objects. If either of the two objects is already bound to a variable, the object is replaced by that variable; if not, the type of the object is noted, and a new variable of that type is created[10]. The object is replaced in the relation by the newly created variable, and the association (binding) between the object and the variable is noted in a binding list. This variable will substitute all future occurrences of this object within the same schema.

As an example, let's consider the schema of [Shavlik 88]. The instantiated version of this schema is shown below:

ref: {(solves EBL learning-problem)}

def: {(entails EBL number-generalization-problem)
(solves bagger-technique number-generalization-problem)}

Processing the first relation in **ref**, we create a binding T1 for EBL and P1 for learning-problem. Thus the **ref** of the skeletal schema becomes {(solves T1 P1)}. Processing the next relation (i.e., the first relation in **def**), we note that EBL is already bound to T1, but number-generalization-problem is unbound. We create a new variable P2 and bind it to number-generalization-problem. Thus the the **def** of the skeletal schema becomes {(entails T1 P2)}. Processing the last relation, we see that number-generalization-problem is already bound, but bagger-technique is unbound. So we create another new variable T2 to substitute for this object. All the relations in the instantiated schema have now been processed; the skeletal schema obtained from it is shown below:

ref: {(solves T1 P1)}

def: {(entails T1 P2)
(solves T2 P2)}

This skeletal schema corresponds to the following heuristic:

> If there exist P1 and T1 such that T1 solves P1, then suggest, "You can propose an emergent problem P2 of T1. You could then propose a technique T2 to solve P2."

Notice that this generalization procedure made a big conceptual leap: a single example (the instantiated schema) that involved a set of specific objects belonging to certain types was assumed to apply to any objects belonging to those types. In doing this, we constructed neither a deductive justification nor an inductive generalization. You are invited to ponder this one until Chapter 5 where I will justify RA's generalization strategy.

[10]Chapter 2 introduced our convention for typing the variables: Prefix T for technique, P for problem, C for concept and Pr for property.

4.4 RA II: Examples

This section illustrates RA's learning strategy by using six different examples. These examples have been chosen to illustrate different kinds of research schemas learned by RA: Example 1 illustrates the boundary case where the **def** contains a single pre-object. Examples 2 and 3 illustrate cases where the **ref** is fairly general, but the **def** is increasingly specific. Examples 4 and 5 illustrate increasingly specific **ref** but increasingly general **def**. Finally example 6 revisits the [Rosenbloom 88] schema which we saw in Chapter 1 in order to consider a schema whose **ref** as well as **def** are fairly specific. Table 4.5 summarizes the characteristics of the six examples considered in this section.

Example	Characteristic
1	Boundary case with one pre-object.
2	Boundary case with two pre-objects.
3	General **ref**, specific **def**.
4	More complicated **ref** with 2 pre-objects.
5	Specific **ref**, general **def**.
6	Specific **ref** with 3 pre-objects, specific **def**.

Table 4.5: Characteristics of the Six Examples

Before proceeding, let me clarify the sense of the terms 'general' and 'specific' as used in the last paragraph. In Section 4.2.3, I used the term 'specific' in defining the properties of **ref**. There I showed that the **ref** should include the most specific set of relations that connect the pre-objects in the **def** (Case 4, page 105). This use of the term 'specific' relates to how the schema captures the paper in question. The current use of the two terms 'specific' and 'general' refer to a schema when used as a heuristic rule. A schema has a very general **ref** if the **ref** contains a small number of relations (usually one). Such a schema, as a heuristic rule, will be widely applicable since its applicability conditions are weak. In contrast, a schema that has several mutually dependent relations in its **ref** has a very specific or strong **ref** since such a schema, when used as a heuristic rule, will apply only under very specific circumstances. Similarly, a schema that has a number of relations in its **def** tends to offer a strong or specific advice; by contrast, a schema with a small number of relations in its **def** offers a weak or general advice.

4.4.1 Example 1: [Shavlik 88]

This example involves the boundary case where the **def** has only one pre-object. Hence, the purpose of the **ref** is simply to connect the **def** to the rest of the ontology. The **def** of [Shavlik 88] is shown below:

{(entails EBL generalizing-number-problem)
(solves bagger-technique generalizing-number-problem)}

Figure 4.10 shows the pre-ontology of this paper[11]. Against this pre-ontology, [Shavlik 88] introduces two new-relations that involve the following three objects: EBL, generalizing-number-problem, and bagger-technique. Of these, EBL is a pre-object and the other two are new-objects. This paper asserts that EBL has an emergent problem called generalizing-number-problem and that bagger-technique solves that problem. Since EBL is the only pre-object, the attachment point of this paper to the pre-ontology is precisely the EBL node. As a first approximation, let's consider a **ref** as follows: '{EBL}'. This corresponds to the following abstract:

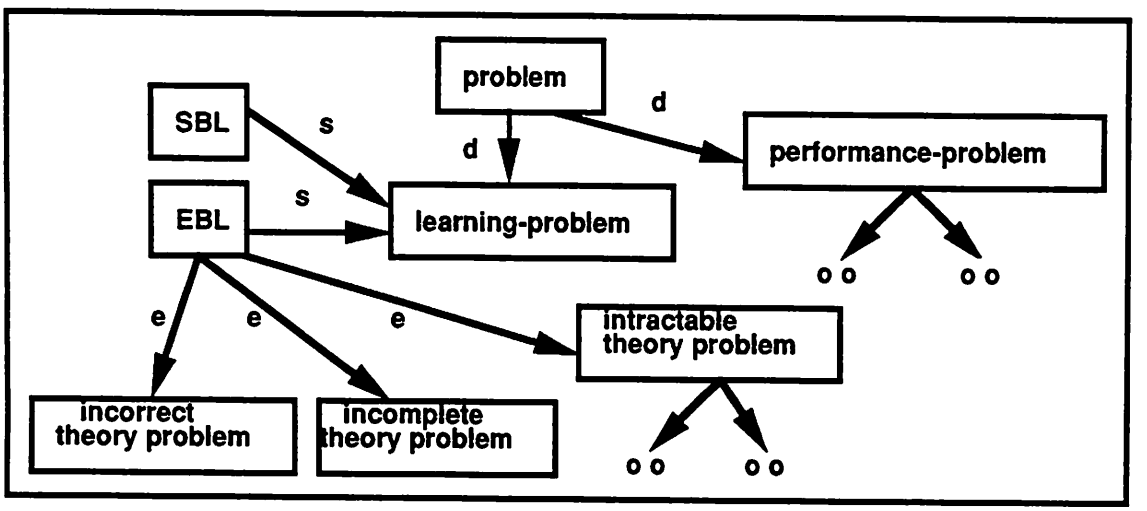


Figure 4.10: The Pre-Ontology of [Shavlik 88]

> There is a technique called EBL [ref]. In this paper, I show that EBL has an emergent problem called generalizing-number-problem. I propose a technique called bagger-technique to solve generalizing-number-problem.

However, our convention (1) (page 97) demands that only relations be part of **ref** and **def**. Therefore, we need a pre-relation involving EBL. Four such relations exist in the pre-ontology in Figure 4.10. The predicate-preference ordering (Table 4.4) is used to choose the most significant one, namely, (solves EBL learning-problem). This gives us the following schema for [Shavlik 88]:

ref: {(solves EBL learning-problem)}

[11] In all figures in this section, only the relevant portions of the pre-ontologies are shown.

def: {(entails EBL generalizing-number-problem)
(solves Bagger-technique generalizing-number-problem)}

This schema corresponds to the following abstract:

> EBL is a technique to solve the learning-problem [Mitchell et al 86]. In this paper, I show that EBL has an emergent problem called generalizing-number-problem. I propose a technique called Bagger-technique to solve generalizing-number-problem.

The next phase in the learning process is to convert this schema into a skeletal schema. As we saw in Section 4.3.2, this schema corresponds to the following skeletal schema:

ref: {(solves T1 P1)}

def: {(entails T1 P2)
(solves T2 P2)}

This skeletal schema is shown in Figure 4.11. It corresponds to the following heuristic:

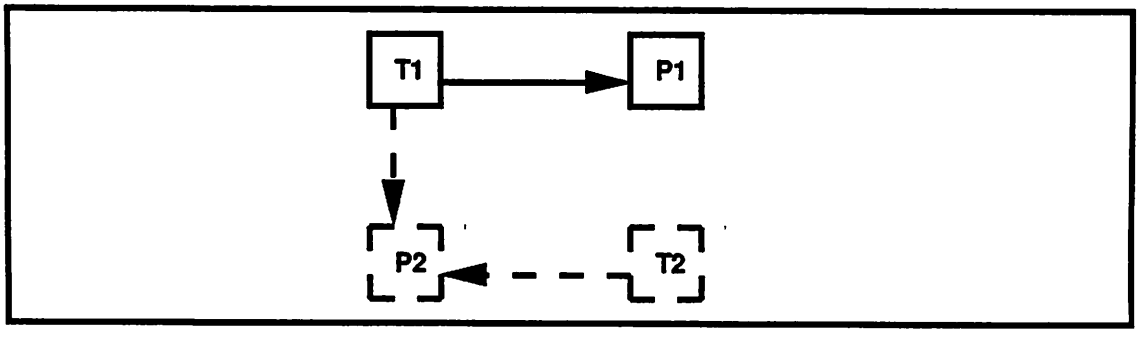


Figure 4.11: The Skeletal Schema of [Shavlik 88]

> If there exist P1, T1 such that T1 solves P1, then suggest, "You can look for an emergent problem P2 of T1. Then you can propose a technique T2 to solve P2."

As a heuristic, this schema is a very general schema: its applicability condition, i.e., its **ref**, is satisfied for any 'solves' relation between any problem and any technique. The next example involves a research schema where the **def** has two pre-objects. The resulting schema is still very weak, but the **def** is better constrained.

4.4.2 Example 2: [Samuel 67]

Let's now consider the simplest case in which the **def** of a paper has more than one pre-object. In Section 4.2.3, we saw that when there is more than one pre-object in the **def**, the **ref** should not only connect the **def** to the ontology, but also connect the pre-objects to each other. This example considers a case where a single relation connects the two pre-objects to the pre-ontology as well as to each other.

To illustrate this case, I will use two classical papers in machine learning. In 1959, Arthur Samuel developed a technique to solve the problem of learning to play checkers [Samuel 59]. The schema of this paper is shown below:

ref: {(dominates problem learning-problem)}

def: {(instantiates learning-problem checker-learning-problem)
(solves samuel-59-technique checker-learning-problem)}

Eight years later, in 1967, Samuel published another paper, [Samuel 67], in which he proposed some enhancements to his original technique. The new technique involved a data-structure called a signature table, so let's call it the signature-table-technique. The **def** of [Samuel 67] is shown below:

{(solves signature-table-technique checker-learning-problem)
(R signature-table-technique samuel-59-technique)[12]}

The pre-ontology of [Samuel 67] is shown in Figure 4.12. Of the three objects (among the two relations) in the **def** of [Samuel 67], two are pre-objects, i.e., they exist in the pre-ontology. These two are checker-learning-problem and samuel-59-technique. To infer the **ref** of the paper, RA starts an intersection search from these two pre-objects by adding one relation involving each pre-object to the tentative **ref**. There are two relations involving checker-learning-problem; the predicate-preference ordering is used to choose the following relation as the preferred one: (solves samuel-59-technique checker-learning-problem). There is only one relation involving samuel-59-technique: (solves samuel-59-technique checker-learning-problem). These two relations are added to the tentative **ref**. Now the **ref** is examined to see if the pre-objects have been connected. RA finds that the two relations in the tentative **ref** are, by golly, the same one that also happens to connect the two pre-objects. Hence the construction of **ref** is complete and RA attributes the following schema to [Samuel 67]:

ref: {(solves samuel-59-technique checker-learning-problem)}

def: {(solves signature-table-technique checker-learning-problem)
(R signature-table-technique samuel-59-technique)}

[12]Remember that R is our generic relation for any subject-dependent relation. In this case, R stands for 'extends' or 'enhances.'

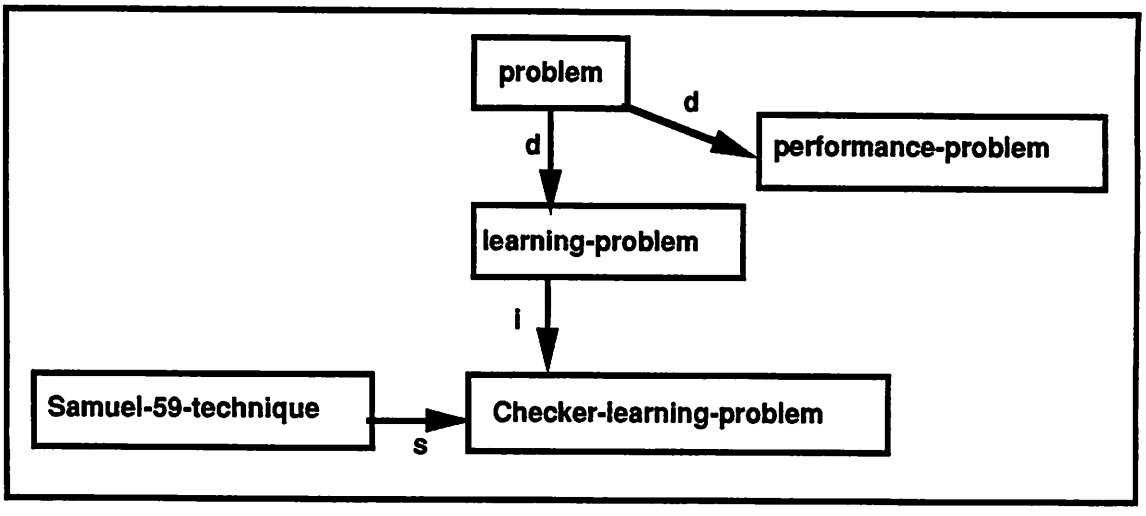


Figure 4.12: The Pre-Ontology of [Samuel 67]

This schema corresponds to the following abstract:

> I proposed samuel-59-technique to solve the checker-learning-problem [Samuel 59]. In this paper, I propose a technique called signature-table-technique to solve the same problem; signature-table-technique is R-related to (extends) samuel-59-technique.

This schema corresponds to the following skeletal schema:

ref: {(solves T1 P1)}

def: {(solves T2 P1)
(R T2 T1)}

This skeletal schema is shown in Figure 4.13. It corresponds to the following heuristic:

> If there exist T1, P1 such that T1 solves P1, then suggest, "You could propose a technique T2 to solve P1 such that T2 is R-related to T1."[13]

Like the schema of Example 1, the **ref** of this schema is also quite general. As a heuristic rule, it will be triggered by any 'solves' relation between any two objects. However, the suggestion part is a little more specific: the new object in the suggestion part, T2, is constrained in two ways. Not only is T2 required to solve P1, but it should also be R-related to T1. The next example considers a schema with a much more specific **def**.

[13]As we have seen in Chapter 3, the extra slots used with R-relations can be used to make more specific suggestions; in this case that would be "You could propose a technique T2 to solve P1 such that T2 extends T1."

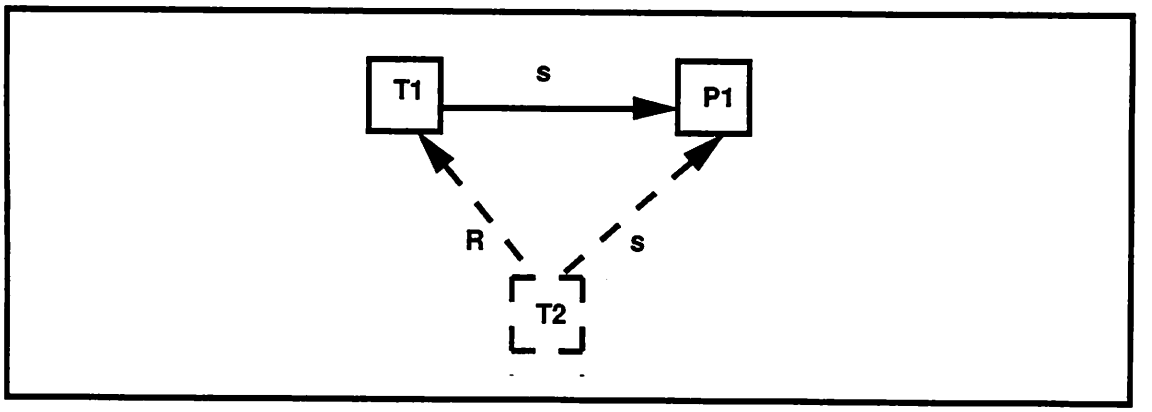


Figure 4.13: The Skeletal Schema of [Samuel 67]

4.4.3 Example 3: [Minsky & Papert 69]

This subsection revisits the work of Minsky and Papert on perceptrons [Minsky & Papert 69]. This schema was considered briefly in Chapter 1, but in a somewhat simpler form. Here, I consider this schema again to illustrate a case involving a very general **ref** but a highly constrained **def**. Before we get to [Minsky & Papert 69], let's consider the work of Rosenblatt that gave rise to Minsky and Papert's work. In 1957, Rosenblatt proposed a learning algorithm called 'perceptrons' to learn boolean functions. Twelve years later, in 1969, Marvin Minsky and Seymour Papert published a book that proved an important negative result about perceptrons [Minsky & Papert 69]. In what follows, I will use the abbreviation 'bf' for 'boolean functions' and 'lp' for 'learning problem.' The **def** of [Minsky & Papert 69] is shown below:

{(dominates bf-lp linearly-separable-bf-lp)
(dominates bf-lp linearly-not-separable-bf-lp)
(solves perceptron linearly-separable-bf-lp)
(not-solves perceptron linearly-not-separable-bf-lp)
(not-solves perceptron bf-lp)
(entails perceptron linearly-not-separable-bf-lp)}

This **def** introduces 6 new-relations among four objects; of the four, bf-lp and perceptron are pre-objects and the other two, linearly-separable-bf-lp and linearly-not-separable-bf-lp, are new-objects. To find the **ref**, RA looks for a set of relations connecting the two pre-objects, bf-lp and perceptron. A single unique relation exists between them: (solves perceptrons bf-lp). Hence RA infers the following schema for [Minsky & Papert 69]:

ref: {(solves perceptron bf-lp)}

def: {(dominates bf-lp linearly-separable-bf-lp)
(dominates bf-lp linearly-not-separable-bf-lp)
(solves perceptron linearly-separable-bf-lp)
(not-solves perceptron linearly-not-separable-bf-lp)
(not-solves perceptron bf-lp)
(entails perceptron linearly-not-separable-bf-lp)}

This schema corresponds to the following abstract:

> Perceptrons solve the bf-lp [Rosenblatt 57]. In this paper, we show that bf-lp consists of two problems, linearly-separable-bf-lp and linearly-not-separable-bf-lp. Perceptrons can solve the former but not the latter. Therefore perceptrons do not solve bf-lp. Linearly-not-separable-bf-lp is an emergent problem of perceptrons.

This schema is converted into the following skeletal schema:

ref: {(solves T1 P1)}

def: {(dominates P1 P2)
(dominates P1 P3)
(solves T1 P2)
(not-solves T1 P3)
(not-solves T1 P1)
(entails T1 P3)}

This skeletal schema is shown in Figure 4.14. It corresponds to the following heuristic:

> If there exist P1 and T1 such that T1 solves P1, then suggest, "You can find an emergent problem P3 of T1. One strategy for finding the emergent problem P3 of a technique T1 is to see if the technique really solves the problem P1 it purports to solve. Can you find subclasses P2 and P3 of P1 such that T1 solves one subclass P2, but not the other subclass P3. Then T1 does not solve P1, and P3 is an emergent problem of T1."

Note that this schema also has a fairly trivial applicability condition, but its suggestions are quite specific: the new-objects in the **def** are constrained in several ways by the pre-objects in the **ref**. Unlike the suggestion from the schema of Example 1 that simply asked you to look for an emergent problem of a technique, the suggestion from this schema provides several additional constraints on what this emergent problem could be. While most of these constraints were taken from the **def** (i.e., the input), the one constraint in **ref** that RA infers by itself is crucial for this schema: without knowing that T1 solves P1, it will be rather silly to suggest, "Maybe you could show that T1 does not solve P1!" Hence, even when the **ref** is simple and weak, it can be an important piece of information for the applicability of a schema.

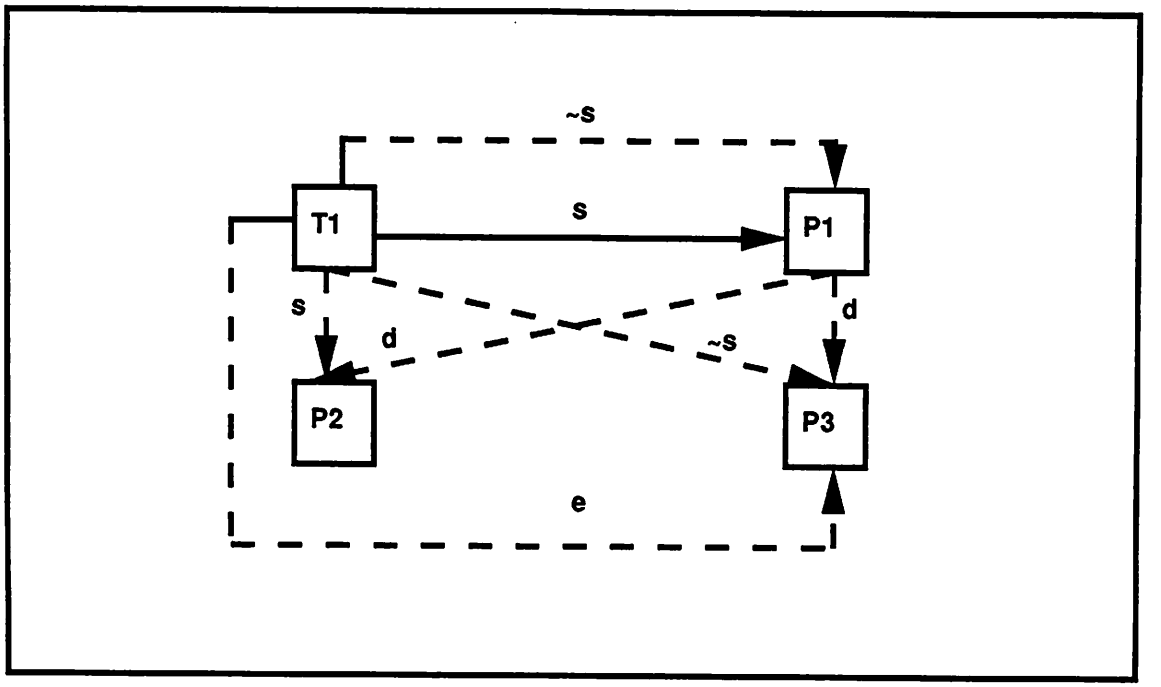


Figure 4.14: The Skeletal Schema of [Minsky & Papert 69]

4.4.4 Example 4: [Rajamoney 88]

The paper [Rajamoney 88] was used earlier (in Section 4.2.2) as an example to solidify some of the terminology introduced in Table 4.2. Here, I will consider this paper again to illustrate a case where the system has to do a little more work to infer the **ref**. The **def** of [Rajamoney 88] is shown below:

{(instantiates incorrect-theory-problem rajamoney-88-problem)
(instantiates incomplete-theory-problem rajamoney-88-problem)
(solves theory-revision rajamoney-88-problem)}

The pre-ontology of this paper is shown in Figure 4.15. The **def** of this paper introduces three relations among four objects. Two of these, incorrect-theory-problem and incomplete-theory-problem, are pre-objects, and the other two, rajamoney-88-problem and theory-revision, are new-objects. There is no direct connection between the two pre-objects but there are two relations that connect incorrect-theory-problem and incomplete-theory-problem via the EBL node: {(entails EBL incomplete-theory-problem) and (entails EBL incorrect-theory-problem)}. These two relations relate the pre-objects to the ontology, as well as connect the two pre-objects to each other. Hence RA attributes the following schema to [Rajamoney 88]:

ref: {(entails EBL incorrect-theory-problem)
(entails EBL incomplete-theory-problem)}

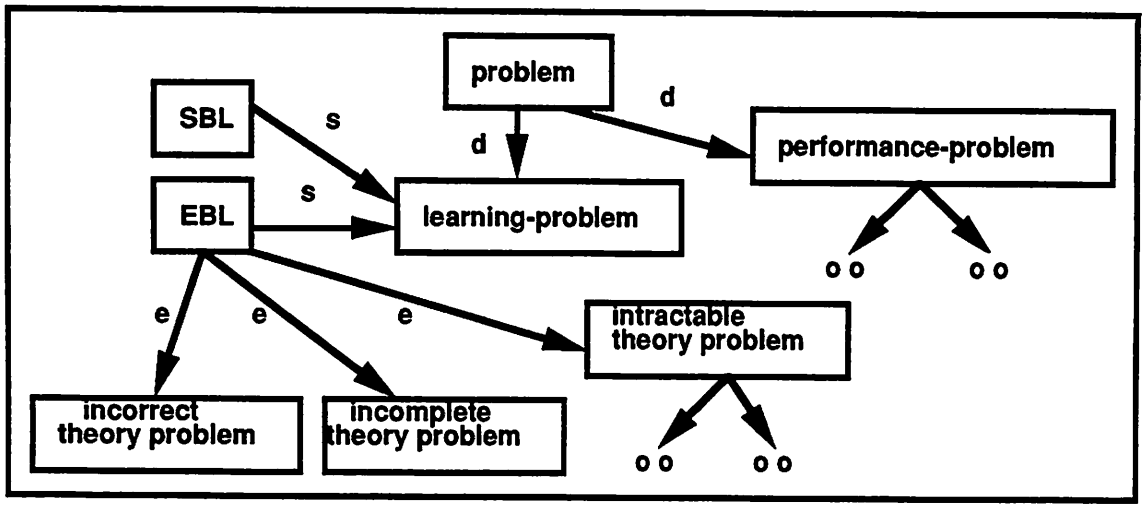


Figure 4.15: The Pre-Ontology of [Rajamoney 88]

def: {(instantiates incorrect-theory-problem rajamoney-88-problem)
(instantiates incomplete-theory-problem rajamoney-88-problem)
(solves theory-revision rajamoney-88-problem)}

This schema corresponds to the following abstract:

> EBL has two emergent problems: incorrect-theory-problem [Mitchell et al 86] and incomplete-theory-problem [Mitchell et al 86]. In this paper, I propose a new problem called rajamoney-88-problem that has both an incomplete and an incorrect theory. I propose a technique called theory-revision to solve Rajamoney-88-problem[14].

This schema is turned into the following skeletal schema:

ref: {(entails T1 P1)
(entails T1 P2)}

def: {(instantiates P1 P3)
(instantiates P2 P3)
(solves T2 P3)}

This skeletal schema is shown in Figure 4.16. It corresponds to the following research heuristic:

[14] ML The incomplete theory problem of EBL refers to the case where a system's theory of the domain is not causally complete. The incorrect theory problem seems to include a broad class of cases where the domain theory has rules that are downright wrong, that are mutually contradictory (also called inconsistent theory problem) or lead to multiple, often mutually exclusive, explanations (also called the promiscuous theory problem). The theory revision technique designs experiments to detect incorrect rules in the domain theory and to infer new causal rules to complete the theory.

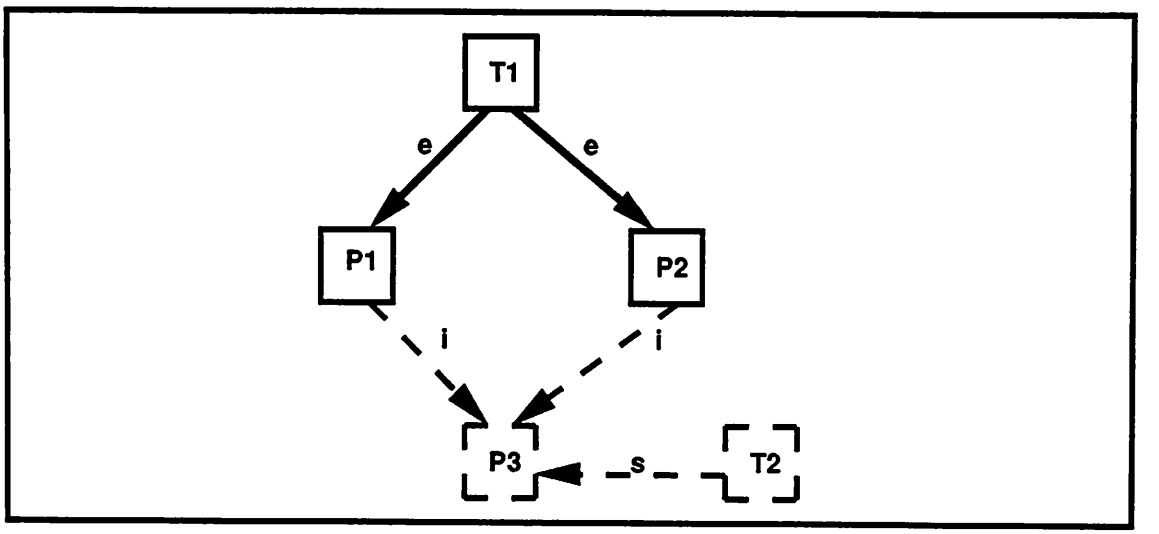


Figure 4.16: The Skeletal Schema of [Rajamoney 88]

> If there exists a technique T1 such that it has two emergent problems P1 and P2, then suggest, "You can propose a new problem P3 that is an instantiation of both P1 and P2. You can propose a technique T2 to solve P3."

This schema has a more specific applicability condition. The condition requires three objects, T1, P1 and P2 to be in a certain specific relationship to each other: T1 should have both P1 and P2 as its emergent problems. When such a configuration is found, this schema suggests that you find a new problem that's an instantiation of both these emergent problems; as if this is not enough, it suggests that you also solve this new problem.

4.4.5 Example 5: [Hirsh 88]

This example describes a schema where the **def** consists of a single relation, but the **ref** consists of three relations to relate two far-flung objects in the **def**. The paper [Hirsh 88] takes a known property of similarity-based learning, search-property, and shows how it applies to hybrid techniques that combine similarity-based learning with explanation-based learning[15,16]. The **def** of this paper is shown below:

{(R lebowitz-86-technique search-property)}

[15] I have simplified this schema somewhat. While the actual paper considers how the search property applies to hybrid techniques in general, I show the paper as asking a simpler question: how does the search property apply to one specific hybrid technique, Lebowitz-86-technique.

[16] [ML] Remember that search-property stands for the statement that all similarity-based learning techniques may be seen as a search in a hypothesis space [Mitchell 81]. In this paper, Hirsh shows that hybrid techniques that combine SBL with EBL can be classified into three groups: for one group, the

Figure 4.17 depicts the pre-ontology of [Hirsh 88]. This paper asserts a new relation between two pre-objects: lebowitz-86-technique and search-property. To find the **ref**, RA starts an intersection search from the two pre-objects. This search process finds that lebowitz-86-technique encapsulates another technique called lebowitz-86-sbl-component which is an instantiation of SBL; further SBL exhibits search-property[17]. Thus a connection between lebowitz-86-technique and search-property has been found, and RA infers the following schema for [Hirsh 88]:

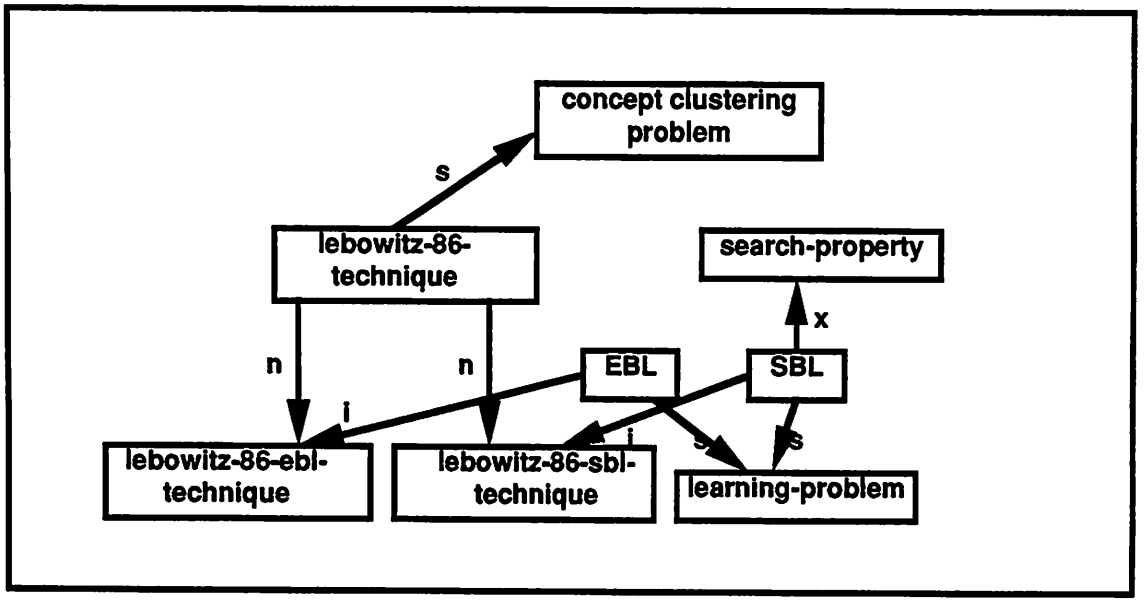


Figure 4.17: The Pre-Ontology of [Hirsh 88]

ref: {(encapsulates lebowitz-86-technique lebowitz-86-sbl-technique)
(instantiates SBL lebowitz-86-sbl-technique)
(exhibits SBL search-property)}

def: {(R lebowitz-86-technique search-property)}

This schema corresponds to the following abstract:

> Lebowitz-86-technique encapsulates an SBL technique called Lebowitz-86-sbl-component [Lebowitz 86]. SBL has a known property called Search-property [Mitchell 81]. In this paper, I show that this property is R-related to Lebowitz-86-technique.

combination with EBL results in the search space of the pure SBL technique being decreased; for the other, the search space is increased; for the third, the search space remains the same.

[17]To avoid the tedium, from now on I will skip the description of how this connection is found.

This schema is converted to the following skeletal schema:

ref: {(encapsulates T1 T2)
(instantiates T3 T2)
(exhibits T3 Pr1)}

def: {(R T1 Pr1)}

This skeletal schema is shown in Figure 4.18. It corresponds to the following research heuristic:

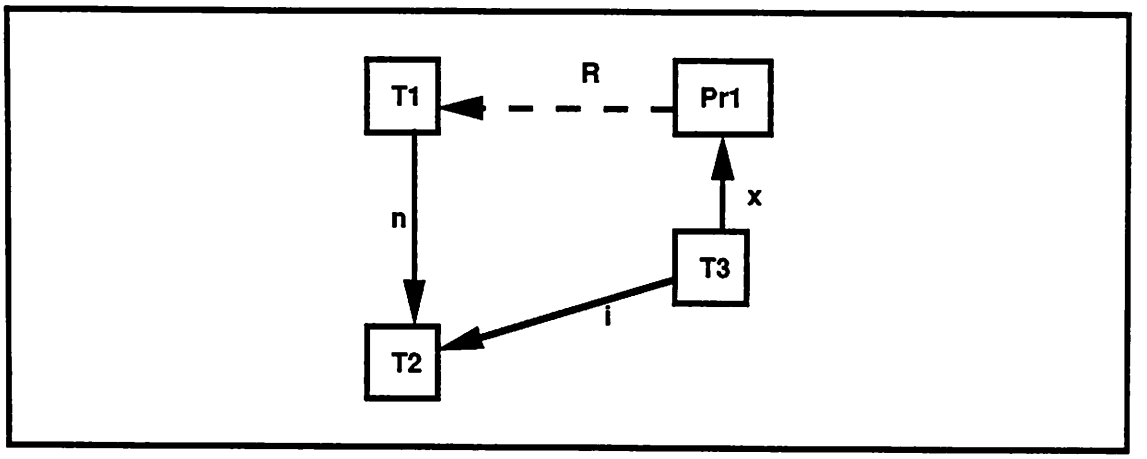


Figure 4.18: The Skeletal Schema of [Hirsh 88]

> If there exist T1, T2, T3, and Pr1 such that T1 encapsulates T2, and T2 is instantiated by T3, and T3 has a property Pr1, then suggest, "You can see how the property Pr1 applies to T1."

Since all hybrid techniques are represented as an encapsulation of two different techniques, this heuristic is tantamount to asking how a known property of one of the components (of the hybrid technique) applies to the hybrid technique as a whole[18]. Notice that this schema has a very specific **ref**: it requires T1 and Pr1 to be in a very specific relationship to each other. However, its **def** is fairly weak: it simply asks how you would relate Pr1 to T1.

[18]The following is a common sense interpretation of this heuristic: if you know that tigers have the 'stripes-property,' and that you are told that some animal in the zoo is a hybrid between a tiger and a lion (tigon?), you could ask whether it has stripes.

4.4.6 Example 6: [Rosenbloom 88]

This last example involves a schema that has more than two pre-objects in its **def**, and has a fairly specific **ref** as well as **def**. Let's look at the schema of [Rosenbloom 88] which we discussed briefly in Chapter 1. The **def** of this paper is shown below:

{(exhibits SBL rosenbloom-88-property)
(exhibits EBL rosenbloom-88-property)
(R rosenbloom-88-property search-property)}

This **def** asserts three new relations among four objects. Of the four, SBL, EBL and search-property are pre-objects and rosenbloom-88-property is a new-object. In trying to find a connection among the three pre-objects, EBL, SBL and search-property, RA finds that there is a direct relation between SBL and search-property: (exhibits SBL search-property). The shortest path between SBL and EBL (using the predicate-preference rules) involves the following two relations: (solves EBL learning-problem) and (solves SBL learning-problem). Since all the three pre-objects have now been connected, RA infers the following schema for [Rosenbloom 88]:

ref: {(solves SBL learning-problem)
(solves EBL learning problem)
(exhibits SBL search-property)}

def: {(exhibits SBL rosenbloom-88-property)
(exhibits EBL rosenbloom-88-property)
(R rosenbloom-88-property search-property)}

This schema corresponds to the following abstract:

> SBL [ref] and EBL [ref] are two techniques to solve the learning problem. SBL has a property called Search-property [Mitchell 81]. In this paper, I propose a new property called rosenbloom-88-property that is R-related to search-property. Rosenbloom-88-property applies not only to SBL but also to EBL.

This schema is converted into the following skeletal schema:

ref: {(solves T1 P1)
(solves T2 P1)
(exhibits T2 Pr1)}

def: {(exhibits T1 Pr2)
(exhibits T2 Pr2)
(generalizes Pr2 Pr1)}

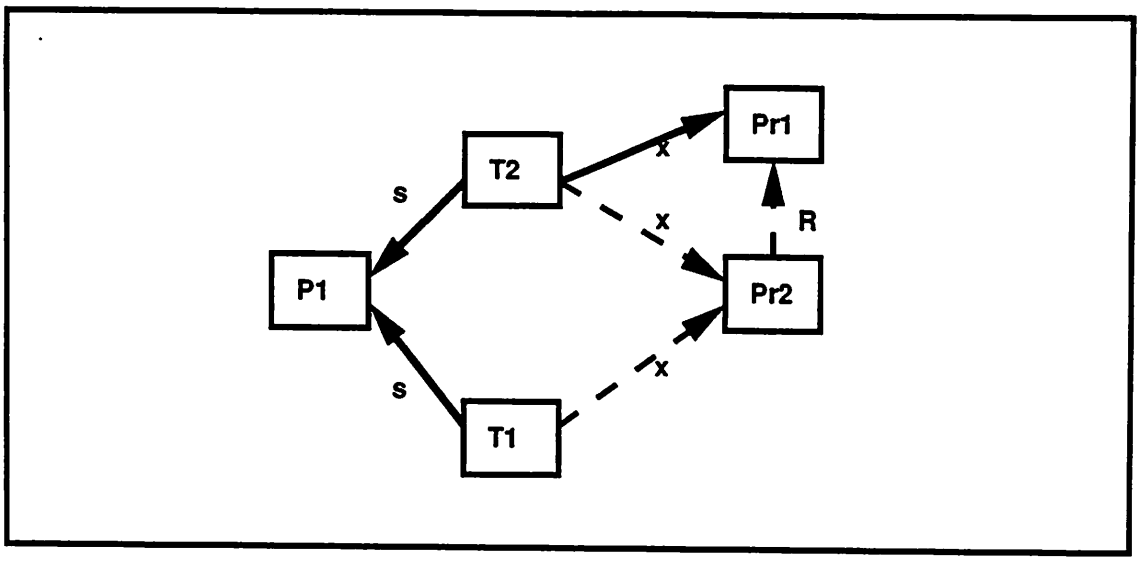


Figure 4.19: The Skeletal Schema of [Rosenbloom 88]

This skeletal schema is shown in Figure 4.19. It corresponds to the following research heuristic:

> If there are two techniques T1 and T2 to solve a problem P1, and one of them, T1, has a known property Pr1, then suggest "You could find a new property Pr2 that is R-related to Pr1 such that it applies to both T1 and T2."

The point of this series of examples is to show, conceptually, what it means to infer the **ref** of a schema. Example 1 is the simplest case, where the **def** has a single pre-object and RA finds one relation that involves that object. Examples 2 and 3 involved fairly general **refs**, but more specific **defs**. Examples 4 and 5 presented schemas with increasingly specific **refs**, but increasingly general **defs**. Finally, Example 6 presented a schema with a specific **ref** as well as a specific **def**. In all cases, the **ref** that RA infers by itself is crucial, even if it contains only one relation. For example, in the schema of Example 3 (Minsky & Papert schema), the strength of the schema comes from the **def**, i.e., the input, and the **ref** has just a single relation. However, this relation is crucial for this schema. As I pointed out earlier, unless you know that T1 solves P1, it will be ludicrous to suggest that one could see if T1 really solves P1.

Before closing this section, I would like to restate the two assumptions behind RA's learning scheme: (1) If you don't have a causal theory for a domain, assume that a set of new-relations among some known objects is not random. The new relations are probably due to some already existing relationships among these known objects. (2) In

addition, the more direct these existing relations, the more relevant they are likely to be. Relations farther removed may be purely coincidental. These assumptions enable RA to infer the **ref** by searching for a shortest path connecting the pre-objects in the **def**.

> *"It's getting quite boring," Carlo complained cheerfully as he picked up his fishing rod. "Not like Ra I, nothing to repair, no breaking timber, no ropes to splice."*
>
> —Thor Heyerdahl, The RA Expeditions.

4.5 RA II: Two Known Deficiencies

This section discusses two known deficiencies (emergent problems?) of RA's learning strategy. The first is a technical problem that is well known among EBL researchers; it is the problem of structural generalization. The second is a much deeper problem that calls into question the assumptions underlying RA's learning strategy. These two problems are best understood from the perspective of research schemas as heuristic rules.

4.5.1 Problem 1: Structural Generalization

To understand this problem, let's consider the schema of [Rajamoney 88] that we saw in Section 4.4.4. This paper combined two emergent problems of EBL into one, i.e., it proposed a new problem that was an instantiation of two emergent problems of EBL. The schema of [Rajamoney 88] is shown below:

ref: {(entails EBL incomplete-theory-problem)
(entails EBL incorrect-theory-problem)}

def: {(instantiates incomplete-theory-problem rajamoney-88-problem)
(instantiates incorrect-theory-problem rajamoney-88-problem)
(solves theory-revision rajamoney-88-problem)}

From this schema, RA obtains the following skeletal schema:

ref: {(entails T1 P1)
(entails T1 P2)}

def: {(instantiates P1 P3)
(instantiates P2 P3)
(solves T2 P3)}

When used as a research heuristic, this skeletal schema suggests that one could propose a new problem that combines two emergent problems of some technique. However, there is

nothing sanctimonious about the number 'two': one could very well combine three, four or eighteen problems into one[19]. In other words, while the constants in the (instantiated) schema have been generalized into variables, the structure of the schema stays the same. In order to learn a heuristic rule to combine, say, three entailed problems into one, the system has to see an example of a schema where three problems are combined.

This problem has been recognized with EBL systems before, and has been called the problem of 'generalizing number' [Shavlik 88]. Jude Shavlik has built a system called Bagger which analyzes an explanation in order to identify repetitive applications of a single inference rule, also called a 'focus rule.' This analysis results in a general rule that parameterizes the number of applications of the focus rule[20]. However, Bagger uses a causal theory of the domain in order to identify the pre- and post-conditions for each repeated rule application. Since RA has no causal theory available, there is no obvious way in which Bagger's structural generalization techniques can be adapted to generalize the structure of research schemas.

This paragraph outlines a possible approach to solve the structural generalization problem. To understand this approach, let's view a research schema as a labelled directed graph. If two nodes of the graph are indistinguishable from each other except for a difference in their node labels, then we have two identical (isomorphic) subgraph configurations within the research schema. This may suggest that the structure of the schema be generalized by allowing an arbitrary number occurrences of this configuration.

4.5.2 Problem 2: Negative Conditions

A second problem with RA's learning strategy is that it can only learn conditions under which a schema can apply, but not conditions under which a schema cannot apply[21]. This is a fairly deep problem and is related to the famous 'qualification problem' [McCarthy 80]. To explain this, let's consider the schema of [Rosenbloom 88] described in Section 4.4.6. To recap, Rosenbloom took a known property of SBL and generalized it to apply to both EBL and SBL. The skeletal schema of [Rosenbloom 88] is shown below:

```
ref: {(solves T1 P1)
      (solves T2 P1)
      (exhibits T1 Pr1)}

def: {(exhibits T1 Pr2)
      (exhibits T2 Pr2)
      (R Pr2 Pr1)}
```

[19]Whether that makes sense or not is a different question.

[20]Another interesting work in this area is that of William Cohen [Cohen et al 88b]. Cohen's Adept system views number generalization as the problem of learning control-knowledge for a theorem prover that constructs the explanation. By using a richer representation for expressing this control knowledge, Adept incorporates number generalization and learning from multiple examples into a single framework.

[21]No, this is not same as saying it learns sufficient conditions but not necessary conditions.

The idea behind this schema is that if there are two techniques T1 and T2 to solve a problem P1, and one of them, T1, has a property Pr1, then you could generalize Pr1 to apply to both T1 and T2. In common-sense terms, a crucial condition for the application of this heuristic is that the property Pr1 currently apply just to T1 and not to T2; only then does it make sense to suggest that you generalize Pr1 to apply to both T1 and T2. However, the schema as it was learned does not include this condition. To understand the effect of applying this schema, let's assume the following scenario: RA is presented with the **def** of [Rosenbloom 88]. RA assimilates this **def** into its ontology, and learns the above skeletal schema from this paper. The post-ontology of [Rosenbloom 88] contains the following relations:

{(solves EBL learning-problem)
(solves SBL learning-problem)
(exhibits SBL rosenbloom-88-property)
(exhibits EBL rosenbloom-88-property}

Suppose you now ask RA for possible research directions involving rosenbloom-88-property: since the relations above will satisfy the applicability conditions of the schema learned from [Rosenbloom 88], one of RA's suggestions would be "You could generalize rosenbloom-88-property to apply not only to SBL, but also to EBL." But rosenbloom-88-property already applies to both EBL and SBL.

This is an inherent problem of RA's learning strategy. This strategy rests on the assumption that the new relations among the pre-objects are probably due to the already existing relationships among them. However, these new relations could also require that certain currently non-existing relationships among these pre-objects not exist among them! Since RA does not have a causal theory of the domain, there is no way for RA to learn about relevant but non-existing relations.

This problem is related to the 'qualification' problem, also called the 'potato-in-the-tail-pipe' problem. Suppose you see somebody starting a car by turning the key in the ignition while holding the throttle down. From this you would learn that the conditions that enable this action (of starting the car) are 'turning the key' and 'holding down the throttle.' However, there are several other conditions that are also necessary for starting the car: in particular, it is necessary that there not be a potato in the tail pipe of the car! The qualification problem refers to the difficulty of stating every possible condition under which a particular plan applies.

The situation is similar with RA's learning. RA assumes that the explicit conditions that existed when a particular action was performed (i.e., when a particular paper is assimilated) are the only necessary conditions for that action. However, there could be some non-existing conditions (such as there not being a potato in the tail pipe or there not being a relation between search-property and EBL) that are also part of the enabling

conditions for an action. At the time of this writing there is no obvious way in which RA could avoid this problem without recourse to causal knowledge of the domain.

4.6 Summary

The learning strategy consists of two phases: the assimilation phase and the generalization phase. During the assimilation phase, RA infers the **ref** of a paper from its **def**; during the generalization phase, it generalizes this **ref**, **def** pair into a skeletal schema by replacing constants by typed variables.

Two assumptions enable RA to infer a paper's **ref** from its **def**: (1) In the absence of a causal theory, assume that the new relations among known objects are not random — they were probably due to some already existing relationships among these objects; (2) further, out of all possible existing relationships, the most specific ones are probably the most relevant.

This chapter illustrated RA's learning strategy through several examples. These examples were chosen to illustrate several different kinds of research schemas, i.e., schemas with various combinations of strong and weak **ref** and **def**.

A significant aspect of RA II's learning technique is that it fits neatly within our model which sees scientific research in terms of knowledge evolution. Put another way, research schemas are an adequate representation for the acquisition of new research schemas.

Despite superficial similarity, RA's learning is quite different from EBL. RA does not construct an explanation from general (uninstantiated) domain knowledge. It accesses (instantiated) knowledge directly from memory in order to find the most specific knowledge that connects the various objects in the input.

RA's generalization strategy is also different from EBL's generalization strategy. Since RA does not construct an explanation from domain theoretic rules, it would appear that there is no basis for RA's generalization. RA's generalization works for a deeper reason having to do with the category structure of RA's world. This is discussed in the next chapter.

There was a curious aura of mystery here, something unwritten and unsaid that compelled attention, conjecture. This was not the time to be content with preconceived ideas.

—Thor Heyerdahl, The RA Expeditions.

Chapter 5

Analysis

The voyage itself was intended as an experiment, a study trip into the dawn of civilization. But there was room for an experiment within the experiment. A study trip into a crowded, overpopulated tomorrow.

—Thor Heyerdahl, The RA Expeditions.

There are some unusual things about RA's learning strategy: RA generalizes a fact about a set of specific objects in a research schema into a general fact about all objects that belong to certain categories or types such as problems and techniques. In SBL-style learning, multiple examples and some form of encoded bias provide a basis for generalization. In EBL-style learning, the causal knowledge of the domain provides a basis for generalization. RA does not use either. So, what is the basis for RA's generalization?

While some of the assumptions about RA's assimilation phase sounded reasonable, is there any justification for these assumptions? Why should **ref** include only the most specific relationship among the pre-objects in the **def**? For example, research papers, particularly journal papers, include lot more background knowledge than the 'most specific information' in the **ref** of a research schema. Are there any reasons behind RA's assumptions regarding the structure of research schemas?

To answer these questions, we turn to the so-called basic-level of abstraction. This idea is a profound one. It started with the writings of Roger Brown, and culminated in the empirical demonstrations of Eleanor Rosch. The basic-level idea states that, of the various levels of abstraction from which objects can be conceptualized, there is a primary or a 'basic' level of abstraction that is neither too general, nor too specific. A dog is primarily a dog before it is a mammal or a poodle; a chair is primarily a chair before it is a furniture or a rocker. Of all levels of abstraction, the basic level has the most semantics unique to its level.

I define two terms, *associativity* and *discriminability* to characterize the basic level. Associativity refers to the generality of a description, and discriminability to its specificity. The basic-level, then, is the level of description that is both associative and discriminative.

RA's learning strategy can be justified in terms of basic levels: RA generalizes the constants in research schemas into variables of the categories that are both associative and discriminative. The assimilation phase of RA is also justified in terms of basic levels: the structure of the **ref** ensures that a research schema, as a whole, is both associative and discriminative.

This analysis gives rise to some interesting speculations: Is basic level the universal bias in generalization? Do basic level relational structures exist?

Further, I show that the basic-level effect is not a 'mere psychological fact,' but is in fact a design principle. I provide a case study of three Case-Based Reasoning systems and show that their design is motivated by the need to categorize their world into categories that are both associative and discriminative. Table 5.1 contains a guide to Chapter 5.

Section	On first reading	Description
1	read	Describes why some aspects of RA's learning strategy are surprising.
2	read	Describes categorization and basic levels. If familiar with these, read Section 5.2.5.
3	read	Relates to RA. Explains RA's generalization and assimilation in terms of associativity & discriminability.
4	read	Speculations about these ideas as they apply to KR. Do basic-level relational structures exist?
5	skip	Case study of three AI systems. Explains their design in terms of associativity & discriminability.
8	read	Summary.

Table 5.1: Guide to Chapter 5

5.1 Two Questions

As we saw in Chapter 4, RA's learning scheme consists of two phases: During the assimilation phase, RA infers a paper's **ref** and hence its instantiated schema; during the generalization phase, RA generalizes the instantiated schema into a skeletal schema by replacing the constants by *typed* variables. Each of these phases has a surprising feature. The two subsections of this section discuss these features and explain why they are surprising.

5.1.1 Question 1: Generalization

Let's first consider the generalization phase. After seeing a research paper that involves a specific set of objects (constants), RA assumes that the paper embodies a general research strategy that is applicable to any object that belongs to certain categories (types). What is the basis for this inductive leap? To understand this question, let's quickly review the notion of 'inductive bias' and how generalization is performed in SBL and EBL systems[1]:

- **Inductive Bias**: Whenever you generalize a specific situation into a general situation, you need some basis for this generalization. A specific situation can be generalized in a number of different ways, and your criteria for choosing one over several possibilities is known as your 'bias' [Mitchell 80] [Utgoff 84]. For example, suppose you see a boy beating a dog. Based on observing this specific situation, you might make a generalization, "Boys beat dogs." The same situation might also have been generalized as "Children beat dogs," "Children torture animals," "Humans torture animals," or "Boys beat dogs on Sundays and girls beat dogs on Wednesdays." Each of these generalizations involves a different 'bias.'

- **Similarity-Based Learning**: In SBL-style generalization, the system designer encodes the system's bias. For example, an SBL system might be provided an abstraction hierarchy as follows: boys and girls constitute children, men and women constitute adults, and children and adults constitute humans. If the system first sees a situation in which a boy beats a dog, it makes a tentative generalization, "Boys beat dogs." When it also sees a girl beating a dog, it moves up the hierarchy one level, and makes the generalization, "Children beat dogs." If it then sees a man beating a dog, it will move all the way up the hierarchy to 'humans' in order to include boys, girls, and men, and make the generalization "Humans beat dogs." Such a generalization may be expressed as "If x is a human and y is a dog, then x beats y." In SBL-style learning, the abstraction hierarchy and the multiple examples together constitute the basis for generalization.

- **Explanation-based Learning**: In this style of generalization (see [Mitchell et al 86] [DeJong & Mooney 86]), one has a causal theory of the domain. Hence you know the association between boys' psyches and dog-beating behavior, the association between the boys' parents' psyches and the boys' psyches, the association between childhood trauma and dog-beating, and on and on. When you see a boy beating a dog, you crank your inference engine to explain why this particular boy beats this particular dog. This explanation might include facts like "this boy is being ignored at home, and this causes irritability; irritability causes a desire to inflict injury on others who cannot fight back. A dog cannot fight back. Hence the boy beats the dog." This explanation

[1]For an extended description, see Section 5.4.4.

is then generalized to any situation so long as the explanation structure holds. For example, the above explanation may be generalized as "any x who is being ignored at home will inflict injury upon any y that cannot fight back." In EBL-style learning, the causal knowledge of the domain provides the constraints for the variables in the generalization; hence this causal knowledge, also called a domain theory, constitutes a basis for generalization.

When we look at RA's learning strategy, we find that there is no apparent basis for RA's generalization. Since RA learns a research schema from a single example, the basis for generalization cannot be the interaction among multiple examples as in SBL. Since RA does not use a causal theory to construct an explanation, the basis of RA's generalization cannot be any causal knowledge either. So what is the basis for RA's generalization[2]?

The answer to this question turns out to be interesting. Of all possible categories (types) in RA's abstraction space, the categories problem, technique, property and concept are the most differentiated categories: these categories exhibit two properties called *associativity* and *discriminability*. The categories at higher levels of abstraction have lower discriminability and the categories at lower levels of abstraction have lower associativity. RA generalizes an instantiation to a level that is both associative and discriminative, so that an instantiation is placed into a category that contains all intrinsically similar objects and no intrinsically dissimilar objects. Section 5.2 introduces the idea of categorization and basic-levels of abstraction, and defines the terms associativity and discriminability.

5.1.2 Question 2: Assimilation

In some ways, there is no need for RA to learn any research heuristics. It already knows all possible heuristics that are expressible using its types and relations. To understand that statement, let's look at RA as follows: Initially, even before any paper has been input, RA knows a set of heuristics. These are precisely the type constraints on RA's various relations, such as "A problem can be solved," "A technique can solve a problem," "A technique can exhibit a property," and so on. Hence, if there is an object of the type problem, RA can in fact use these heuristics to suggest "Why don't you propose a technique to solve this problem?" Since all research heuristics expressible in terms of RA's types and relations are simply configurations of these simple heuristics, in some sense, these are the only heuristics one needs.

However, as you can see, the heuristics corresponding to the type constraints are extremely general and weak. As RA assimilates more papers into its memory, it acquires

[2] Btw When I first built RA II, I did the 'intuitively' obvious thing by replacing the constants by typed variables without thinking twice about it. Only later did I realize that this generalization was somehow counter to conventional wisdom. This chapter is as much an analysis of my intuition as it is an analysis of RA's generalization scheme.

a set of more specific research heuristics. The question is, when do we stop? How do we know that each paper does not employ an entirely new heuristic unlike anything else?

To make this question more concrete, let's consider some examples. Figure 5.1 shows five configurations of **refs** and **defs**[3]. Let's first consider configurations A and B. These two configurations correspond to the following heuristics:

> *[A]* If there is a technique T1 to solve P1, then one could look for an emergent problem P2 of T1. Then one could propose a new technique T2 to solve P1 while avoiding P2.

> *[B]* If there is a technique T1 to solve P1, then one could look for an emergent problem P2 of T1. Then one could propose a new technique T2 to solve P2.

Compared to the general and weak heuristic (say W), "A problem can be solved with a technique," these two are more specific: in A, the technique T2 solves P1 while avoiding P2, whereas in B, T2 solves the emergent problem P2. Even though both A and B embed the weaker heuristic W, they are worth learning because they discriminate between two inherently different research strategies.

Let's now consider the configurations C, D and E. C and D both include the configuration A, and E includes C. The configurations C and D may be described as follows:

> *[C]* If P1 is an instantiation of two problems P4 and P5, and there is a technique T1 to solve P1, then one could look for an emergent problem P2 of T1. Then one could propose a technique T2 to solve P1 while avoiding P2.

> *[D]* If a problem P4 is solved by a technique T3 which instantiates a technique T1 which solves a problem P1, then one could look for an emergent problem P2 of T1. Then one could propose a new technique T2 to solve P1 while avoiding P2.

If A and B are different research strategies despite containing the configuration '(solves T P)', are C and D different research strategies despite containing the configuration A? Is E an even more specific and different research strategy than A and C? When do we stop making these discriminations?

The answer to this question also turns out to be interesting. The assumptions behind the assimilation phase ensures that a research schema, as a whole, is both associative and discriminative. Therefore, A and B are inherently discriminable research strategies that are worth learning (despite the fact that they embed the weaker research heuristic that a

[3]Relations in ref are, as usual, shown in solid lines, and relations in def are shown in dashed lines.

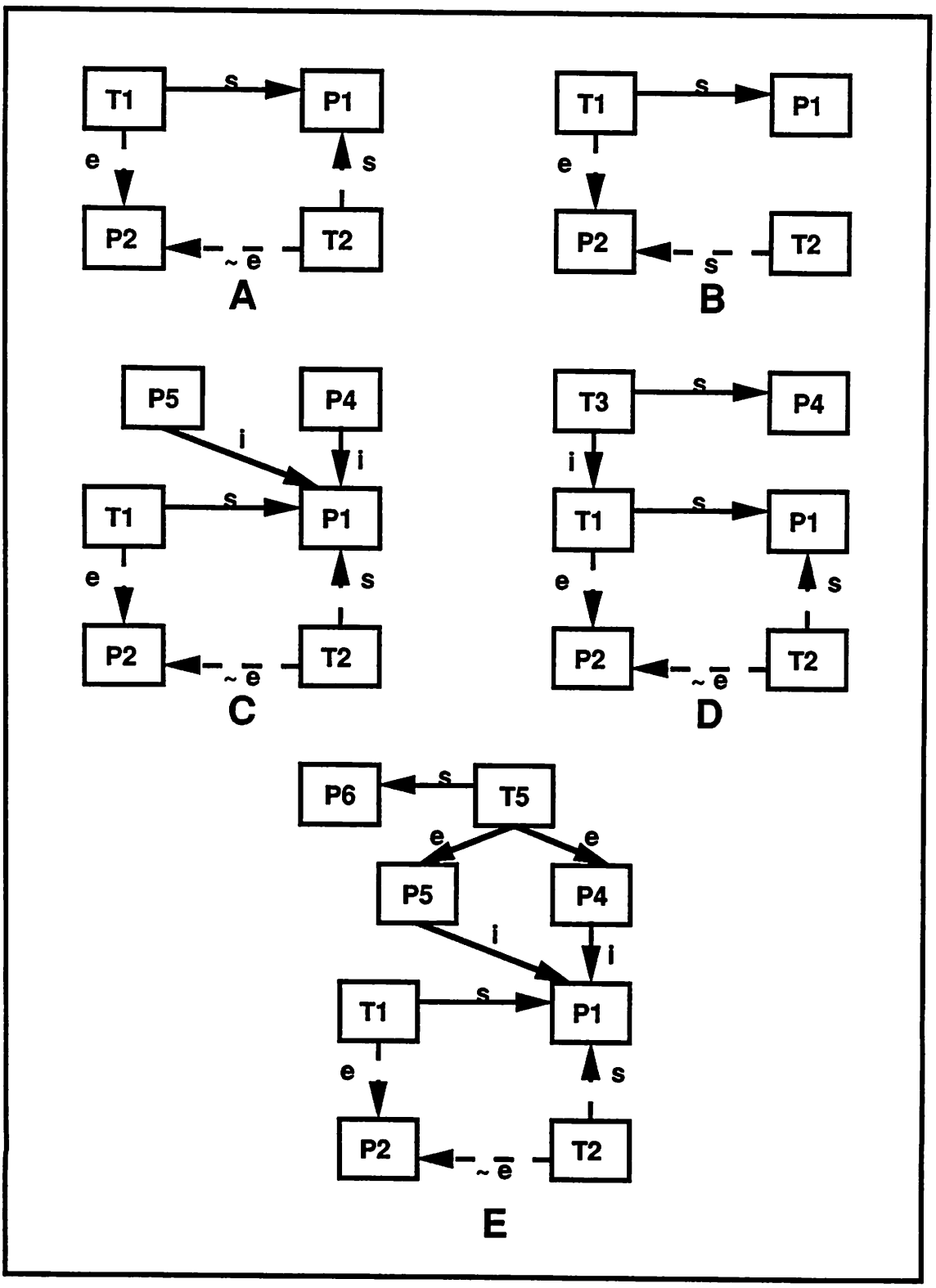


Figure 5.1: Five ref, def configurations

problem can be solved by a technique), whereas C, D, and E are no more discriminative than A; however, they are all less associative. The next section introduces the idea of categorization, basic levels of abstraction, and defines the terms associativity and discriminability.

5.2 Categorization

Suppose you've been bitten by a dog once. The next time you see a creature that you believe to be a dog, you may decide to avoid it, or decide to arm yourself with a big stout stick. By grouping objects into categories (such as dogs), you are able to transfer your experience from one member of a category to others. This saves you the trouble of having to learn the properties of each object in the world separately, thereby resulting in a great deal of cognitive economy. Categorization refers to the process of dividing the world into categories.

An important philosophical debate over the centuries has been over the question whether categories are determined by the world or by the mind. Alternately, do categories exist in the world in some immutable form, or do humans, by social convention, cluster the world in basically arbitrary ways? To the extent that several properties (intensions) can be written down as formulas that have nothing to do with humans and human cognition, the categories (extensions) corresponding to these properties exist in the world independently of humans. Therefore, categories are determined by the world.

However, there are also cases where categories are basically arbitrary: the standard example used to support this argument comes from the two different approaches to devising a 'correct' biological taxonomy. In his essay, *What if anything is a zebra?*, Stephen J. Gould describes the biological taxonomy preferred by the pheneticists and the cladists [Gould 83]. The pheneticists' taxonomy is determined by overall perceptual similarities and inter-breeding possibilities. This taxonomy does include the common folk categories like zebras and fish. In contrast, the cladists' taxonomy is determined solely by evolutionary history and branching order. Gould writes:

> I regret to report that there is no such thing as a fish. About 20,000 species of vertebrates have scales and fins and live in water, but they do not form a coherent cladistic group. Some — the lungfish and the coelacanth in particular — are geneologically close to the creatures that crawled out on land to become amphibians, reptiles, birds, and mammals. In a cladistic ordering of trout, lungfish, and any bird or mammal, the lungfish must form a sister group with the sparrow or elephant, leaving the trout in its stream... A coelacanth looks like a fish, tastes like a fish, and therefore — in some legitimate sense beyond hidebound tradition — *is* a fish[4]. Unfortunately,

[4] BIW It is unlikely that Gould has ever tasted a coelacanth. Coelacanths are a very ancient deep-water fish known to us from the fossil records and at the time Gould's article was written, believed

> these two types of information — branching order and overall similarity — do not always yield congruent results (p. 363).

Brent Berlin, a cultural anthropologist, spent several years among the Tzeltal and the Aguarana tribes in Central America studying their classification of plants. He found that their folk taxonomy corresponded very well with the Linnean scientific taxonomy at the level of a genus, but at higher and lower levels of abstraction (at the life form or species levels), the two systems had little similarity (see [Berlin 78] [Lakoff 87]). The work of Eleanor Rosch has gone much farther to show that at a certain level of abstraction (called the 'basic-level'), the world is very well differentiated — i.e., the categories have a great deal of internal cohesion, and are very well separated from other categories [Rosch et al 76]. Hence, at this level of abstraction, the categories of the world are also the categories of the mind; at other levels of abstraction, the clustering of the world into categories is essentially arbitrary. To understand the importance of these arguments and results, let's start with the classical view of categories.

5.2.1 Classical View

In their comprehensive survey of the various theories of categorization, Edward Smith and Douglas Medin [Smith & Medin 81] used the term *Classical View* to refer to the traditional view of categories held by philosophy, mathematics, and I may add, Artificial Intelligence. This view is based on the tenet that every intension has an extension — i.e., every property defines a category of objects that exhibits that property. In other words, a classical category is defined by necessary and sufficient conditions for category membership. This view is identical to the mathematical view of sets defined through characteristic functions. Objects satisfying the necessary and sufficient conditions are instances of the category, and others are non-instances. Hence, a classical category is characterized by a crisp boundary that separates its instances from its non-instances; within the category, any instance has an equal status of membership as any other. For example, the category *prime* numbers is defined by the following formula: $prime(x) \Leftrightarrow divisors(x) \equiv \{1, x\}$. All numbers satisfying this condition are members of this category, and every member of this category satisfies this condition.

The classical view has little to say about category taxonomies. Typically, set inclusion determines the level of abstraction of a particular category within a taxonomy of categories. Thus, 'mammal' is at a higher level of abstraction than 'dog,' which in turn is at a higher level of abstraction than 'poodle.' Property inheritance works from the top downwards with lower level categories inheriting all the properties of higher level categories.

---

to have been extinct for at least 60 million years. In 1978, a coelacanth was caught by a fisherman in South Africa. A subsequent expedition found coelacanths quite alive and swimming in the Indian Ocean off the coast of Comoro Islands [Fricke 88].

The classical view is generally attributed to Aristotle. For an excellent review of the classical view, see chapter 3 of Smith and Medin's *Categories and Concepts* [Smith & Medin 81]. Within AI, the classical view is taken for granted in most work on knowledge representation [Brachman & Levesque 85] and machine learning [Michalski et al 83].

5.2.2 Challenges to the Classical View

Wittgenstein was perhaps the first to raise an objection to the classical view of categories. In [Wittgenstein 53], he considered the category 'game' and argued that while this is an honest-to-god category, one cannot give necessary and sufficient conditions for something to be a game. Some games like chess involve competition; some games like Ring-around-the-rosie involve pure amusement; yet others like poker involve chance; most games involve multiple participants, yet a few, like Solitaire, involve just one person. Wittgenstein proposed that categories are characterized by 'family resemblances.' Suppose some instance x is a prototypical member of a category C. Then an instance y that is very similar to (i.e., has a high degree of family resemblance to) x is also likely to be included in C. Since y is a member, an instance z that is similar to y is now likely to be included in C and so on. As we get farther and farther away from x, the question of membership (in category C) becomes hazier until we reach an instance that is clearly not a member of C[5]. Wittgenstein's proposal has been borne out by Rosch's demonstration of the 'prototypicality effect' [Rosch & Mervis 75].

In a paper titled *How shall a thing be called?*, Roger Brown first speculated that objects have a primary category affiliation: of all names that one can given an object, a particular name, at particular level of abstraction, has a superior status [Brown 58]. Such "real names" are shorter and used more often. Brown also speculated that these are the names that children first acquire; then they proceed to superordinate and subordinate categories by "achievements of imagination." Brent Berlin, in his study of folk biological taxonomies found that the basic-level of classification was the level of a 'genus.' This level had primary lexemes to name the plants and trees in the language. Subordinate levels had derived names, and (frequently) superordinate levels were unnamed [Berlin 78]. Alyssa Newport and Ursula Bellugi, in their study of the the American Sign Language (ASL) found that basic-level categories such as chairs and shirts and screwdrivers had unanalyzable hand configurations, whereas subordinate categories were created from basic-level categories using modifiers; once again, superordinate categories (frequently) did not have separate hand configurations [Newport & Bellugi 78]. Empirical results from a wide range of phenomena support the claim that there is a 'basic' level

[5] Biw Biological speciation of this sort occurs in nature. When a species spans a large geographical area, each locale may develop its own slight variations over time. Thus members at locale x can interbreed with members at an adjacent locale y, which in turn can interbreed with members of locale z, but x and z may not be able to interbreed [Mayr 84]. Further, such chains may also form rings, with the terminal populations of a chain coexisting in the same habitat as two distinct species [Stansfield 77].

of abstraction at which natural category cuts occur; objects in the world are primarily affiliated to categories at this level. This so-called 'basic-level effect' is discussed below.

5.2.3 Basic Level Effect

Using a comprehensive set of experiments with human subjects in a laboratory setting, Eleanor Rosch demonstrated a wide variety of psychological phenomena that are collectively known as the 'basic level effect' [Rosch et al 76]. These results are summarized below[6]:

- Reaction Time: Subjects identify objects as members of their basic level category faster than they can identify them as members of super- or sub-ordinate categories. For example, an object is identified as a chair faster than it is identified as a furniture or a kitchen chair. Rosch et al write: "We may speculate that after identification of the basic class of an object, superordinates are derived by inference from the class membership of the basic object, and that subordinates are derived from observation of attributes — additional to those needed to perceive the basic object — which are relevant to subordinate distinctions" (p. 414). This is a radical departure from a hierarchical view of knowledge organization, where inheritance is always assumed to flow downward from the most abstract category.

- Children's language acquisition: Categories at this level are the first categories named and understood by children. Further, children as young as 3 years can sort objects into basic-level categories almost perfectly (96% accuracy), but perform poorly (55% accuracy) at sorting tasks that involve superordinate categories.

- Freenaming: When asked "What is X?" and shown a picture of some object X, subjects will overwhelmingly choose a basic-level name.

- Cognates in Languages: Basic level categories are the most necessary categories in languages. The American Sign Language has fewer signs for concrete objects than does spoken English, although the ASL speakers share the same environment as do hearers-speakers of English. Rosch et al. found that basic-level terms were almost as common in ASL as in spoken English, whereas super- and sub-ordinate names were significantly less common. This result is consistent with Newport's study of ASL as well as Berlin's study of folk biological taxonomies described above[7,8].

[6] In addition to Rosch et als's results, this list also lists some results that have been reported by others. These are expliitly cited.

[7] [Biw] It has been reported that, in German, basic-level categories have gender, whereas superordinate categories are gender-neutral [Zubin & Kopcke 86].

[8] [Biw] If basic-level categories are indeed the basic category cuts across cultures and languages (and most reliably and consistently named categories), then these categories should perhaps form the cognates for machine translation systems. Ponder that one!

- Visualization and Sketching: The basic level is the highest level at which subjects can visualize and sketch objects; for example, subjects can visualize and sketch a car or a shirt, but not a vehicle or clothing.

- Recognition: The basic level is the highest level at which subjects can identify the category from the averaged shape of the category members; for example, subjects can recognize a car or a screw-driver from the averaged shape of the various members of the category; however, subjects cannot recognize the category 'vehicle' from the averaged shape of all vehicles, or the category 'clothing' from the averaged shapes of various pieces of clothing[9].

- Motor Actions: The basic level is the highest level at which motor actions involved in our interaction with objects of a category are similar. For example, the motor movements involved in some common actions involving all chairs are similar; however, the motor actions involved in interacting with all furniture are not.

- Neutral Contexts: This is the level at which category names are used in neutral contexts. For example, "there is a dog on the porch" can be used in a neutral context, but "there is a wire-haired terrier on the porch" or "there is a mammal on the porch" require special contexts [Cruse 77][10].

- Abstract Categories: Beth Adelson has demonstrated some of the basic-level effects for abstract categories used by computer scientists. She showed that lists, trees, sorting, searching etc. are basic level categories; algorithms and data structures are super-ordinates; linked-lists, and heapsort are sub-ordinates [Adelson 85].

- Information Content: The basic level corresponds to the level that has the most semantics. In one experiment, subjects were given category names from various levels of abstraction and were asked to list all features that they can think of for each category. The result, averaged over several categories, shows that super-ordinate categories have few features; basic level categories have a large number of features; sub-ordinate categories have few new features. The number of features listed by the subjects for super-

[9] Btw This is not surprising because super-ordinate categories are characterized by functional attributes whereas basic-level categories are characterized by perceptual attributes. The attribute common to all pieces of clothing is that 'you wear it,' whereas a shirt has arms, pockets, collar and so on. Hence a superordinate category is basically a disjunction of a set of basic level categories. In fact, Smith and Medin speculate that superordinate categories are perhaps the only disjunctive categories in the world [Smith & Medin 81].

[10] Btw Basic-level cognates appear to be the default for normal discourse. Whenever a discourse shifts to other levels, there is a shift in the focus of interest. In analyzing question-answering, Wendy Lehnert argues that in the question, "Why did John go to McDonalds?" the focus is on McDonalds, whereas, in the question "Why did John skateboard to McDonalds?" the focus is on skateboarding [Lehnert 78]. For a computational approach to using basic-level cognates in discourse, see [Peters & Rapaport 90].

ordinate, basic, and sub-ordinate categories are shown in Figure 5.2[11]. The transition from superordinate to basic has the steepest slope, indicating that most information about objects is learned at the basic level, and very little new information is learned by going from the basic to the subordinate level. Rephrased, the super-ordinate category has little semantics, and the sub-ordinate category has little extra semantics over and above the basic-level. Thus, there is very little you can say about all furniture in general, there is a lot you can say about chairs or tables, but very little else about a kitchen chair or a work table.

In their classic paper describing the basic-level effect, Rosch et al write: "The aim of the present research is to show that the world does contain 'intrinsically separate things.' The world is structured because real-world attributes do not occur independently of each other. Creatures with feather are more likely also to have wings than creatures with fur, and objects with the visual appearance of chairs are more likely to have functional sit-on-ableness than objects with the appearance of cats. That is, combinations of attributes of real objects do not occur uniformly. Some pairs, triples, or ntuples are quite probable, appearing in combination with one, sometimes another attribute; others are rare; others logically cannot or empirically do not occur."

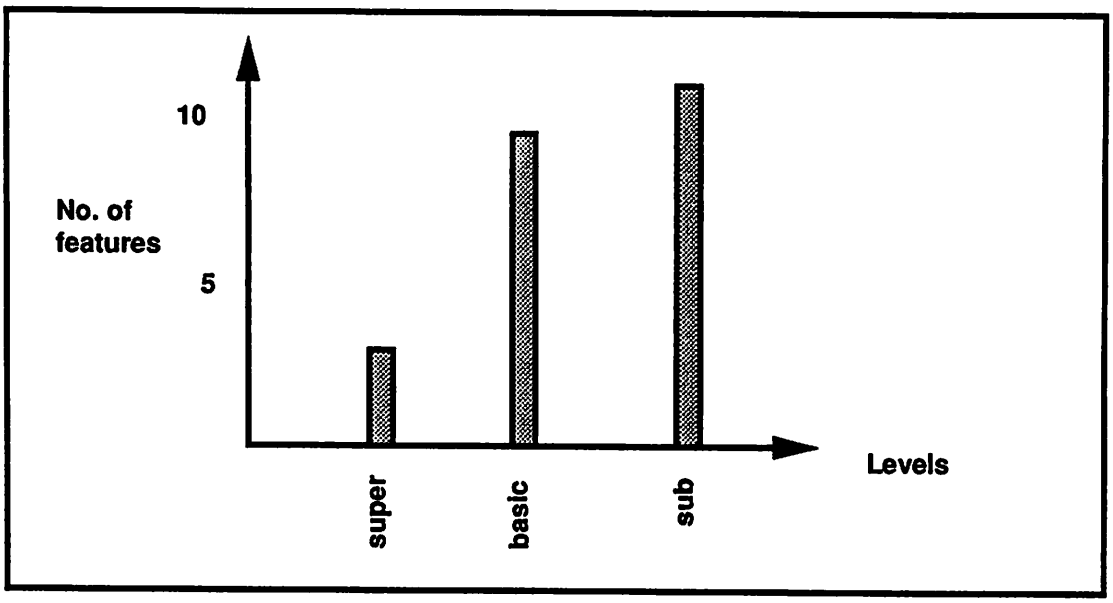


Figure 5.2: Abstraction Levels vs Number of Features

[11]This graph was generated by averaging the data for all non-biological categories listed in [Rosch et al 76].

The basic-level effect has been more-or-less completely ignored by AI[12]. Only in the last few years, researchers in machine learning have attempted to use a statistical measure called 'Cue Validity' that is related to basic levels in their clustering algorithms [Fisher 86] [Aha & Kibler 89]. See Pat Langley's editorial in *Machine Learning* urging the field to pay attention to these results from psychology [Langley 86].

The basic-level effect should be seen not as a 'mere psychological fact' but as a design principle: After showing how it applies to RA, I will provide a case study of three case-based reasoning systems to demonstrate that their design can be explained in terms of the basic-level effect[13].

5.2.4 Interpreting Basic Level

Every object in the world is, in some sense, entirely unique. If we treated each object as a totally unique entity, two problems ensue: (a) we have an infinite number of objects to record and remember and (b) we cannot transfer our experience from one object to another. By grouping objects into categories, we keep the number of items to a manageable level, and further, we can predict the properties of previously unseen objects by knowing their category names.

If you thought that makes sense, think again. The number of categories that can be generated for n objects is 2^n. Hence, categorization is no panacea, unless some constraints are imposed. The basic level effect provides some constraints on which categories are useful.

I have an object — let's call it G001 — in mind. This object is made of glass, has a volume of about 50cc, weighs about 100 grams, has beveled edges, has about 8 little holes on its lid, and contains salt. It usually sits on my dining table. I take this object with me when I go camping. If I fling this at you in a fit of rage, it could hurt you. If I drop it on the floor, it might break. This object belongs to several categories: 'thing to take on a camping trip', 'breakable object', 'thing to throw at people if I want to hurt them', 'salt-shaker' and so on. Still, of all these possible categories, 'salt-shaker' is somehow its

[12]Neither the AI handbook (4 volumes), nor the knowledge representation bibliography in [Brachman & Levesque 85] has any reference to Rosch's work.

[13]Btw Bibliographic note: Lakoff's *Women, Fire and Dangerous Things* is an excellent treatment of the basic-level effect. Lakoff discusses the cognitive, philosophical as well as the linguistic implications of these results. Howard Gardner refers to the view of categorization suggested by the prototypicality and basic level effects as the *Natural View* of categorization [Gardner 85]. Gardner's book (Chapter 12) summarizes the natural view within a historical context. Also see Rosch and Lloyd's *Cognition and Categorization* [Rosch & Lloyd 78]. Scholnick's edited book *New Trends in Conceptual Representation: Challenges to Piaget's Theory?* contains several papers that discuss the imlications of the natural view for developmental psychology [Scholnick 83]. See Murphy and Medin's paper on an expansion of these results, and for the argument that basic-level and prototypicality effects, while providing some constraints, do not completely explain why certain categories are 'coherent' and others are not [Murphy & Medin 85].

real category, because most of its properties are predictable by the name 'salt-shaker'; all other categories can in fact be derived by knowing that it is a salt-shaker. The category 'thing to take on a camping trip' is dynamically constructible from one's knowledge of salt-shakers and camping; the category 'breakable object' is derivable from the fact that most salt-shakers are made of glass; the category 'thing to throw at people if I want to hurt them' is derivable from the fact that it is a fairly small rigid object (hence I can grasp it, and throw it with some force) that's quite heavy for its size. The features that co-occur to make G001 a salt-shaker capture the essence of G001; hence this category is likely to be the most useful category for this object in most contexts.

Such primary categories are characterized by feature co-occurrences, because feature co-occurrences reflect deep causalities. For example, the height of a human and the size of his/her head are not entirely independent features. Heights and head-sizes, within a narrow range of variation, co-occur. My head-size is determined by the size of the pelvic bones of the females of my species, which in turn is determined by their height and balance requirements. As a male, my height cannot be completely independent of the heights of the females of my species (for whatever evolutionary reasons), and hence my height and head-size are, for some deep reasons, correlated. You are unlikely to see an object that is identical to me on one-hundred features, but is seventy-eight feet tall. Categories determined by feature correlations reflect these deep causalities, and hence the 'essence' of the objects that belong to these categories. In contrast, categories such as 'things to take on a camping trip' are defined intensionally, and pretty much the only thing that members of such concepts have in common is their intensional definition. In the next subsection, I will define two terms, *associativity* and *discriminability* in order to characterize basic level categories.

5.2.5 Associativity and Discriminability

For the moment, let's forget about features and categories and talk about 'descriptions.' A description can be a feature, a feature cluster, or a category name. We will say that a description for some object S is highly *associative* if it matches or recalls most similar objects. Similarly, we will say that a description C for some object S is highly *discriminative* if it does not match or recall any dissimilar object[14].

To illustrate, let's take a particular chair and name it G111. This chair can be described in a number of different ways — as G111, as a rocker, as a chair, as a piece of furniture, or as a thing. The description G111 describes it as an entirely unique object — i.e., it discriminates it from every other object in the world. Since this description does

[14]I originally defined these two terms in [Swaminathan 88b] while discussing the properties of indexing mechanisms for Case-Based Reasoning systems. Most of the material in Section 5.5 is an expansion of that discussion. When my paper was published, I was unfamiliar with the basic level effect. The relationship between case-based indexing and basic levels has since been pointed out by others as well, see for example [Fisher et al 90].

not match or recall any other object, and therefore not any other 'dissimilar' object, we will say that this description has high discriminability. However, this description does not associate G111 with any other objects in the world, some of which are likely to be similar. Hence, we will say that this description has low associativity.

In contrast, let's take a description such as 'thing.' This description associates G111 with practically everything in the world. Therefore, if the world contains things that are similar to G111, this description most definitely matches or recalls those objects. Hence we will say that the description 'thing' has high associativity — it associates G111 with all similar things. However, this description also matches or recalls a large number of objects that are dissimilar to G111 — 'thing' does not discriminate G111 from all the objects that are dissimilar to it. Hence we will say that this description has a low discriminability.

So far, I have shown that there is a tradeoff between associativity and discriminability: the more associative a description, the less discriminative it is and vice versa. In the classical view of categories, there is always a tradeoff: a particular description is good for some purposes and bad for others; you choose the best one for your given purpose. Thus salt-shaker is a good description in some contexts and 'thing to take on a camping trip' is good in others. What the basic level effect demonstrates is that there are levels of description that maximize *both* associativity and discriminability, or at least provide an *optimal* tradeoff between the two. Since the features of objects are not uniformly distributed, at some levels of abstraction, feature clusters co-occur. The categories circumscribed by these co-occurring feature clusters are very well-differentiated: they group intrinsically similar things in the same category and intrinsically dissimilar things in different categories. In other words, such categories contain *all* instrinsically similar things, and *no* instrinsically dissimilar things. Hence a description of an object in terms of these categories is a good one with respect to most purposes: such a description associates the object with all the similar things and none of the dissimilar things. For example, a description of G111 as a 'chair' matches most objects with which G111 shares a significant number of features, namely all other chairs; hence 'chair' is a highly associative description. Further, this description also fails to match or recall other objects, particularly tables and other pieces of furniture, with which G111 shares few features; hence it is also a highly discriminative description. Therefore, 'chair' is the best description for this object. The basic level effect demonstrates that the primacy of such descriptions is psychologically real: given an object, subjects will describe the object in terms of its basic level category[15].

[15] Btw Traditionally, basic-level categories have been discussed using the notions of predictiveness and predictability of features — i.e., whether a category name predicts that an object of that category will have a certain feature, and whether a certain feature predicts that objects having that feature belong to a certain category. Rosch uses these terms frequently, and the so-called 'Cue-Validity' measures use these notions to identify the basic level categories (see [Gluck & Corter 85]). Lebowitz used these terms in the context of learning generalizations, though without reference to basic-level categories [Lebowitz 86].

- The *coding* of a stimulus S is some description of the stimulus. This description can be a single feature such as 'animate', a category name such as 'animate being' or 'cat', or a feature cluster such as '{animate, four-legs, says-meow}'. I also use the phrase 'description of an object' as a synonym for 'coding of a stimulus'.
- A coding C for a stimulus S is highly *associative* if it associates S with most of the other intrinsically similar stimuli. Alternately, C is highly associative if it matches — and therefore retrieves — most of the other intrinsically similar stimuli. In numerical terms, the *associativity* of a coding C for a stimulus S is the ratio of the number of intrinsically similar objects that match that code to the total number of intrinsically similar objects.
- A coding C for a stimulus S is highly *discriminative* if it discriminates S from other intrinsically dissimilar stimuli. Alternately, C is highly discriminative if it fails to match — and therefore fails to retrieve — most of the other intrinsically dissimilar stimuli. In numerical terms, *discriminability* of a coding C for a stimulus S is the ratio of the number of intrinsically similar objects that match that code to the total number of objects that match that code.

Table 5.2: Definition of Associativity and Discriminability

The above definition of associativity and discriminability will suffice in following the rest of the discussion in this chapter. However, if you'd like a formal definition, refer to Table 5.2. In what follows, I will use the three phrases 'a description is both associative and discriminative' 'a description maximizes associativity and discriminability' and 'a description optimizes the tradeoff between associativity and discriminability' almost interchangeably.

To solidify these ideas, let's quickly consider two examples, one from Information Retrieval (IR), and another from Case-Based Reasoning (CBR). An IR system has access to a large set of documents. Given a query, the system should retrieve *all* the documents that are relevant to the query; this criterion is called 'recall.' At the same time, the system should retrieve *none* of the documents that are irrelevant to the query; this criterion is called 'precision.' A good description or indexing scheme for an IR system should maximize both recall and precision [Salton 88]. Such a description should associate a given query with all revelant documents, and discriminate it from all irrelevant documents., i.e., it should maximize both associativity and discriminability.

A CBR system has a large memory of cases which are prior problem solving episodes. When the system is given a problem to solve, it resorts to its prior experience in solving similar problems. This requires that the right precedents be retrieved: i.e., the system ought to retrieve all the cases that are similar to the current problem, and none of the cases that are dissimilar to the current problem. Typically, CBR systems use an indexing mechanism to help retrieve the right cases. Hence the indexing scheme may be seen as a description of the cases in memory. The best indexing scheme (description) is one that associates a given problem with all similar cases in memory, and discriminates it from all dissimilar cases in memory, i.e., such an indexing scheme should be both associative and discriminative. In Section 5.5, I will review three CBR systems to show that their design is strongly motivated by their need to categorize their world into categories that are both associative and discriminative.

Figure 5.3 provides a pictorial interpretation of the Classical View. This view is adequate in dealing with intensional concepts defined by isolated properties. Each property constitutes an entirely different discriminating grid, cleaving the world in arbitrary ways. Figure 5.4 provides a pictorial interpretation of categories suggested by the basic-level effect. These categories, to use Rosch's terms, "cleave the world at its seams." Table 5.3 contains a synopsis of the basic-level effect.

In summary, the basic level effect points out that at some level of abstraction, the categories of the world are very well differentiated, and that at this level of abstraction, the categories of the world are also the categories of the mind. Unlike intensionally

Besides being utterly confusing, these two terms are tied to a particular representation of categories in terms of feature vectors. My terms are more general in that they are based on the notion of 'description', which can be any arbitrary description of an object: features, feature clusters, or category names. Hopefully, these terms are also less confounding than predictiveness and predictability. In Section 5.3.2, these terms are extended to characterize relational structures as well.

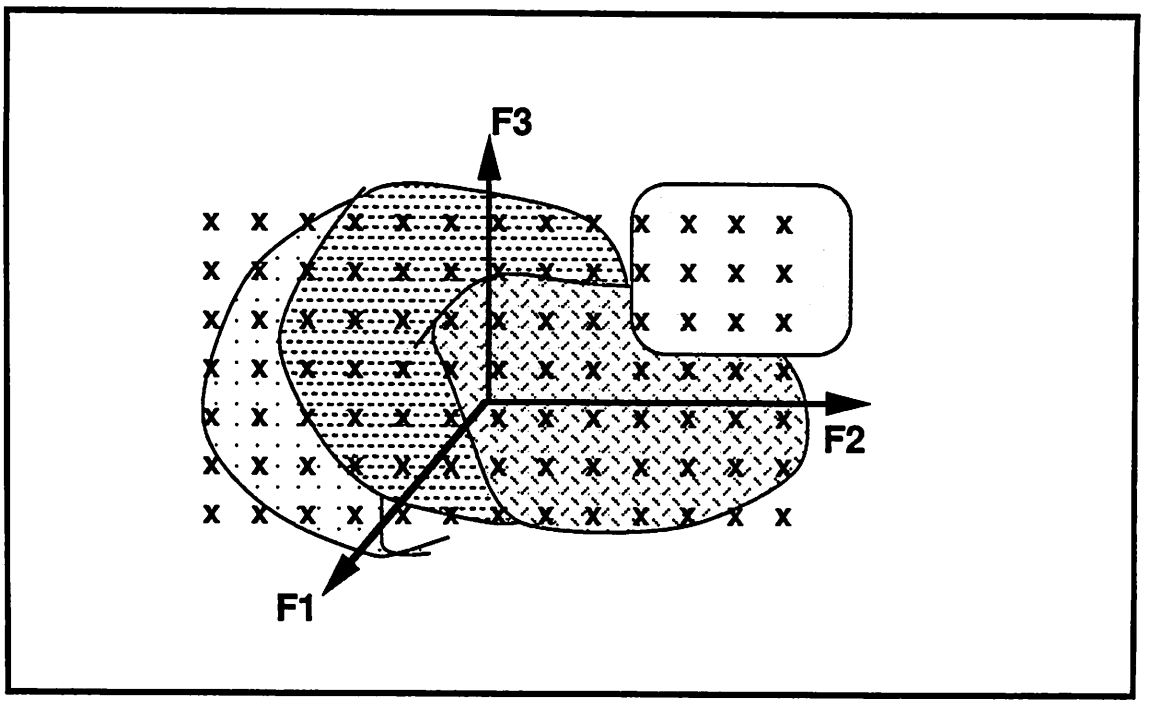


Figure 5.3: A Pictorial Interpretation of the Classical View

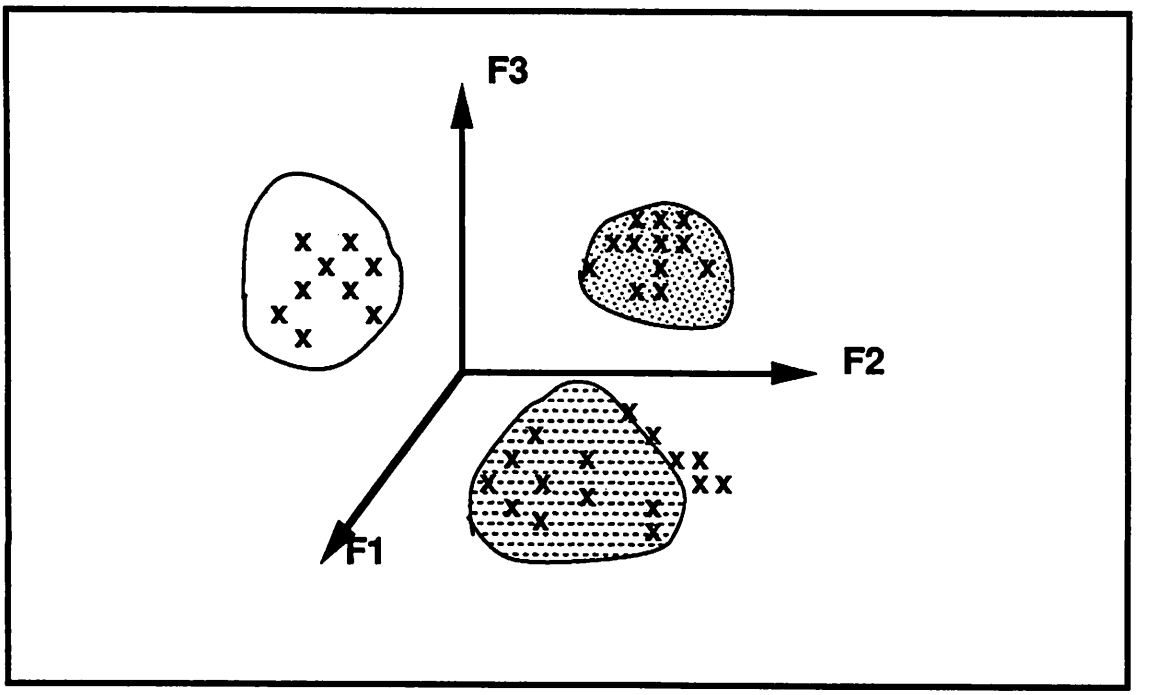


Figure 5.4: A Pictorial Interpretation of Basic-Level Categories

- The *Classical* view of categories defines categories through intensional properties such as 'the set of all things to take on a camping trip.'
- The *Basic-Level* effect points out that there is a level of abstraction at which the categories in the world are very well differentiated. These categories are characterized by information rich bundles of co-occurring features. Objects in the world are primarily affiliated to these basic-level categories.
- *Associativity* and *Discriminability* refer to any description, whether it be a feature, feature cluster, or a category. A highly associative description groups an object with other intrinsically similar objects; a highly discriminative description discriminates an object from intrinsically dissimilar objects.
- Basic-level categories maximize both associativity and discriminability: they *associate* an object with most other intrinsically similar objects and discriminate it from most other intrinsically dissimilar objects.

Table 5.3: Synopsis of Basic-level Categorization

defined categories such as 'things to take on a camping trip,' basic-level categories have a great deal of internal structure due to the co-occurrence of feature clusters. Hence the basic level categories group intrinsically similar objects together in the same category, and intrinsically dissimilar objects in different categories. I defined two terms, *associativity* and *discriminability* to characterize the basic-level categories: basic-level categories are both associative and discriminative in that they *associate* an object with most other intrinsically similar objects, and *discriminate* it from most other intrinsically dissimilar objects.

Are we still on the Ra*? Yes, the papyrus is creaking. There are stars outside; we are far away from the misty coast.*

—Thor Heyerdahl, The RA Expeditions.

5.3 Two Answers

In Section 5.1, we asked two questions about RA. The first question was concerned with RA's generalization phase, and the second question was concerned with RA's assimilation phase. The two subsections of this section answer these questions in terms of our discussion on basic-level categories. Since basic level categories are a 'psychological' phenomenon, I won't explicitly use this term in this section, but only the terms associativity and discriminability. But the implications should be obvious. In Section 5.4, I will discuss the conjecture that research schemas are basic level episodic structures.

5.3.1 Answer 1: Generalization

In Section 5.3.1, we asked the question, "What is the basis for RA's generalization?" In order to answer this question, let's now phrase it in terms of a concrete example. Shown below is the instantiated schema of [Rajamoney 88]:

ref: {(entails EBL incorrect-theory-problem)
(entails EBL incomplete-theory-problem)}

def: {(instantiates incorrect-theory-problem rajamoney-88-problem)
(instantiates incomplete-theory-problem rajamoney-88-problem)
(solves theory-revision rajamoney-88-problem)}

This schema is depicted in Figure 5.5. Let's assume that RA's assimilation phase has just inferred this schema based on its **def** (see Section 4.6.4 for details on how this is done). Now the generalization phase replaces the constants in the schema by *typed* variables to obtain the following skeletal schema:

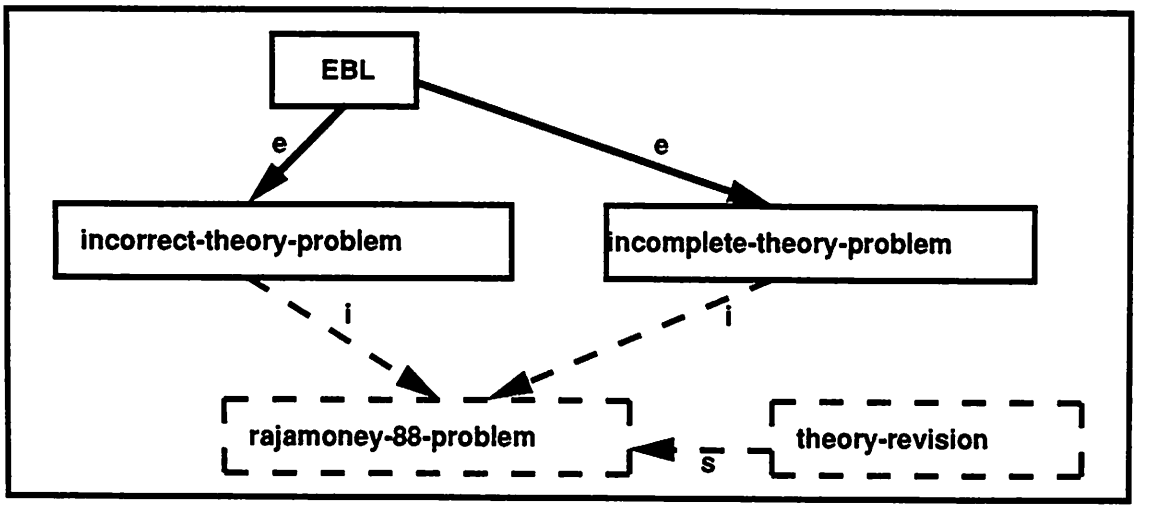


Figure 5.5: The Instantiated Schema of [Rajamoney 88]

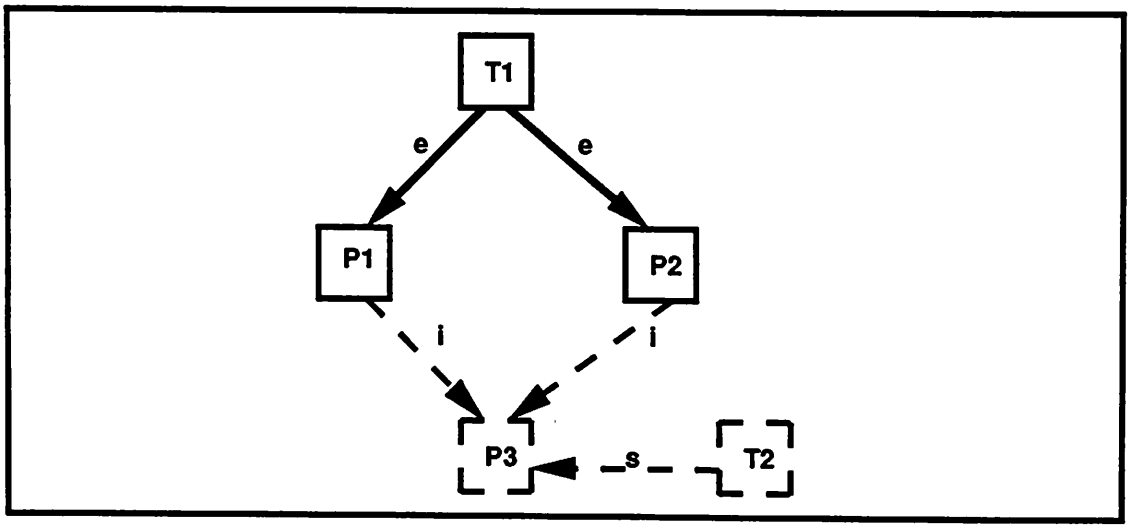


Figure 5.6: The Skeletal Schema of [Rajamoney 88]

ref: {(entails T1 P1)
(entails T1 P2)}

def: {(instantiates P1 P3)
(instantiates P2 P3)
(solves T2 P3)}

This skeletal schema is shown in Figure 5.6. RA has seen a paper that used some research strategy in the context of some specific objects like EBL, incomplete-theory-problem, incorrect-theory-problem and so on; from this, RA generalizes this strategy to any objects of the types problem and technique, i.e., it constrains the variables P1, P2 and P3 to be of type problem and the variables T1 and T2 to be of type technique. What is the basis for this conceptual leap?

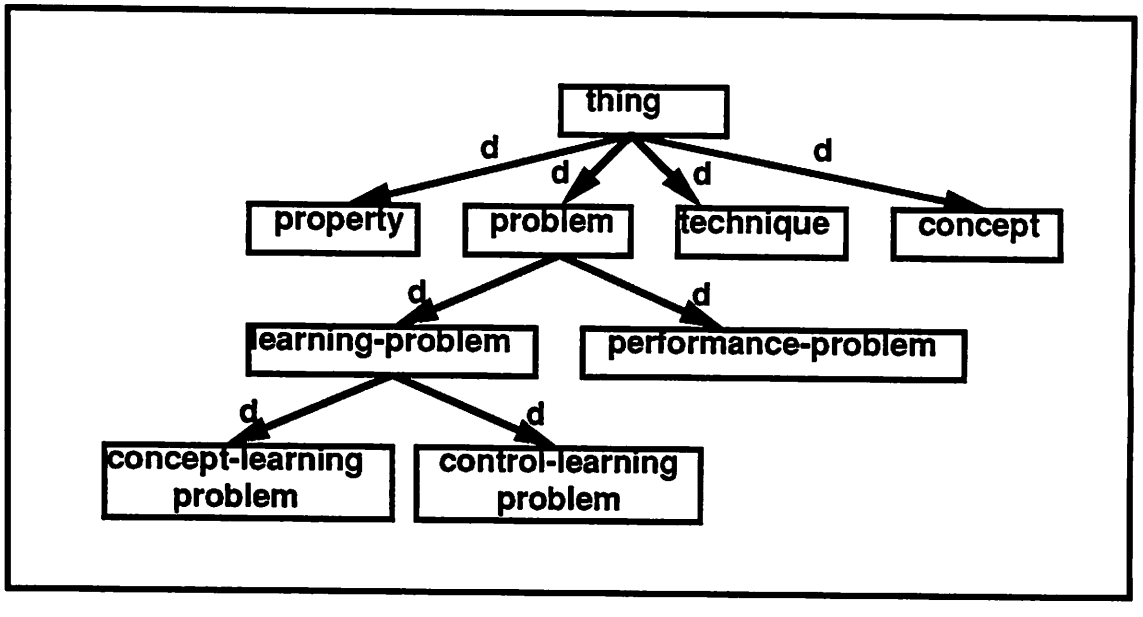


Figure 5.7: RA's Abstraction Space

First we note that there are other possible generalizations that RA could have made. Figure 5.7 shows some relevant parts of RA's abstraction hierarchy. The top-level is the root of this structure, and is marked thing[16]. At the next level are nodes like problem, technique, etc. Let's focus on the tree rooted at the node problem. problem consists of entities such as performance-problem and learning-problem; learning-problem consists of concept-learning-problem and control-learning-problem. Looking back at the schema of [Rajamoney 88], why did we generalize incomplete-theory-problem to be of the type

[16]This is a standard practice in building hierarchical structures. The root is usually a vacuous entity like thing.

problem? Why not concept-learning-problem? Or learning-problem? Or thing? What is the reason for choosing the type problem over the other types in this hierarchy?

Of all possible categories in RA's abstraction space, the categories problem, technique, concept and property are the most differentiated categories, i.e., they are both associative and discriminative. The category above these, thing, has no discriminability; the categories below these have lower associativity. RA generalizes an instantiation to a level that is both associative and discriminative, so that an instantiation is placed into a category of objects that are intrinsically similar to that instantiation, and intrinsically dissimilar instantiations are placed into different categories. For example, EBL is intrinsically similar to techniques and is generalized into a variable of type *technique*; EBL and incorrect-theory-problem are intrinsically dissimilar objects and are generalized into variables of different types. Thus RA's generalization is justified because it generalizes a fact about a specific object into a general fact about all intrinsically similar objects, but not into a general fact about any intrinsically dissimilar objects.

To understand this argument, let's ask what RA knows about the various categories (types) in its world. The semantics of the various types of nodes are determined by what sorts of relations in which nodes of that type can partake. For example, all that RA knows about the type technique is that it is something that can solve problems, can entail problems, can exhibit properties, and so on — precisely the information expressed in the type constraints for RA's relations (see Table 2.5). Hence, these type constraints can be seen as the 'features' of these types or categories. For example, the features of the type problem are: {dominates, instantiates, encapsulates, solved-by, entailed-by, exhibits, defines, R}; similarly, the features of the type technique are: {dominates, instantiates, encapsulates, solves, entails, exhibits, defines, R}.

The three epistemological relations dominate, instantiate, and encapsulate are applicable to all types, and hence have no discriminability, i.e., they do not discriminate among the various types of objects. Thus these three relations, while used to organize RA's knowledge base, are essentially vacuous properties that do not contribute to the content or 'essence' of the various types[17]. Similarly, the relation R also applies to any object of any type, and hence does not contribute toward the essence of the types. This leaves us with the other four relations: solves, entails, exhibits, and defines. When treated as features, these four relations circumscribe the following four categories:

problem: {solved-by, entailed-by, defines, exhibits}
technique: {solves, entails, defines, exhibits}
concept: {defined-by}
property: {exhibited-by}

[17]These relations are similar to features such as 'animate.' In a world of only animate objects, this is a vacuous property. However if the world also contained inanimate objects, then this feature has discriminability. In RA's world, the three epistemological relations are applicable to all objects, and hence do not provide any information about the specific types.

Thus, for example, a problem is something that can appear as a primary in any one of the following relations — solved-by, entailed-by, involves, and exhibits; and a technique is something that can appear as a primary in one of the following relations — solves, entails, defines, and exhibits. The type concept always co-occurs with the feature defined-by, and the type property always co-occurs with the feature exhibited-by. Similarly, the type problem always co-occurs with the feature cluster {solved-by, entailed-by, involves, exhibits} and the type technique always co-occurs with the feature cluster {solves, entails, involves, exhibits}[18].

These four types are the most differentiated categories in RA's world, since they are both associative and discriminative. The level of abstraction above these types, thing, does not have enough discriminability: it does not discriminate among types (the four types above) that have different clusters of features. A subordinate level of abstraction, such as learning-problem, has no more discriminability because there are no extra features (i.e., relations) that are applicable to learning-problem, but not to problem. However, the type learning-problem is less associative: it associates an object with all other learning-problems, but not performance problems, whereas from the point of view of type-constraints and heuristics, there is no distinction between learning-problems and performance-problems. Hence the four types above are the most differentiated categories that reflect the inherent category structure of RA's world. RA's generalization is justified in that it generalizes a specific object (an instantiation) into a variable of the category that includes all of the intrinsically similar objects.

To understand this argument, let's turn to a concrete example — the paper [Rajamoney 88] we considered above. Let's look at a couple of possible alternative generalizations for this schema. The following is a generalization of [Rajamoney 88] to the level of abstraction thing:

ref: {(entails a b)
(entails a c)}

def: {(instantiates b d)
(instantiates c d)
(solves e d)}

In this generalization, all instantiations have been generalized into the type thing which is same as saying that the variables a, b, c, d and e are basically typeless. If we look closely at the schema, there is only one consistent assignment of types to these

[18]The correlation between the feature cluster {solves, entails} and the type technique, and that between {solved-by, entailed-by} and the type problem are two-way correlations: for example, solves always co-occurs with problem, and vice-versa. The correlation between the features involves and exhibits and the type problem, on the other hand, is not a two-way correlation: while exhibits co-occurs with problem, problem does not always co-occur with exhibits. However, involves and exhibits cannot be moved up to the level of thing, because these features do not apply to concept and property.

variables. Referring to the type constraints of the various relations (Table 2.5, Chapter 2), the entails relation constrains a to be of type technique and variables b and c to be of type problem. If b and c are of type problem, this forces d to be of type problem as well since the instantiates relation is constrained to link objects of the same types. The solves relation now constrains e to be of type technique. Hence, this generalization is equivalent to the 'correct' generalization (page 152). It is no more general or associative than the 'correct' generalization, even though it generalizes the instantiations to a higher level of abstraction. However, this generalization, as it stands, is less discriminative: it does not discriminate among the various types of objects, whose type constraints are nevertheless present in the schema. Besides being annoying, these 'overgeneralizations' will result in higher matching costs when this schema is used as the following heuristic rule:

> If there are nodes a, b, and c such that a entails both b and c, then suggest: "You could find a d that is an instantiation of both b and c. Then you can propose a e to solve d.

When RA attempts to use this heuristic, it will first check the applicability conditions (the **ref** or the 'if' part) to see if the variables in the **ref** part are instantiable in the neighborhood of the anchor node (see Chapter 3). While 'a' can only match against a node of type technique and b and c against a node of type problem, this information is not present in the schema. Hence RA will have to try several futile matches, resulting in unnecessary matching costs. Thus, generalizing a research schema to the level thing is not advantageous because such a generalization does not reflect the type constraints that are inherently present in the schema.

Next, let's consider a generalization of [Rajamoney 88] to a subordinate level of abstraction as follows:

ref: {(entails LT1 LP1)
(entails LT1 LP2)}

def: {(instantiates LP1 LP3)
(instantiates LP2 LP3)
(solves LT2 LP3)}

In this generalization, assume that the prefixes LT and LP stand for variables constrained to be of type learning-technique and learning-problem. After having seen a paper that involved specific objects such as EBL and incorrect-theory-problem, we have generalized it to apply to any objects of the types learning-technique and learning-problem. However, there is no information in the instantiated schema that is specific to learning-problems per se. All the relations in the schema are applicable to all kinds of problems, including performance problems. Hence this generalization contains no more special information about learning-problems than the 'correct' generalization, but is less widely

applicable since it cannot be applied to performance problems. In other words, this generalization is no more discriminative than the 'correct' generalization, but is less associative. In contrast, as we saw before, the generalization to the level thing (the previous case) is no more associative than the 'correct' generalization, but less discriminative. Thus the 'correct' generalization is both associative and discriminative.

In Summary, RA's generalization strategy is initially surprising since it appears as if RA has no basis for generalization. A closer look reveals that RA's world contains a level of abstraction at which the categories (types) of its world are well-differentiated, i.e., these categories are both associative and discriminative and have the most features (semantics) unique to their level. RA generalizes instantiations (constants) into variables constrained to these categories. RA's generalization is justified because it generalizes a fact about a specific object into a general fact about all intrinsically similar objects, but not into a general fact about any intrinsically dissimilar objects. Table 5.4 contains a synopsis of this section.

- In SBL-style learning, multiple examples plus some form of encoded bias provide a basis for generalization. In EBL-style learning, the domain rules provide a basis for generalization.
- RA's generalization phase uses neither multiple examples, nor a causal theory of the domain. So, what is the basis of RA's generalization?
- RA's types, namely, problem, technique, concept, and property are well differentiated categories in that each of these is both associative and discriminative. The type at a higher level of abstraction, thing, is no more associative, but less discriminative. The types at a lower level of abstraction, for example, learning-problem, is no more discriminative, but less associative.
- RA generalizes an instance by placing it into a category that contains all the intrinsically similar objects, and none of the instrinsically dissimilar objects.
- The category structure of RA's world, therefore, provides a basis for RA's generalization strategy.

Table 5.4: Generalization Phase: Question and Answer

5.3.1.1 The Story of Acq

The category structure of RA's world gets disturbed by the relation acq. To recap, this relation asserts a relationship between a 'performance technique' and a learning problem that emerges from it. For example, DeJong's Frump system uses a schema-based technique called sketchy scripts to solve the story-understanding problem [DeJong 79]. In Frump, the sketchy scripts are hand-coded into the system. In the Genesis system, Dejong and Mooney consider how one could automatically learn or acquire the schemas of the sort used by Frump [Mooney & DeJong 85]. The relationship between Frump's schema-based-technique and Genesis's schema-acquisition-problem is denoted by the relation acq. Thus, acq states the relationship between a (non-learning) performance technique and a learning problem that emerges from it. See Chapter 2 for an extended description of this relation.

Since acq introduces further discriminations into the category structure of RA — it distinguishes techniques in general from performance-techniques and problems in general from learning-problems — our simple generalization strategy will not correctly generalize schemas involving acq. To see this, let's consider the generalization of [Mooney & DeJong 85]. One possible canonical abstract for this paper is shown below:

> Frump's schema-based technique is a performance technique that solves the story-understanding problem. However, it is a performance technique that does not learn its schemas. In this paper, we propose a concept-learning problem, the problem of acquiring schemas automatically. Let's call it the schema-acquisition-problem. Explanatory schema acquisition is a technique that solves the schema-acquisition-problem.

The abstract above corresponds to the following schema:

ref: {(dominates performance-problem language-understanding-problem)
(solves schema-based-technique language-understanding-problem)
(dominates learning-problem concept-learning-problem)}

def: {(acq schema-based-technique schema-acquisition-problem)
(instantiates concept-learning-problem schema-acquisition-problem)
(solves explanatory-schema-acquisition schema-acquisition-problem)}

Let's assume that RA's assimilation phase has just acquired this instantiated schema from the **def** of the paper. The following is the 'correct' generalization that RA should obtain:

ref: {(dominates PP P2)
(solves PT1 P2)
(dominates LP3 LP)}

def: {(acq PT1 LP5)
(instantiates LP3 LP5)
(solves LT2 LP5)}

which corresponds to the following heuristic:

> If there is a performance problem P2 (i.e., a problem P2 that is dominated by performance-problem) that is solved by a technique PT1 (a performance technique), and if there is a kind of learning-problem LP3 (i.e., LP3 is dominated by learning-problem), then "You could find a learning problem LP5 that is an instantiation of LP3 and is acq-related to PT1, and propose a learning technique LT2 to solve LP5."[19].

However, RA will generalize this into the following (incorrect) skeletal schema:

ref: {(dominates P1 P2)
(solves T1 P2)
(dominates P3 P4)}

def: {(acq T1 P5)
(instantiates P4 P5)
(solves T2 P5)}

This generalization corresponds to the following heuristic:

> If there are problems P1, P2, P3, and P4 such that P1 dominates P2, and P3 dominates P4, and there is technique T1 that solves P2, then suggest: "You could find a new problem P5 that is both acq-related to T1, and is an instantiation of P4. Then you could propose a technique T2 to solve P5."

The point that is missed by this generalization is that P1 and P3 are required to be specific nodes in memory, namely performance problem and learning-problem, because the type constraints on acq requires that it link a performance technique to a learning problem. With respect to acq, schema-based-technique is not just any old technique, but a learning technique; schema-acquisition-problem is not any old problem but a learning problem. Hence replacing a performance technique like schema-based-technique by a variable constrained to be *any* technique categorizes this object with other objects (such as learning techniques) that are dissimilar to it with respect to the acq relation. Thus, coding schema-based-technique as technique (rather than as performance technique) is highly associative, but not discriminative enough, resulting in over-generalization.

[19]In simpler terms, this heuristic stands for the following: if there is a (non-learning) performance technique, then suggest "can you come up with either a concept-learning-problem or a control-learning-problem that emerges from this performance technique? Then you can propose a technique to solve this learning problem." In even simpler terms, "Can you find anything that is hand-coded into this system? If so, you could consider how to learn this automatically."

An easy solution I adopted was to disallow acq in the learning system RA II. In the case of RA, the four categories (the four types) are very fragile in that they are characterized by a very small number of features. Hence, a subordinate category discrimination (such as learning-problem) relevant to just a single relation, acq, becomes important. Suppose there were 45 relations that circumscribed the four types. Then, minor subordinate categories created by a small number of features (e.g., acq) are not very serious. The simple generalization strategy of replacing the constants with variables from the categories that are most differentiated will be correct most of the time, and will result in over-generalization in only a small number of cases.

5.3.2 Answer 2: Assimilation

In Section 5.3.2, I raised the question, "Why are configurations A and B different research strategies despite containing a common configuration '(solves T P)', whereas C and D not different research strategies despite containing the configuration A?" In this section, I will argue that the structure of the **ref** ensures that a research schema, as a whole, is both associative and discriminative. For example, '(solves T P)' has low discriminability since it does not discriminate between the research heuristics A and B; configurations C and D have low associativity since they make an unnecessary distinction between two configurations that involve the same research strategy. Configurations A and B are both associative and discriminative and are therefore the best descriptions of their underlying research strategies. This argument is somewhat complex, so we will take it in small doses.

5.3.2.1 Instantiated Schemas as Descriptions of Research Papers

For the moment, let's forget all about research strategies and heuristic rules and talk about an *instantiated* research schema as a description of a particular research paper. When given a disjointed set of new relations belonging to the **def** of a paper, the assimilation phase finds a set of known relations that connect the various objects in the **def**.

Let me point out that there is nothing new here: this is an age-old technique for natural language inference traceable to Quillian's semantic nets [Quillian 67]; it was also used by Rieger's Memory [Schank & Rieger 74] and Norvig's Faustus [Norvig 87]. The claim behind this inference technique can be stated as follows: When given two pieces of information as part of the same discourse, one needs to find a connection between the two in order to 'understand' the discourse. With all the three systems just mentioned, there is the problem of deciding when you have inferred enough. To handle this problem, each system used several, essentially arbitrary, conventions. For example, Faustus allows only proper inferences, where proper is defined as plausible, non-explicit, relevant and

easy. Similarly, RA assumes that the **ref** should contain *just enough* relations to connect all the pre-objects in the **def**. Is this just an arbitrary convention or something more?

With respect to the use of an instantiated schema as a description of a particular research paper, this assumption is not strictly necessary. Hence, this is a convention that may be graced with the adjective arbitrary. For example, a research paper can include arbitrary amounts of background material, and hence a research schema, as a description of the paper, can contain arbitrary relations in its **ref**. For instance, [Utgoff 84] might be described in terms of two different schemas that correspond to the following abstracts:

> [1]: Winston's technique solves the concept-learning problem [Winston 71]. Vere showed that Winston's technique entailed a backtracking problem called backtracking1, and proposed Veres-technique to solve the concept-learning problem while avoiding backtracking1 [Vere 75]. Then Mitchell showed that Vere's technique entailed another kind of backtracking problem called backtracking2, and proposed Version spaces to solve the concept-learning problem, while avoiding backtracking2 [Mitchell 78]. In this paper, I show that Version-spaces entails a problem called the fixed-bias problem; I propose a technique called bias adjustment to solve this problem.

> [2]: The Version-spaces technique solves the concept-learning problem [Mitchell 78]. In this paper, I show that Version-spaces entails a problem called the fixed-bias problem; I propose a technique called bias adjustment to solve this problem.

These two abstracts are depicted in Figure 5.8. As a description of [Utgoff 84], both the abstracts above are equivalent, with the first one including more background than the second. In fact, archival journal papers typically include much more background material in describing the same research than say, a conference paper on a specialized topic. Hence, with respect to instantiated research schemas and their use as a representation for papers, our assumption about what should go into a **ref** is no more than an arbitrary convention. However, with respect to the use of skeletal research schemas as a *description of the research strategy in a paper*, our assumption about the structure of the **ref** is more than a convention: it guarantees that, as a description of a research strategy, a skeletal research schema is both associative and discriminative. To present this argument, we will start by distinguishing between a research schema and the underlying research strategy — i.e., between a heuristic and its heuristic content.

5.3.2.2 Heuristics vs Heuristic Content

Let's consider the following (weak) heuristic rule which we will refer to by the label W: "If P is a problem, then one could propose a technique T to solve it." This heuristic

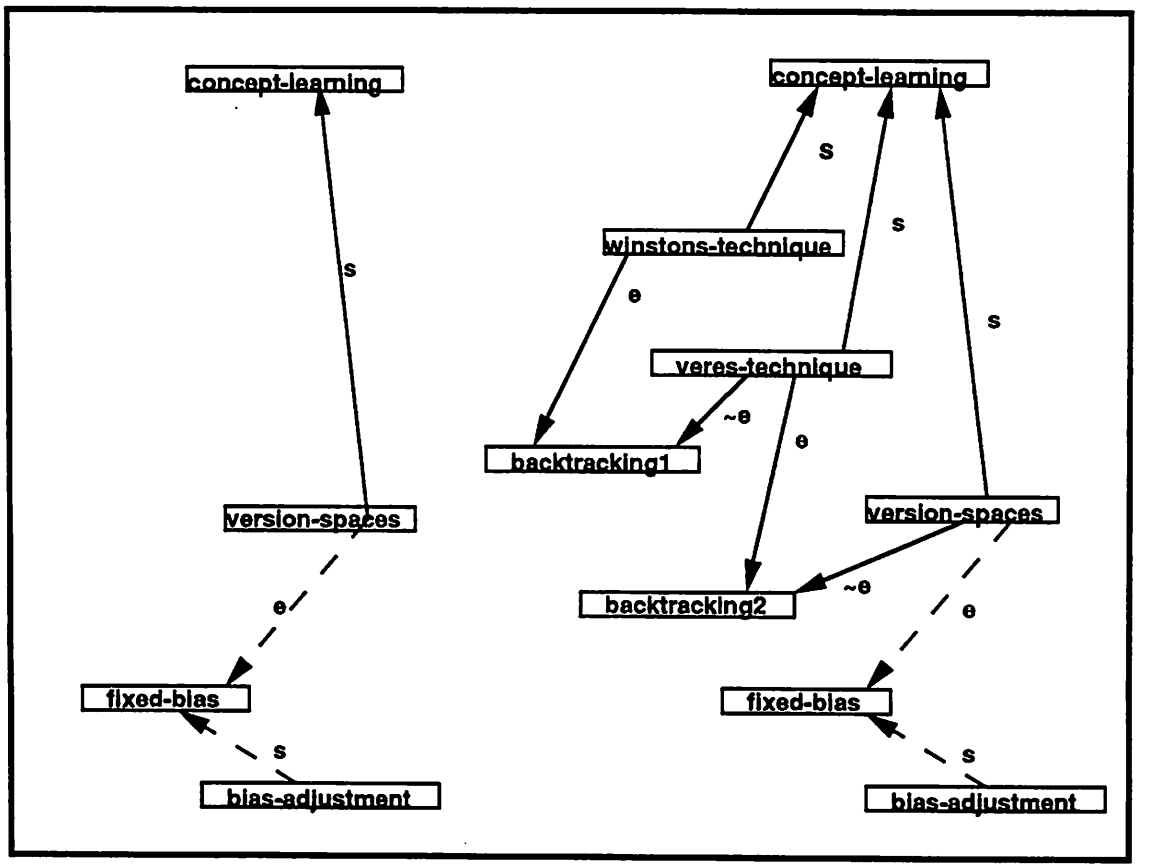


Figure 5.8: Two Abstracts of [Utgoff 84]

essentially restates the type-constraints of the relation solves; since these type constraints may also be seen as the features of the types, it is also a restatement of what it means for something to be problem (it can be solved) or a technique (it can solve problems). Thus W is a statement about an overt property (feature) of problems and techniques.

In addition to such overt properties, these types also have hidden or covert properties. To see that, let's consider the following two heuristics:

> [X] If there is some technique T1 to solve some problem P1, then one can see if T1 has any deficiency or emergent problem P2. Then one could propose a new technique T2 to solve P1 while avoiding P2.

> [Y] If P1 and P2 are two sibling problems (i.e., they are both dominated by the same parent problem P), and one of these, P1, has a technique T1 to solve it, then one could see if T1 can be adapted to solve P2 as well.

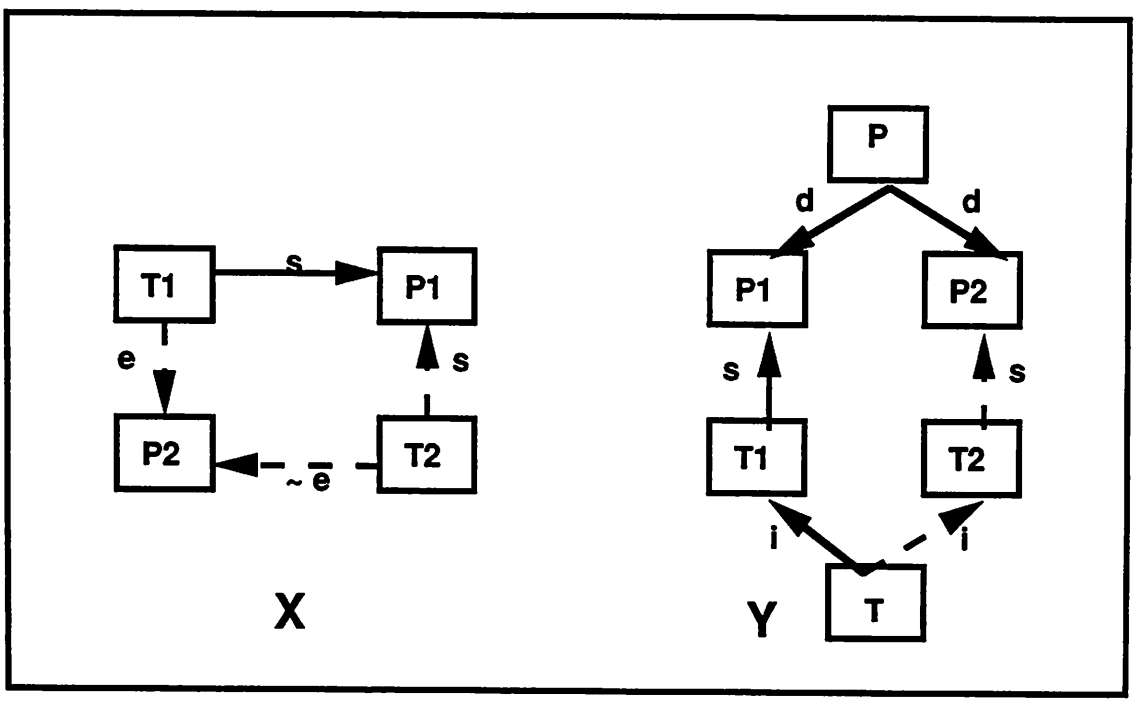


Figure 5.9: Research Schemas X and Y

These two heuristics, labelled X and Y, are shown in Figure 5.9[20]. Why are these well-motivated research heuristics? While they 'feel' right, it is difficult to state why they are good heuristics. If hard-pressed, one could come up with some "intuitive" reason as

[20] X is same as the configuration A we saw before.

follows: "Well, people *really* like to solve problems, and a deficiency is something people like to rectify. Hence providing a new technique that rectifies a blemish is of interest," or "Two sibling problems are likely to have some nitty-gritty, some core aspect that is similar. When a technique solves one of these problems, we expect that this technique somehow addresses and solves this core. Hence it might be possible to use a technique of the same kind to address the core of the other (sibling) problem." It is likely that you disagree with my reasons and you may be able to formulate better reasons why these are motivated research strategies; the point here is that, whatever your reasons be, they are not derivable from the overt properties of the types and the relations[21]. Such reasons refer to what might be called covert properties of these objects that are revealed when a number of objects interact with each other. For example, the first heuristic connects problems and techniques with people, their goals and intentions[22]. The second heuristic reveals that a problem might have a core, and a technique may address this core. These are covert or *interactional* properties that are hard to even verbalize and emerge only when we consider the interaction of several objects with each other.

I will use the term *heuristic content* or *research strategy* to refer to the reasons underlying a heuristic. These reasons may refer to the somewhat elusive covert and interactional properties of various objects. A *heuristic rule* (alternately, a *heuristic*) is simply a *description* of this elusive heuristic content in some manageable way.

To understand this better, let me give an example from a common-sense domain. Let's imagine a five year old boy, John, playing with a set of black and white checker pieces. Somehow, a few fridge magnets have also gotten mixed up in his collection of checker pieces. Let's assume that the fridge magnets look similar to the checker pieces, but are colored yellow. During play, John accidentally discovers that, unlike the black and white pieces, the yellow pieces have the bizarre property of getting stuck to various things such as chairs, fridges, kitchen cabinets, paper clips and so on. Fascinated, John tries out all his yellow pieces against various objects in the house and finds that they stick to a lot of things. He also tries out the black and white pieces and finds that they don't stick to anything. From this, John might learn a heuristic that yellow pieces, and only yellow pieces, stick to things. What John has learned is a *heuristic rule*, "yellow pieces stick to things," which is really a description of a deeper *heuristic content*, "magnetic

[21]Such 'intuitive' reasons characterize several of AM's heuristics as well [Lenat 76]. For example, one of AM's heuristic is "A non-constructive existence conjecture is interesting." Lenat writes: "Thus the unique factorization theorem is judged to be interesting because it merely guarantees that some factoring will be into primes. If you gave an algorithm for that factoring, then the theorem actually loses its mystique and (according to this rule) some of its value" (p. 241). Also, consider the classic heuristic that the air distance between two cities is a good approximation of their road distance. Ponder the 'content', i.e., the assumptions about human societies, roads, and earth's terrain that underlie this heuristic.

[22]For example, why is the following not a terribly well-motivated research strategy: "Technique T1 solves problem P1. In this paper, I show that T1 has a deficiency P2. I propose a new technique T2 to solve P1. My technique also has the deficiency P2."

objects stick to other ferro-magnetic objects." In John's world, his *heuristic rule* is a good description of its *heuristic content* because this description is associative since it sticks (categorizes) all sticking things together; it is also discriminative since it does not stick non-sticking things with sticking things.

5.3.2.3 The Structure of Research Schemas

Having distinguished between a heuristic and its content, we can now talk about research strategies and research schemas. A skeletal research schema is simply a description that stands for an underlying heuristic content or research strategy. Hence, we should ask, "How do we know that research schemas are *good* descriptions?" In this section, I will show that the assumptions behind the structure of the **ref** ensures that a research schema is both associative and discriminative — i.e., it associates a paper with other papers that have inherently similar research strategies and discriminates a paper from other papers that have inherently dissimilar research strategies.

As we saw before, initially, even before any papers are input, RA has a set of weak research heuristics that are simply the type constraints of RA's relations. For example, the type constraint that a solves relations should link a problem to a technique can be seen as the following research schema:

ref: {P}

def: {(solves T P)}

This schema corresponds to the weak research strategy W above: "If there is a problem P, one could propose a technique T to solve it."[23].

Since schemas like W are directly derived from the type constraints of RA's relations, they refer only to the overt properties of the types. However, the covert interactional properties of the types can induce more discriminated research strategies; for example, although both X and Y (in Figure 5.9) do solve problems with techniques, they refer to different interactional properties of problems and techniques. Hence W is a bad description of the heuristic content underlying X and Y: if papers that use the research strategy X as well as those that use the strategy Y are described or indexed in terms of the research schema W, such a description is associative (it does categorize all papers with similar strategy, say X, in the same category) but not discriminative (it also categorizes papers with dissimilar strategies, say X and Y, together). Hence, schemas such as W are inadequate as descriptions of research strategies, and RA has to learn more specific or more discriminative research schemas to describe the research strategies expressible through interactional properties of the various types.

[23]We will call this a research schema for the sake of argument. All of RA's schemas include only relations in their **ref** and **def**.

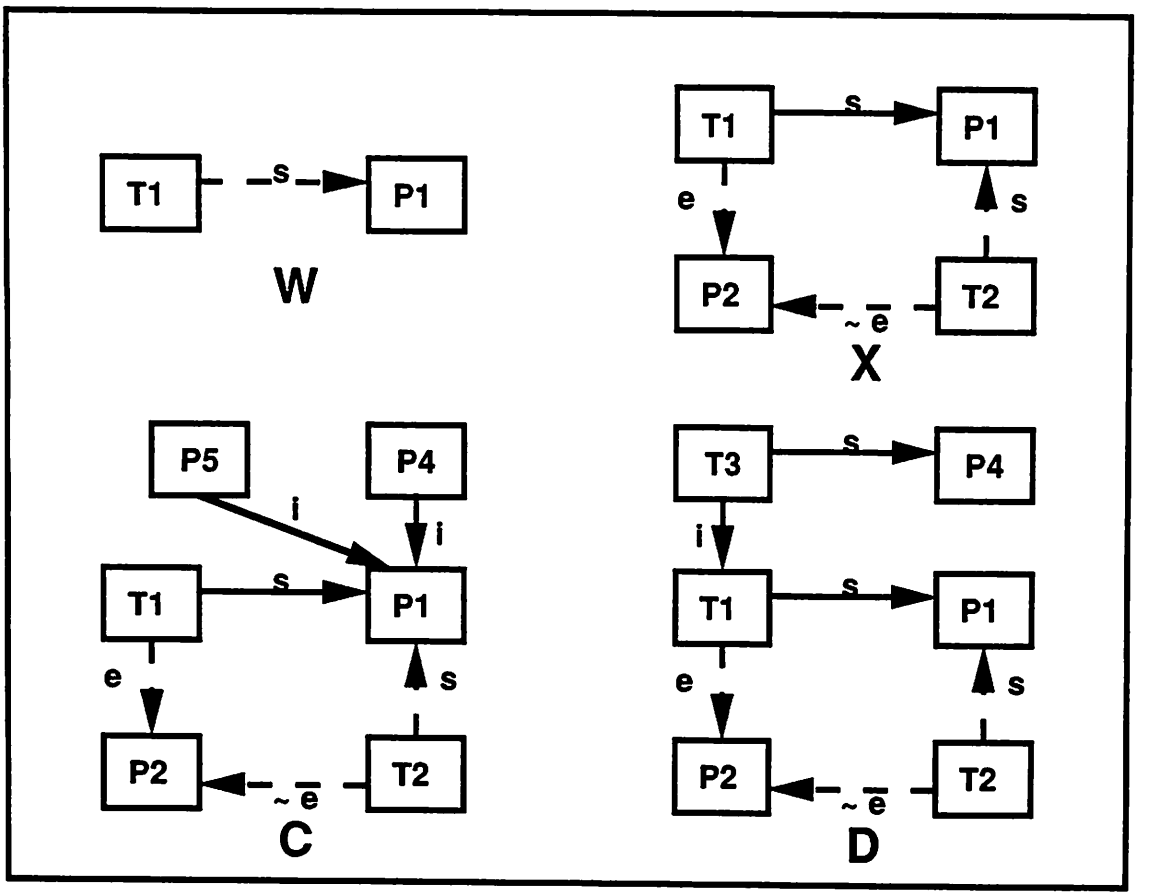


Figure 5.10: Configurations W, X, C, and D

The natural next question is, "When do we stop making infinite discriminations?" Is each paper an entirely unique research strategy? For example, do configurations C and D (repeated in Figure 5.10 for convenience) stand for different underlying research strategies? Should C and D be discriminated from each other and from X?

Let's note that C, D, and X all have the same **def**. The **ref** in X completely brackets the interactions of the objects in the **def**. Hence all possible interactional properties among these objects are captured by the research schema X. Thus the configurations C and D do not introduce any new interactional properties, and hence any new essential discriminations: they are no more discriminative than the schema X in describing the underlying research strategy. However, they are both less associative than X. If two papers that have the same research strategy X are described in terms of configurations C and D (respectively), then such a description fails to categorize papers with inherently similar research strategy X in the same category, resulting in low associativity.

Schemas such as X whose **ref** completely brackets all the objects in the **def** capture not only the overt properties of the objects in the **def**, but also their covert or interactional properties. Configurations such as W capture only the overt properties and hence do not discriminate between research strategies that have different interactional properties. Configurations such as C do not introduce new interactional properties and therefore no new discriminations. Thus the assumptions behind the assimilation phase — include all those relations that connect the pre-objects in the **def**, and only those — ensure that a research schema is discriminative since it includes all possible interactional properties; they also ensure that the schema is associative since it does not include any unnecessary information that would make it less associative.

In summary, the assumptions about what should go into the **ref** is just an arbitrary convention when we focus on instantiated schemas as a description of research papers: a paper can include an arbitrary amount of background material, and hence the **ref** could include an arbitrary number of relations from the pre-ontology of the paper. However, from the point of view of capturing the research strategy of the paper, these assumptions are not arbitrary: they guarantee that the research schema, as a description of the underlying research strategy, is both associative and discriminative. By requiring that the **ref** include all relations that connect the pre-objects in the **def**, the assimilation phase ensures that a research schema captures all the interactional properties among the various objects in the **ref** (providing discriminability). By requiring that the **ref** include only these relations, the assimilation phase ensures that a research schema does not include unnecessary conditions (providing associativity). Table 5.5 contains a synopsis of this section.

- Since RA has a set of heuristics (derived from the type constraints) even before learning, why does RA need to learn any new research schemas at all?
- If the original set of heuristics are not enough, the when do we stop? How do we know that each paper is not an entirely unique research strategy?
- The assimilation phase assumes that the **ref** should contain a set of relations that connect all the pre-objects in the **def**, but no more. This completely brackets the interactions among all the objects in the **def**.
- The assumptions about what should go into the **ref** of a schema is an arbitrary convention with respect to an *instantiated* schema as a description of a particular paper. However, these assumptions are crucial with respect to a *skeletal* schema as a description of the underlying research strategy.
- The initial set of weak heuristics (derived from the type constraints) refer only to the overt properties of the types. As such, they are low in discriminability since they do not discriminate between different research strategies that involve different covert interactional properties. Hence RA needs to learn more specific research schemas.
- However, including all and only those relations that completely bracket the interactions among all the objects in the **def**, RA ensures that it does not treat each paper as an entirely unique research strategy.
- Thus the assumptions behind the assimilation phase and the generalization phase together ensure that a research schema, as a description of the underlying research strategy, is both associative and discriminative.

Table 5.5: Assimilation phase: Question and Answer

5.4 Discussion

What now? How to slip away from this increasingly involved situation, away from the mob, back into the peaceful night from which I had come?

—Thor Heyerdahl, The RA Expeditions.

In the last three sections, I took you through a fairly complicated route to see a lot of forests. Are these forests part of a larger eco-system? In this section, I will attempt to draw a bigger picture based on our analysis of the last three sections. Most of this is speculative, and I raise more questions than I answer. But first, let's quickly review what we saw in the last three sections.

- In Section 5.1, I raised two questions about research schemas and phrased them in terms of RA's learning strategy. The first question was concerned with RA's generalization phase: what is the basis for RA's generalization of instantiations into types? The second question was concerned with RA's assimilation phase: how do we know that each paper is not an entirely unique research strategy? When do we stop making infinite discriminations?

- In Section 5.2, we stepped back from RA and reviewed theories of categorization. Here we saw that the so-called 'basic-level' of abstraction has several interesting properties. Psychologically, the basic-level categories are the primary categories of affiliation for objects because categories at the basic level of abstraction exhibit the most number of cooccurring features unique to that level (see Figure 5.2). Hence, these categories reflect the inherent category structure of the world.

- I defined two terms *associativity* and *discriminability* to characterize the basic-level. The basic-level categories are both asociative and discriminative.

- In Section 3, we returned to the two questions raised in Section 1. In Section 3.1, I showed that the four types in RA were both associative and discriminative. Hence, RA's generalization is justified in that it generalizes an instantiation by placing it into a category that contains all instrinsically similar objects and none of the instrinsically dissimilar objects.

- In Section 3.2, I showed that the structure of a research schema can also be explained in terms of associativity and discriminability. The assumptions about the structure of the **ref** was an arbitrary convention with respect to the instantiated schema as a description of the content of a paper, but was crucial with respect to the use of the skeletal schema as a description of the research strategy of the paper. I showed that the assumption that a research schema should include a set of relations that bracket

all the objects in the **def** ensures that the schema, as a whole, is both associative and discriminative.

It is only fair to ask, "What does this all mean?" In answering this question, I can only speculate and raise several new questions. The rest of this section is meant to be read in that same spirit.

5.4.1 RA's Types: Basic Level Categories?

Throughout Section 5.3, I was careful not to mention the term basic-level with respect to research schemas. The analysis focused entirely on the terms 'associativity' and 'discriminability' to show that RA's types are all (both) associative and discriminative. While basic-level categories have the property of being both associative and discriminative with respect to the *real world,* RA's types have the same property with respect to *RA's world.* We can speculate about how RA's world maps onto the real world: are problems and techniques basic-level categories in the real world as well? Let's consider arguments both for and against.

In addition to the overt features RA's types (which are simply the relations in which the types can partake), we saw that research schemas make sense to us only because we attribute lot more covert or interactional properties to the types as well (see Section 5.3.2). Hence, more than RA's toy world is in operation here: if RA's types are basic-level categories only in RA's world, then there is no reason why RA's generalizations should make a great deal of sense to us, the humans, who attribute covert features to these objects in the real world. This suggests that RA's types are well-differentiated categories also with respect to their covert or interactional features — i.e., features not explicitly represented in RA's world, but attributed by us when we make sense of a schema — and are hence basic-level categories in the real world as well.

However, there also exists a counter-argument to the above [24]. One of the properties of basic level categories is that they are the most stable and perceivable categories: if two people were asked to describe a scene at all possible levels of abstraction, the basic level description is likely to be the most consistent. This is what Brent Berlin found in his study of Tzeltal and Aguarana taxonomies of their biological world [Berlin 78] [Lakoff 87]. At the level of the genus, the categories are so well differentiated in the world that folk taxonomies corresponded very well with scientific taxonomies at this level. So much so, that Berlin wrote that basic level categories are literally crying out to be named. Rosch et al also found that, while the American Sign Language (ASL) had far fewer signs (for certain categories of objects) as compared to words in English, the number of ASL signs for basic-level categories was almost as numerous as in English [Rosch et al 76]. Hence, we would expect that of all possible descriptions of the world, a description at the basic level is likely to be the most consistent.

[24] This argument has been pointed out by Paul Cohen (Personal Communication).

The counter-argument is that, if RA's types were basic level categories, then a characterization of a research field using these types is likely to result in a consistent knowledge base. However, while building RA's knowledge base, I encountered what I called the problem of 'concept drift': the notions problems and techniques mean slightly different things in different neighborhoods of the knowledge base (see Section 2.2.2). For example, the problem solved by [Samuel 59] was called the checker-learning-problem and the problem solved by [Winston 71] was called concept-learning-problem, thereby mapping the notion of a problem to objects at different levels of genericity. If problems and techniques were in fact basic level objects, then such concept drifts should not occur.

One could argue as follows: some concept drift is inevitable, even at the basic-level of conceptualization; however, of all levels, the concept drift is likely to be the least at the basic level; therefore, I would have seen even more concept-drift had I tried to characterize the field using some other types! At this point, there is no way to know.

This suggests some experiments that are likely to be of great value in building large knowledge bases. If different people are shown an object, say a house, and asked to build a knowledge-base that describes the house, are these knowledge bases likely to be most consistent at the basic-level? Is there at all a correlation between consistency and the level of abstraction?

Why is this question of any great interest? One of the biggest problems in building and maintaining large knowledge bases is inconsistency among different uses of a concept (see [Lenat & Guha 89] [Loiselle & Cohen 89]). A standard methodology in building knowledge bases is to start at the very top — i.e., the most abstract and vacuous categories such as 'any-thing' and 'any-action' — and grow the knowledge-base from the top downwards. Let us suppose that our experiments (suggested above) show that the basic level is the level of abstraction at which our ontologies are most consistent. In that case, it makes most sense to build a knowledge-base from the middle-out: first, encode the knowledge corresponding to the basic level categories because these are categories that have the most 'semantics'; then build the generalizations and specializations. This is almost identical to Roger Brown's characterization of how children learn: Brown speculated that children first learn objects at a certain intermediate level of abstraction and then proceed to lower and higher levels by "achievements of imagination" [Brown 58].

5.4.2 Research Schemas: Basic-Level Structures?

One could speculate that research schemas are basic-level research strategies: they are composed with basic-level objects that are related to each other in a structural configuration that is also (both) associative and discriminative (see Section 5.3.2).

The statement above interesting, even as a speculation: while the term 'basic-level' is normally used with respect to semantic levels of abstraction, to the best of my knowledge, this is the first time that anyone has even speculated that the idea of basic levels might

be applicable to relational structures. To understand what this means and why this is interesting, let's briefly explore the idea of a schema.

There are at least two different ways in which the term 'schema' can be used. The first use of the term corresponds to a schema as a processing convenience that groups all relevant items together in one place[25]. Hence, a schema is simply a feature vector of attribute value pairs. Under this interpretation, the speculation that there might be basic-level schemas is nothing new: almost all the researchers that have dealt with basic level categories have dealt with feature vectors and have shown that basic-level categories are characterized by features that are both predictive and predictable. If schemas are no more than feature vectors, then the speculation that there are basic-level schemas is simply a restatement of the claim that basic-level categories exist.

A second use of the term 'schema' includes an underlying assumption that the whole is more than the sum of the parts. When a set of entities are brought together in a schema, the schema, in addition to the overt properties of these entities, somehow also represents the interactional properties of these entities. For example, 'buying' may be seen as a schema that involves two abstract transfers (ATRANSes) — the transfer of an object from the seller to the buyer in return for the transfer of money from the buyer to the seller. However, 'buying', as a schema, stands for more than the two transfers. It includes several covert interactional properties that emerge when these two atomic actions interact: the deep meaning of 'buying' involves, among other things, the causal connection between the two actions and the assumptions about the role of money and goods in our society[26]. Under this interpretation, a schema is not simply a set of features. The features are related to each other and these relations reflect interactional properties. For this interpretation of schemas, a feature vector representation is simply inadequate.

The standard way in which basic level categories are defined is through the use of the two terms *predictiveness* and *predictablity* [Gluck & Corter 85] [Fisher 86]. In this approach, basic level categories are categories whose features are both predictive and predictable: i.e., C is a basic-level category if it predicts that an object belonging to that category will have the feature f, and conversely, if an object is known to have the feature f, one can predict that it belongs to C. This characterization of basic-levels is inadequate to even talk about basic-level schemas if schemas are more than feature vectors.

In Section 5.2.5, I introduced an alternative characterization of basic-level categories using the ideas of *associativity* and *discriminability*. These terms get away from the notion of a feature and use a more abstract notion of a *description*: as such, a description can be anything, including features, feature clusters, a category name, or even a

[25]In fact, a lot of frame-based languages cater precisely to this use of the term.

[26]While the object that is ATRANSed in the first ATRANS can be anything, the object in the second ATRANS has to be money. Barter, typically, is not considered buying, except metaphorically, as in "John bought good grades with his looks."

structural configurations of relations (i.e., schemas). Hence, the characterization of basic level categories as a description that is both associative and discriminative provides a vocabulary for even talking about basic-level schemas.

I believe that the question, "Do basic-level schemas exist?" is interesting both as a psychological question, and as a question pertaining to the design of knowledge structures. Some of the implications of the latter are explored in the next subsection.

5.4.3 Functional Flexibility: Due to Basic-Levels?

One of the interesting things about research schemas is that they support RA's several different functionalities. The point to note is that they do not just support different functions, but act slightly differently with respect to each function: with respect to the chronological summarization component, they act as a description of research papers; with respect to the suggestion component, they act as intentional rules for actions; with respect to the analogical summarization component, they stand for the research strategy of a paper. Thus we can say that research schemas exhibit *functional flexibility*[27].

As a description of a paper, a research schema can have any arbitrary structure. A paper can include an arbitrary amount of background material and hence can include an arbitrary number of relations in the **ref** (see Section 5.3.2). In this dissertation, we did not not consider the nuances of natural language communication, but there seems to be some evidence that a description should be at the right level of abstraction to be understandable[28]. Hence, let's assume that a description of an individual item can have arbitrary structure, but needs the 'right' level of abstraction.

Next, let's consider the use of research schemas as rules. A rule needs the right structure — i.e., the 'if' part cannot be anything but should contain just the basis (i.e., necessary and/or sufficient conditions) for the applicability of the 'then' part. However, a rule need not view the world from any 'right' level of abstraction. For example, if you want to state the association between executives and ulcers, the following is one possible way to state this association:

> If x is an executive, and if x lives in a penthouse apartment, then x will get ulcer.

[27]This phrase has been used by Abelson and Black to refer to a representation that supports different functions. For a discussion on why this is important — both psychologically and computationally — see the introduction to [Galambos et al 86].

[28]Eleanor Rosch makes an argument for this by using a review of a novel called *Decades* by Leslie Garris in the Ms. Magazine: "And so, after putting away my 10-year-old Royal 470 manual and lining up my mongol number 3 pencils on my Goldsmith Brother Formica imitation-wood desk, I slide my oversize squirrel-skin L.L. Bean slippers and shuffle off to the kitchen. There, holding *Decades* in my trembling right hand, I drop it, *plunk*, into my new Sears 20-gallon, caledon-green Permenex trash can" (see [Rosch 78]). Also see [Cruse 77]. For a computational approach to using basic-level categories in man-machine communication, see [Peters & Rapaport 90].

The above is a bad description of the statement "executives will get ulcer," because it has the wrong structure — it includes more than what is necessary in the 'if' condition. However, the following is an ok description:

> If x is a male executive, then x will get ulcer.
> If x is a female executive, then x will get ulcer.

In other words, there is no cosmic compulsion — except perhaps to reduce the amount of typing — for a rule to view the world at any 'right' level of abstraction, so long as a rule-base as a whole can state the correspondence between associated concepts (e.g., executives and ulcers)[29]. Hence a research heuristic y that pertains to all problems can in fact be stated as the following two rules:

> If x is a learning-problem, then suggest y.
> If x is a performance-problem, then suggest y.

So far, we have seen that research schemas, as a description of individual papers, are oblivious to the structure of the schemas, but *might* be sensitive to the level of abstraction. Research schemas, as heuristic rules, are oblivious to the level of abstraction, but are sensitive to the structure. How about the use of schemas as memory indices? Indexing, as it is modeled in this dissertation, is a categorization problem: an indexing scheme should associate all like objects and none of the unlike objects so that a given probe retrieves all the relevant objects and none of the irrelevant objects. A research schema, as a memory index, should categorize a given paper at a level of abstraction such that the paper will retrieve all papers (whether they deal with learning problems or performance problems) similar to that paper with respect to the level of abstraction; also a schema should retrieve all papers that have the same underlying strategy, irrespective of how much background material the actual paper contains for descriptive purposes. For example, in Section 5.3.2, we considered two possible descriptions of the paper '[Utgoff 84].' The first one contained just enough background to state the motivations for the paper, whereas the second contained practically a history of the field. An indexing mechanism should be able to categorize a paper such that all papers with the same underlying research strategy (irrespective of the amount of background in the actual paper) are associated with each other. Therefore, research schemas, as an indexing mechanism, are sensitive to both the structure of the schema as well as the level of abstraction of the types in the schema.

[29] Biw Could this be the reason why expert systems have such a hard time trying to describe or explain their reasoning? (For example, Brachman and Levesque, in describing Mycin, write: "... the 'explanations' taken from tracings [of rule applications] are no longer very convincing as realistic explanations [Brachman & Levesque 85]). Since descriptions or explanations require the 'right' level of abstraction, and rules do not, is there a mismatch between explanations and rules?

I would speculate that the functional flexibility of research schemas derives from the fact that research schemas, as memory indices, cater to the most stringent condition — i.e., they are built with types that are both associative and discriminative, linked into structural configurations that are also both associative and discriminative; therefore, the other uses (descriptions of papers and heuristic rules), which have less stringent requirements, simply fall out of research schemas. Admittedly, this is a wild speculation at this point, but it suggests some experiments that can be used to verify the speculation: if we design different knowledge structures in other domains, can we expect the same kind of behavior? Alternately, can we expect that any knowledge structure that exhibits any degree of functional flexibility will have some of the same properties of research schemas? I believe that such abstract analysis of the properties of knowledge and knowledge representation is totally unchartered territory, and constitutes a fruitful direction for exploration.

5.4.4 Basic Levels: A Universal Bias?

In Section 5.3.1, I explained RA II's generalization strategy in terms of the category structure of its world: given a specific instance, RA II generalizes the instance into a variable constrained to a category that is both associative and discriminative. To show that this is in fact a correct explanation, I also showed what happens if we allow the acq relation: acq disturbs the category structure of RA II's world by introducing a further discrimination between learning-problems and performance-problems.

In some ways, the explanation of RA II's generalization strategy (in Section 5.3.1) is not very convincing or fool-proof; this is because RA's toy world is indeed extremely trivial. The number of features for each type — i.e., the number of relations in which a type can partake — is too small for the analysis to be called a 'result.' Further, since the categories are defined by such a small number of features, the category structure of the world is quite fragile: the addition of a single new feature (such as acq) can disturb the category structure in a significant way.

Despite these problems, the question raised by the analysis is quite valid and intriguing. In a nutshell, the question is, "Do basic-level categories constitute a universal bias for generalization?" The rest of this subsection is meant to articulate this question.

Under the classical view of categories, there is no particular assumptions about the structure of the world. Categories are assumed to be arbitrary grouping of objects in the world, with each grouping corresponding to some intension. Categorizing a given object in one way is good for one purpose, and categorizing it in another way is good for a different purpose. With respect to learning and generalization, this raises the general problem of induction: i.e., if a learning system is given a set of instances of some concept, these examples can be generalized into any number of categories. The following example is similar to the one used by Thomas Mitchell in his *Computer and*

Thought Lecture [Mitchell 83] to to illustrate this problem. Let's assume that I tell you I have a concept in mind and that 2, 4, 6, 8 and 10 are positive examples of the concept. Given this, you have no 'basis' for generalization: maybe the concept is 'numbers less than 10,' maybe it is 'even numbers'; maybe it is 'numbers less than 571'; it could even be 'non-primes except 2', which would be rather odd. For you to induce the concept from only a subset of examples, you need some kind of a *bias*, i.e., some criterion for preferring one concept hypothesis over the others. This is also the reason why inductive learning is modeled as a search process — to induce a concept from a set of examples, you are searching in a space of concept hypotheses to choose one among many possible hypotheses [Mitchell 81].

Let's first see how the problem of inducing a concept from specific instances is accomplished in the two major learning paradigms, SBL and EBL. In SBL, the system designer encodes the system's inductive bias. The bias can be encoded in a number of different ways — by restricting the concept description language in some way so that the system can only state a small number of concepts among all the possible ones (see [Utgoff 84] for a discussion), by preferring the concepts that have the simplest syntactic description and so on[30]. Perhaps the best known form of encoding an inductive bias is a hierarchical feature space used in the version spaces approach [Mitchell 78] [Mitchell et al 83]. Let me give a brief and fairly abstract description of this approach by using the illustration in Figure 5.11.

Figure 5.11 shows a world of objects spanned by a feature space. As in Figure 5.3, these objects can be grouped into arbitrary categories. However, in the version spaces approach, the system designer defines a small set of hierarchical categories as the useful categories in the world. During learning, the learning system is presented with a sequence of positive and negative examples of a concept. These are shown as shaded objects P1, P2, N1, and N2. Let's assume that the examples are presented to the system in the following order: P1, N1, P2, and N2.

When the system is first given the positive example P1, it immediately jumps up the hierarchy and assumes that the concept to be learned is at least as big as C1 but can be (in the general case) the whole space. When it is presented with the negative example N1, the system specializes the general hypothesis and assumes that the concept is no bigger than C4. When presented with P2 as a positive example, the system looks for a category that includes both C1 and C2, and jumps to the hypothesis that the concept is at least as big as C3. Finally when presented with N2 as a negative example, it concludes that the concept is no bigger than C3. Since the specific and the general hypotheses about the concept now match — i.e., the concept is at least as big as C3, and the concept is no bigger than C3 — the system has learned the concept C3 based on the encoded bias and the multiple examples.

[30]See [Dietterich 86] for a description of some syntactic biases; Dietterich shows that synatactic biases make a system's learning indescribable at the knowledge-level.

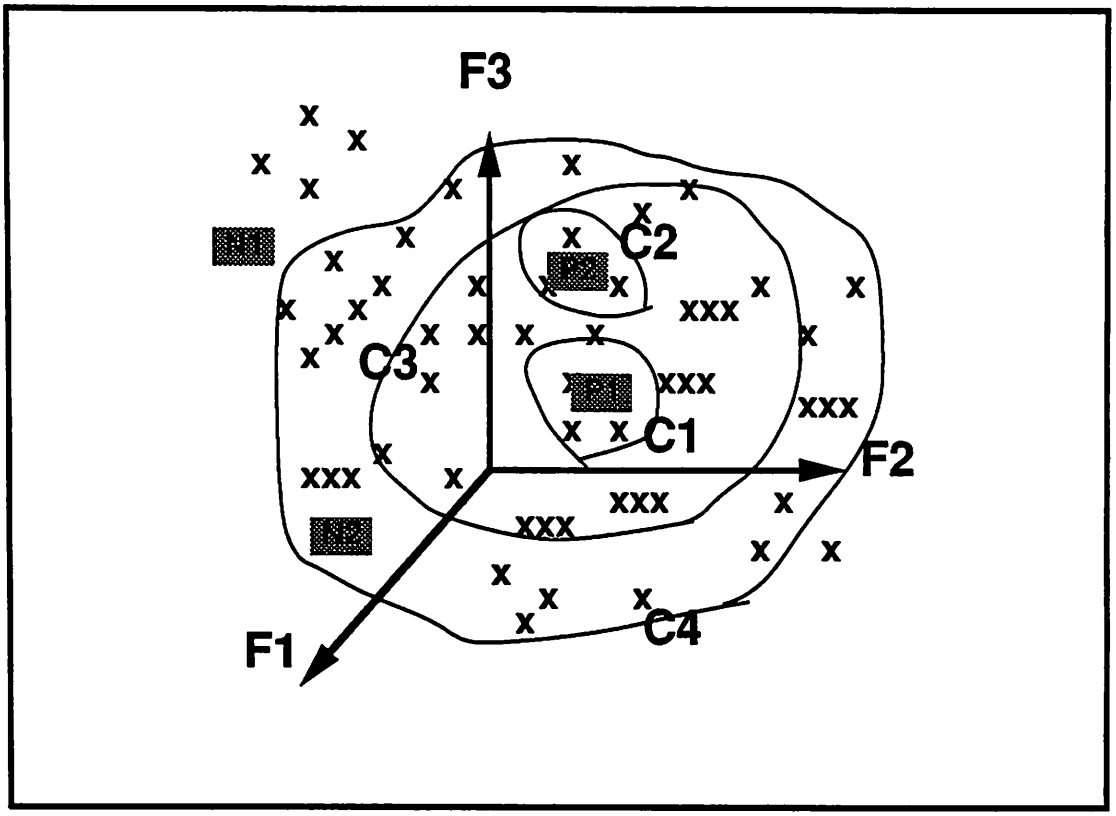


Figure 5.11: A Pictorial View of Version Spaces

In a retrospective analysis of the version spaces approach, Mitchell argued that the generalizations made by this technique were *unjustified*, i.e., the learning system that uses this technique has no knowledge of the purpose of the concepts [Mitchell 83]. The system appears to learn mainly because the system designer — who knows the purpose — encodes only the useful concepts with respect to the purpose at hand. This led Mitchell and his colleagues to develop a learning technique that uses the knowledge of the domain — the causal connections among the concepts, an explicit representation of the system's goals etc — to generalize a specific instance into a category. This style of learning has come to be known as Explanation-Based Learning[31,32].

Let's see how the EBL approach handles the problem of generalization, i.e., the induction of a category from a specific instance. Figure 5.12 depicts a world of objects in a feature space. The shaded object, P, is given to the learning system as a positive example of a target concept; in other words, the system is told, "Here is an example of a cup. Now learn this concept." The learning system, using its domain theory, constructs an explanation as to why the given instance belongs to the target concept. This explanation isolates those features that are relevant to the concept from those that are irrelevant. Assume that the system, using its domain theory, has reasoned that P is a member of the target concept since P has a certain value for the attribute F2, and that the other attributes F1 and F3 are irrelevant. Now all objects that have a similar value for attribute F2 are assumed to belong to the target concept and we say the system has learned this concept (shown in the figure as a blob). Thus, in EBL style learning, the domain theory is used to generalize a specific instance into a concept description.

What the basic-level effect shows is that the structure of the world is not arbitrary. Objects in the world are not uniformly distributed across a feature space; since features tend to cooccur, objects in the world occur in well-differentiated clusters. Basic-level categories reflect this inherent category structure of the world; to use Rosch's phrase, the basic-level categories "cleave the world at its seams" [Rosch et al 76][33].

Figure 5.13 illustrates the view of categories suggested by the basic-level effect. The question raised by this dissertation is that if the world is in fact structured like this, then does generalization become a fairly simple process? From a given instance of a concept, does it make sense to immediately generalize the instance to its basic-level category? Referring to Figure 5.13, if we are given P as an example of a concept to learn, isn't

[31] Other researchers have also arrived at EBL, but through different routes. See Chapter 2.

[32] Btw Mitchell's analysis is stated in the context of the LEX system that uses the version spaces approach to learn heuristic concepts in the domain of symbolic integration. See [Mitchell et al 83] for a description of LEX. Richard Keller's METALEX system learns the same heuristic concepts as LEX, but uses an explicit representation of its learning goal [Keller 87].

[33] Btw As a philosopical aside, I should mention that Rosch does not claim that the world is structured in and of by itself. The structure is imposed by us, not necessarily consciously, but simply because of the way we are. For example, the world of a dog is likely to be very different: since the dog has a superior sense of smell, its world of smells is likely to be lot more structured than a human's world of smells. For an articulation of this idea, called *embodiment*, see [Lakoff 87].

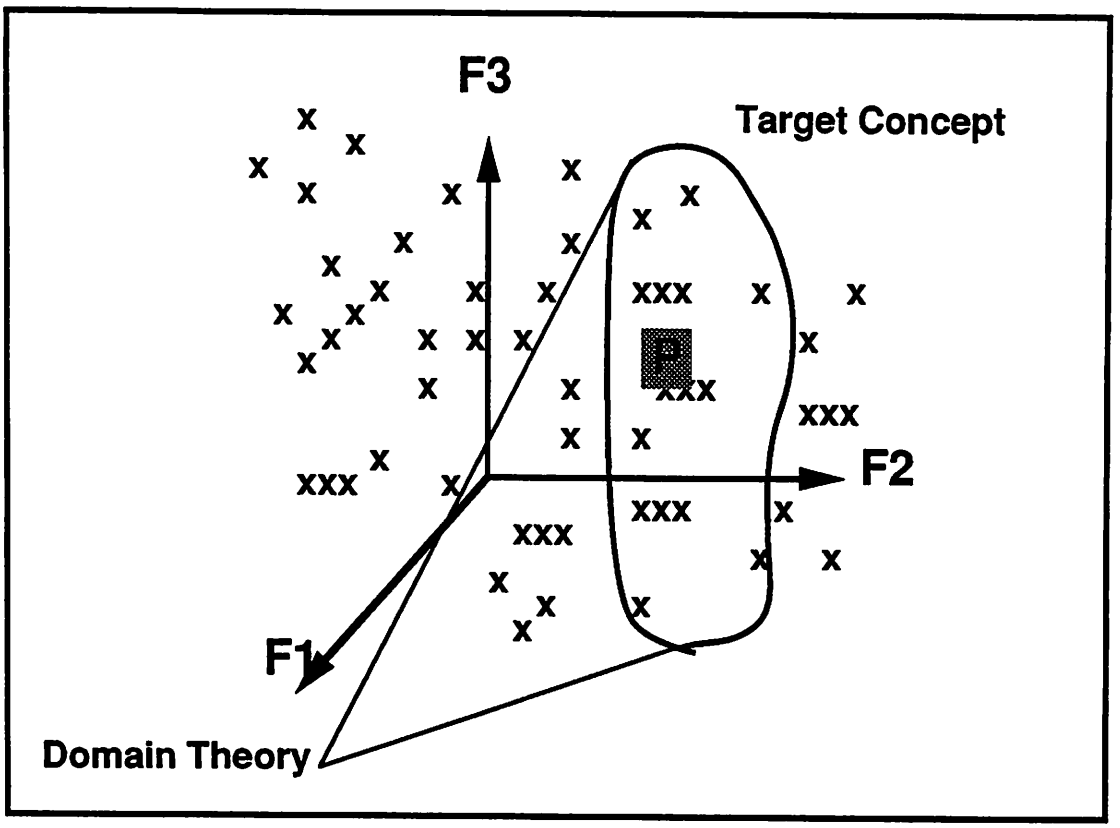


Figure 5.12: A Pictorial View of EBL

C the most likely and plausible generalization to make? In more abstract terms, do basic-level categories constitute a universal bias for generalization?

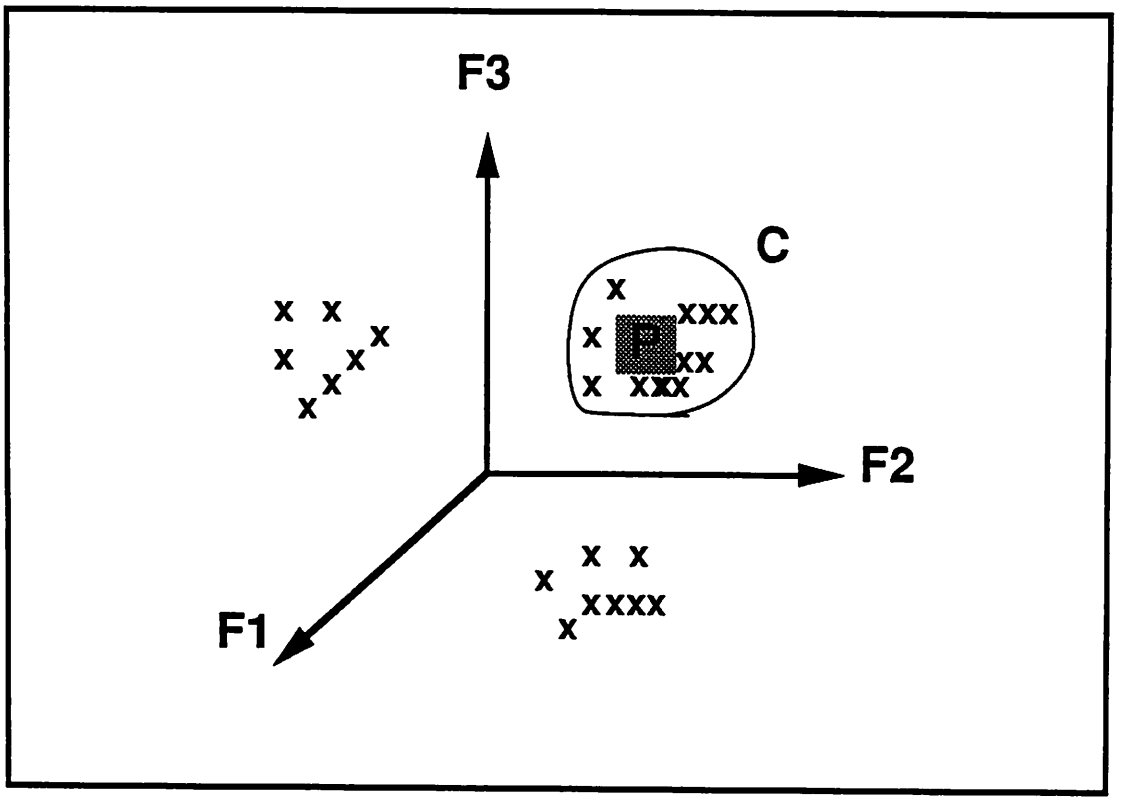


Figure 5.13: A Pictorial View of Basic-Level Generalization

This question suggests two related directions for exploration. The first is that, assuming that the world *is structured*, do we have a new and simple technique for generalization?[34] One method for answering this question involves building a learning system whose world is structured as shown in Figure 5.13, but is much larger than the toy world of RA. In such a world, we can evaluate the tradeoffs among the amount of work needed for generalization, the accuracy of the learned concepts, and their functional flexibility; this tradeoff can be calculated and compared for two different kinds of learning techniques, one that makes no assumptions about the structure of the world

[34] BIW An interesting, if somewhat far-fetched, analogy can be drawn between this question and the typesetting system Tex [Knuth 84]. Tex provides several sophisticated techniques for typesetting arbitrary text. Consequently, Tex is quite a difficult system to use. Leslie Lamport noticed that the world of documents is not arbitrary, but is quite structured. People do not write texts that include, for example, a mathematical equation in the middle of an address for a letter. Instead, people write articles, books, letters, and within these, chapters, sections, footnotes (thank you, Leslie), itemized lists, and so on. Thus, Lamport's Latex system [Lamport 86] provides a good tradeoff between flexibility and ease of use. Latex caters to a structured world, and is perhaps used far more widely than Tex ever was.

and the one that simply generalizes instances to a basic-level category. At this point, I would speculate that the latter technique (that simply generalizes instances to basic level categories) is likely to have the best tradeoff with respect to the three parameters mentioned above. If this is the case, then we have an important result for machine learning: with all tradeoffs considered, if basic-level categories are the best categories for generalization, then learning *cannot be considered a search process.* If generalization is simply the process of jumping to a basic-level category, then the basic-level category is the only hypothesis considered, and 'search' is not the best way to describe this process[35]. For some related results in human generalization, see [Oshersen et al 90].

The second, perhaps more crucial, question is, "Is the world really all that structured?" While it is true that Rosch's work has demonstrated this for a small number of categories of natural and man-made objects, it is by no means clear that Rosch's result is applicable to all categories in general; how true is this result for abstract categories such as, say, the category of ideas in Computer Science? There is some evidence that basic-level categories exist in the world of abstract categories as well [Adelson 85]. However, for us to base a theory of learning on the assumption that the world is structured, we need more evidence that this assumption is a reasonable one. I believe that investigations into the category structure of the world is crucial for machine learning. I also believe that questions of this sort that pertain to the ontology of the world require an entirely different methodology than the current approach that deals with machine learning mostly at the epistemological or symbol level[36].

5.4.5 A New View of Representation?

In most people, the question, "What is a good representation?" elicits the response, "That depends on your purpose." Our fascination with the so-called mutilated checkerboard problem illustrates this attitude toward representation[37]. The standard cliche in

[35] Biw To explain this, let us go back to the example before. If told that 2, 4, 6, and 8 are instances of a concept, Mitchell would claim that these examples can be generalized to any number of possible concepts; for you to choose one concept over all others, you need to search a space of concept descriptions. However, most humans, when given 2, 4, 6, and 8, will probably jump to the generalization 'even-numbers', without for a moment considering the infinite number of other generalizations such as 'all numbers less than 541.' Even with classical categories like 'even-numbers' and 'all numbers less than 541', the former has a certain conceptual coherence determined by a set of co-occuring properties: we can say a lot of interesting things about 'even-numbers', but pretty much the only thing that members of the concept 'all numbers less than 541' have in common is their intensional definition. While there are an infinite number of potential generalizations for 2, 4, 6, and 8, our generalization of this sequence considers precious few. Hence, if the world in in fact well structured, the characterization of generalization as search, it appears to me, is not a particularly revealing one.

[36] For a discussion on ontological and epistemological level theories, see Chapter 6.

[37] Mutilated checkerboard problem: You are given a checkerboard from which the diagonally opposite end squares have been removed. The problem is to cover the checkerboard completely with a set of dominos. This is a computationally complex problem if you consider the various ways in which you can cover the board with dominos. However, if you represent the problem in terms of the number of

AI is that 'representation is important.' A stock example to drive home this point is the mutilated checkerboard problem. This problem is often taken to mean that you need the right representation that is well-suited to each specific purpose. Alternately, the purpose is assumed to determine the goodness of a representation. Finally, this assumption also has a methodological component: since representations ought to be purpose-specific, the best way to evaluate the goodness of a representation is through a performance evaluation with respect to a single task that the representation was designed for. Collectively, these assumptions constitute what I call the *Mutilated Checkerboard Syndrome.*

I believe that the view of representation championed by the mutilated checkerboard problem results from a confusion between *problem* representation and *knowledge* representation. While a representation of a mutilated checkerboard as 32 black squares and 30 white squares is a good one *for solving the particular problem* of dressing the checkerboard with dominos, such a representation does not, by a long shot, capture the *knowledge* you and I have of this mutilated checkerboard. While understanding the properties of good problem representations is an important research problem in its own right[38], problem representations ought not to be confused with knowledge representations.

As I see it, the basic-level effect points out that there are inherently good descriptions for objects, descriptions that are good not because they help you solve isolated problems, but because they capture the real essence of the objects (as descriptions that are both associative and discriminative). Hence, such a description includes all potentially important knowledge pertaining to an object; this description is, therefore, likely to be useful with respect to most purposes. I believe that any theory that calls itself a theory of *knowledge* representation ought to reject the mutilated checkerboard syndrome and its associated assumption that you need a different representation for each isolated purpose. In this regard, the view of representation suggested by basic-level categories is a very promising alternative for exploration.

> *Their shaven skulls gleamed like polished shoes and pointed directly at the waking sun-god Ra, because Abdullah had figured out that Mecca must lie approximately in that direction.*
>
> —Thor Heyerdahl, The RA Expeditions.

---

black and white squares, the problem is trivially simple: the mutilated checkerboard has more squares of one color than the other. Since each domino has to cover exactly one black and one white square, the problem of covering the mutilated board completely with dominos is unsolvable.

[38] Btw For example, see [Subramanian 89] for approaches to finding good problem representations as a reformulation of existing representations.

5.5 *Case Study: Hypo, Chef, and Pro

To illustrate that the basic-level effect is not just a psychological phenomenon, this section discusses three Case-based reasoning systems that involve a complex categorization problem — indexing a case-base such that the system can retrieve the most relevant precedents of a given case. I will argue that the design of these systems is strongly influenced by the need to find a description of cases that is both associative and discriminative.

5.5.1 Case-Based Reasoning

Case-Based Reasoning (CBR) is an emerging AI paradigm that is based on the intuition that human problem-solving is often guided by reasoning based on precedents. For a survey of the evolution of the CBR ideas, see Part I of [Kolodner 88a]. In contrast to rule-based expert systems (whose knowledge is organized in terms of 'if-then' rules, as in [Davis et al 77]) a CBR system uses an episodic memory of cases, i.e., a memory of prior problem-solving episodes. When a new problem instance is presented, a CBR system retrieves a set of 'similar' precedents to help it solve the current problem.

'Indexing' is a crucial notion in the design of the episodic memories of CBR systems. The cases in memory need to be indexed (described or coded) in such a way that a system can retrieve the most similar cases in response to a given problem instance. This can be stated as a categorization problem: Given a new case (a stimulus), the indexing problem is the problem of finding the right indices (the right description or coding) that associate the case with similar cases and discriminate it from dissimilar ones. Thus the indices should maximize both associativity and discriminability — the chosen indices should not only retrieve all the right precedents, but also should not retrieve any of the wrong precedents.

Any interesting problem space has some problem instances that are similar to each other, and some instances that are dissimilar to each other. The indexing scheme of a CBR system should index problem instances to reflect this inherent category structure of the problem space. Unfortunately, the category structure of a problem space may not be uniform everywhere in the problem space. As an example, consider the 8-puzzle. Given an initial state (a certain random configuration of tiles) the 8-puzzle problem is to find a series of moves to reach a particular goal state: ((1 2 3) (4 5 6) (7 8 0))[39]. If we code 8-puzzle solutions in terms of the goal states they satisfy, we maximize associativity but lose discriminability completely: since all problem instances look the same from the point of view of the goal state, the goal state associates a given problem with all problems similar to it, but does not discriminate between any problems at all. In contrast, a coding of solutions in terms of initial states maximizes discriminability

[39] 0 stands for the blank square.

(each board is an entirely unique category), but does not provide any associativity (no two boards, under this coding, are similar). Thus the category structure of the 8-puzzle problem space is not reflected by a coding of the goal state nor by a coding of the initial states. This example illustrates that while inherent categories may be "out there", the indexing problem is a design problem that should identify the right coding that reflects this structure[40].

The next three subsections describe three CBR systems, Hypo, Chef and Pro, and show that the design of these systems — their problem solving strategy and their notion of what a case is — are determined by the need to optimize associativity and discriminability. This analysis is concerned only with the category structure of these systems and does not get into the details of their reasoning strategies[41].

5.5.2 Hypo

HYPO is a CBR system built by Kevin Ashley as part of his dissertation research (see [Ashley & Rissland 88] [Ashley 88]). Hypo is designed as a lawyer's assistant in the domain of trade-secret litigation. Given a legal case, Hypo constructs a legal argument that is favorable for its client. Without loss of generality, let's assume that Hypo represents the defendant. In constructing an argument that is favorable for its client (the defendant) Hypo uses other precedents that resulted in a win for the defendant. The categorization task in Hypo is to index its memory of legal cases in such a way that Hypo is reminded of the cases most similar to a given case.

In Hypo, cases are described in terms of dimensions: predicates that establish whether certain conditions hold for a certain case. Some of the dimensions with respect to the domain of trade secrets litigation are *employee-paid-to-switch, employee-brough-product-tools, corporation2-saved-expense, exists-info-re-security-measures* and so on. Each case is seen as a point in an n-dimensional space spanned by legal predicates. A legal argument may be seen as a mapping of a given case, i.e., a 'current fact situation' (*cfs*), to a goal state (a win for the defendant). Hypo starts by first placing the

[40] BIW Problem instances of an 8-puzzle are continuously distributed, and hence there is not much inherent category structure to the 8-puzzle problem space. Thus any case-based solution to the 8-puzzle has to impose a category structure. In our laboratory, two radically different programs were written for the 8-puzzle problem. The program I wrote was goal-driven: even though the goal state has no discriminability, sub-goals such as 'get 1 to its destination' 'get 2 to its destination' etc are more discriminative. Eight cases (one to get each tile to its final position regardless of its initial position) were used to code the 8-puzzle problem space. Not to be content with obvious solutions, Wendy Lehnert built a forward reasoning CBR system as follows: even though the starting states have no associativity, her solution buys associativity by categorizing all starting states that have the same 'coarse code' as similar. It turns out that boards with the same coarse code will map to the final state through the same sequence of moves, i.e., coarse-codes and move-sequences co-occur. See [Lehnert 87b] for details.

[41] Much of the material of this section is based on a paper I wrote for a CBR workshop [Swaminathan 88b]. When that paper was published, I was unfamiliar with categorization and the the basic-level effect.

cfs onto the n-dimensional space of legal predicates; all cases in the case-base that are close-by points to the *cfs* are deemed relevant, and partake in Hypo's 3-ply arguments. In problem-solving terms, Hypo is a forward reasoning system: it starts from the initial-state (the *cfs*) and constructs an operator sequence (an argument) to map it onto a goal state ('win' for the defendant).

Why is Hypo designed this way? Given that Hypo has a clearly stated goal, why couldn't Hypo be a backward reasoning system, reasoning from its goal? The answer lies in Hypo's need to find a coding (of the cases) that is both associative and discriminative. The final states for legal arguments allow only two categories: 'win' for the defendant or 'loss' for the defendant. Thus a coding of the cases in terms of their final states has very low discriminability. From the point of view of the goal state, there are only two kinds of legal cases, those that win, and those that lose. In contrast, a coding of cases in terms of the starting state (*cfs*) will result in much higher discriminability. The various dimensions (about 31 dimensions are used in one implementation [Ashley 88]) ensure that a wide range of cases are discriminable if coded in terms of their *cfs*.

But how do we know that these starting states are associative? The *stare decisis* doctrine of legal reasoning is based on the tenet that human disputes tend to be similar. Restated, this doctrine enables the lawyer (and Hypo's designer) to assume that cases are not uniformly distributed in Hypo's space of legal predicates, but occur in well-defined clusters of similar cases. The abstract legal predicates established by the courts and legal experts attempt to capture precisely these similarities among cases. Hence Hypo is a forward reasoning system since the starting states (rather than the goal states) are codable in terms of the category structure of Hypo's problem space. Section 5.7.5 gives a pictorial interpretation of Hypo's problem space.

In summary, Hypo's design as a forward reasoner is not arbitrary. It is strongly influenced by the fact that Hypo's goal states are low in discriminability and hence cannot be used to code the legal cases in Hypo's memory. The starting states have a much higher discriminability; the *stare decisis* doctrine assures that the starting states are likely to be quite associative as well. Hence, Hypo codes the category structure of the legal cases in terms of the starting states or the *cfs*. Since cases are retrieved on their starting states, Hypo is forced to be a forward reasoner starting with the *cfs*, constructing an argument that maps the starting state to the goal state[42].

5.5.3 Chef

Chef is a case-based planning system built by Kristian Hammond as part of his dissertation research [Hammond 89]. Chef uses an episodic memory to plan recipes for Szechuan Chinese dishes. Given some specification for a dish — e.g., make a stir-fry

[42]Edwina Rissland has pointed out that my characterization is a somewhat simplified treatment of Hypo's problem space.

with chicken and snow peas — Chef constructs a recipe to actualize the dish. To do that, Chef retrieves similar recipes (e.g., a recipe for beef and broccoli) from its episodic memory and modifies them until the given specification is satisfied.

In Chef, recipes are indexed or coded in terms of the goal states they satisfy. Thus a recipe for a stir-fry dish that includes beef and broccoli is coded as a meat and vegetable dish. A new specification such as 'plan a recipe to include chicken and snow-peas' will retrieve the beef and broccoli recipe because they both satisfy the same abstract goal — make a stir-fry that includes a meat and a vegetable. The retrieved recipe is modified until it satisfies the current specification. In problem-solving terms, Chef is a goal-driven, backward-reasoning system: given a goal (specifications for a dish) it constructs a sequence of moves (a recipe) to map an initial state (a state in which all ingredients are available) to the goal state.

Why is Chef a goal-driven system? The starting states in Chef have no discriminability: Chef assumes that all ingredients are always available, so there is only one starting state, and thus only one category[43]. For Chef, starting states are completely associative (all recipes look the same if coded in terms of their starting states) and completely non-discriminative.

Let's consider the goal states. Chef assumes that it will be called in to construct different recipes (stir-fries, souffles, pastas etc) with different ingredients. Hence the categories circumscribed by the goal states are differentiated from each other. As regards the associativity of the goal states, Chef assumes that all meats, by and large, behave the same way: they take similar amounts of time to cook, they sweat while cooking, and make the vegetables soggy etc; similarly, all vegetables behave the same way: they need to be chopped, they will become soggy if cooked too long and in too much water etc. Thus the ingredients such as 'vegetable' and 'meat' are well differentiated categories of feature co-occurrence. It is assumed that categorizing a goal specification for a 'chicken-and-snow-peas stir-fry' as a specification for a 'meat and vegetable stir-fry' will result in the retrieval of the recipes that have the most similar cooking steps. Therefore, Chef is a goal-driven system since the goal states are codable as both associative and discriminative. Section 5.7.5 gives a pictorial interpretation of Chef's problem space.

In summary, the design of Chef as a goal-driven backward reasoning system is not arbitrary. This design decision is strongly influenced by the fact that Chef's starting state is totally non-discriminating, and does not reflect the inherent differences between different kinds of dishes and recipes. Chef's goal states have much higher discriminability — there are different categories of recipes that Chef is called in to plan. By assuming that all meats can be cooked the same way, Chef ensures that a coding of the recipes in terms of the goal states will enable it to transfer the cooking steps involved in one

[43] BTW Alternately, consider how I normally cook: I open the fridge and see what I have to cook with. I plan what to cook depending on what I have. My cooking is often reasoning forward from the starting states because my starting states are discriminative. The goal state is always 'make a stew'!

kind of meat to another. Hence the goal states are associative as well. Since the goal state reflects the category structure inherent in a class of recipes, Chef codes the recipes in terms of the goal states they achieve. This forces Chef to be a backward reasoning system that reasons from its goals to map the same starting state to different goal states.

5.5.4 Pro

Pro is a CBR system built by Wendy Lehnert [Lehnert 87a]. Pro uses an episodic memory of words and their pronunciations to generate the pronunciation for new words (i.e., words it may not have seen before). During the training phase, Pro is given a sequence of word-pronunciation pairs (e.g., SHOWTIME-*shōtīm*). Using this training set of examples, Pro stores its 'experience' (in pronouncing words) into a casebase. To build this case-base, Pro first segments the words into graphemes; for example, 'showtime' is segmented as {SH OW T I ME}[44]. The phonemes in the pronunciation are now associated with the graphemes in the segmentation as follows: and {SH/*sh* OW/*ō* T/*t* I/*ī* ME/*m*}. To take into account the effect of context on pronunciation, Pro assumes that the pronunciation of a grapheme is affected only by its neighbors on either sides, so that, for example, 'I' will be pronounced as *ī* if preceded by 'T' and succeeded by 'ME.' Thus, the following seven cases are generated from the example SHOWTIME-*shōtīm*:

(START**/*nil* START*/*nil* SH/*sh*)
(START*/*nil* SH/*sh* OW/*ō*)
(SH/*sh* OW/*ō* T/*t*)
(OW/*ō* T/*t* I/*ī*)
(T/*t* I/*ī* ME/*m*)
(I/*ī* ME/*m* END*/*nil*)
(ME/*m* END*/*nil* END**/*nil*)[45].

During the test phase, Pro is given a word and asked to generate its pronunciation. First, it segments the word into graphemes as above. For each grapheme, there might be a number of cases in memory that suggest how that grapheme might be pronounced in different contexts. These hypotheses are pooled into a constraint propagation network, and a unique pronunciation is generated by relaxing the network.

In problem-solving terms, the initial state for Pro is the spelling of the word, and the goal state is the final pronunciation. Pro is definitely not a goal-driven backward reasoning system because the specific goal state, the pronunciation of a given word, is originally unknown. The general goal — generate the pronunciation for the input word — is entirely associative and non-discriminative. In contrast, a coding of cases in terms

[44]The segmentation process is somewhat complicated and is not of great concern here. See [Lehnert 87a] for details.

[45]The tokens START**, END** etc are used to maintain the length of all cases constant

of the starting states, (i.e., the spelling of a word in its entirety), is completely discriminative, but non-associative: if words and pronunciation are stored in their entirety, then such a case is usable only if that very same word appears in the test set.

Given that neither the goal state nor the starting states are codable to reflect the category structure of the problem space, Pro resorts to splitting the cases into parts. If cases are single grapheme-to-phoneme mappings, such cases will be highly associative, but not very discriminative: for example, the graphemes corresponding to the single vowels, a, e, i, o and u, have different pronunciations depending on different contexts (e.g., 'o' in move, love, and cove are pronounced differently). Therefore a coding that associates all these pronunciations of a grapheme into one case is highly associative, but not discriminative. The assumption that the pronunciation of a grapheme is sensitive to its immediate context is an attempt to associate the occurrence of that grapheme in other words in the same context, and to discriminate its occurrence in other words, but in different contexts. As it turns out Pro's assumptions about its problem space does not result in a completely associative and discriminative coding: Pro finds that the case-base often generates multiple hypotheses about the pronunciation of a word, and it has to use a network relaxation procedure to dynamically make further discriminations[46]. Thus Pro is a memory-based reasoning[47] system in which words lose their separate identity. This is because the inherent category structure of its problem space is not reflected in the correlation between entire words and their pronunciation, but in the correlation between grapheme-sequences and phoneme sequences. Section 5.7.5 provides a pictorial interpretation of Pro's problem space.

In summary, Pro's design as a memory-based reasoning system is not arbitrary. For reasons stated above, Pro's goal states are not codable in a way that is both associative and discriminative. A coding of cases in terms of their initial states is completely discriminative, but non-associative. The inherent category structure of the problem space is really in the mapping from grapheme-to-phoneme sequences. By choosing these mappings as cases, Pro obtains a coding that is both associative and discriminative.

5.5.5 Synopsis

The three subsections above considered three CBR systems and showed that their design is strongly influenced by a need to find a coding or description that maximizes the associativity and discriminability of cases. This section provides a pictorial interpretation of the arguments above, and provides a synopsis in terms of feature correlations.

[46]Wendy Lehnert reports that even the network relaxation procedure may fail to choose a unique pronunciation for a word. She proposes extending the cases to include four (instead of the current three) graphemes to provide more context [Lehnert 87a].

[47]This use of the term 'memory-based reasoning' is due to Edwina Rissland [Rissland 87]. Rissland's definition is somewhat different from the use of the same term by [Stanfill & Waltz 86].

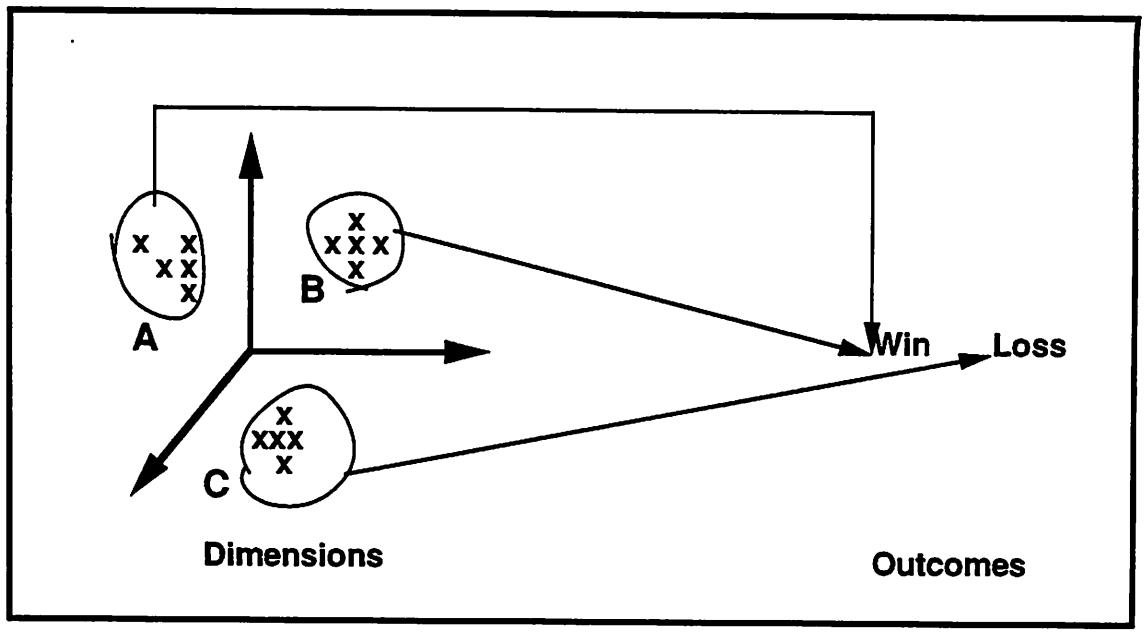


Figure 5.14: Hypo's Problem Space

Figure 5.14 depicts the problem space of Hypo. The abstract legal predicates or dimensions are used to cluster legal cases into similarity classes as shown on the left. The *stare decisis* doctrine ensures that the outcomes of cases are not random, but similar cases should have similar outcome. However, this is a one way correlation: while similar cases will have similar outcomes, entirely dissimilar cases can have the same outcome. For example, clusters A and B in the figure have the same outcome, even though the cases in A are dissimilar from cases in B. Alternately, the dimensions that circumscribe A and B predict the outcome (the feature 'win'), but the outcome does not predict the dimensions of A and B. Thus the dimensions capture the 'essence' of cases much more so than the outcomes, and hence reflect the inherent category structure of Hypo's problem space. This is precisely why Hypo codes its casebase in terms of the dimensions.

Figure 5.15 depicts the problem space of Chef. There are three entirely dissimilar kinds of recipes, stir-fries, souffles and pastas. There is a single starting state in which all ingredients are assumed to be available. Since this starting state co-occurs with all recipes, the starting state does not predict the recipe. The assumption about the co-occurring features of meats (they all require similar amounts of time to cook, they are sweat etc) and vegetables (they all need to be chopped, they all become soggy if overcooked etc) ensures that the cooking steps for these are similar. In other words these assumptions ensure that the cooking steps for related recipes will be more like paths p and q rather than paths $p*$ and $q*$. Hence, if Chef already has the recipe p, it can generate recipe q by modifying p in small ways. Thus the goal states of Chef's problem

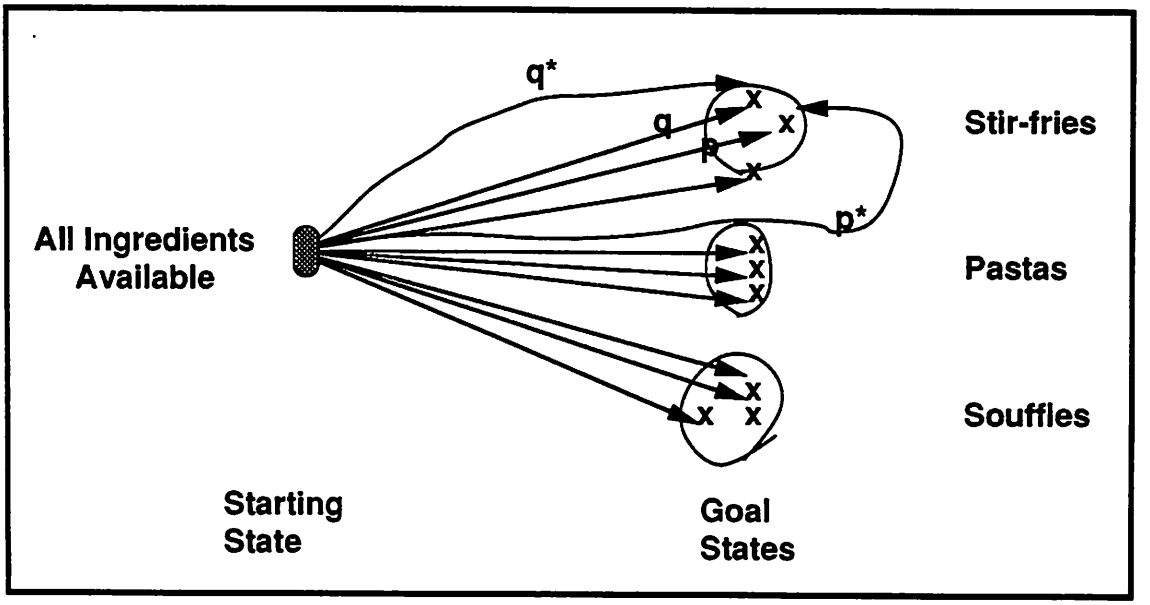


Figure 5.15: Chef's Problem Space

space reflect the inherent feature correlations among the type of dish, the ingredients that go in them, and how they are cooked.

Figure 5.16 depicts the problem space of Pro. If each word is assumed to be atomic, then the pronunciation of each word is entirely idiosyncratic; alternately, if words in their entirety are coded as cases, then this coding does not have any associativity. The figure shows a set of graphemes and their corresponding phonemes[48]. In contrast, a coding of cases in terms of single graphemes is also not adequate: the grapheme *c* co-occurs with different phonemes in different words. However, the grapheme sequence *bcd* co-occurs with the phoneme sequence $\bar{b}\hat{c}\bar{d}$ and the grapheme sequence *xcy* co-occurs with the phoneme sequence $\bar{x}\check{c}\bar{y}$. Hence the category structure of this space is reflected not in complete words, nor in individual graphemes. Grapheme sequences and phoneme sequences are the most reliably co-occurring features, and reflect the inherent category structure of Pro's problem space.

In summary, the fundamental objective for dividing the world into categories is to associate like stimuli with each other and discriminate unlike stimuli from each other. The work of Eleanor Rosch and other cognitive psychologists has demonstrated this principle in human categorization with respect to concrete objects: the basic-level effect shows a range of psychological phenomena associated with such well-differentiated categories. In this section, I extended the notions of associativity and discriminability to complex categorization tasks involving abstract categories, and showed that the need

[48]Note that the letters a, b, etc stand for variables and are not actual graphemes or phonemes themselves.

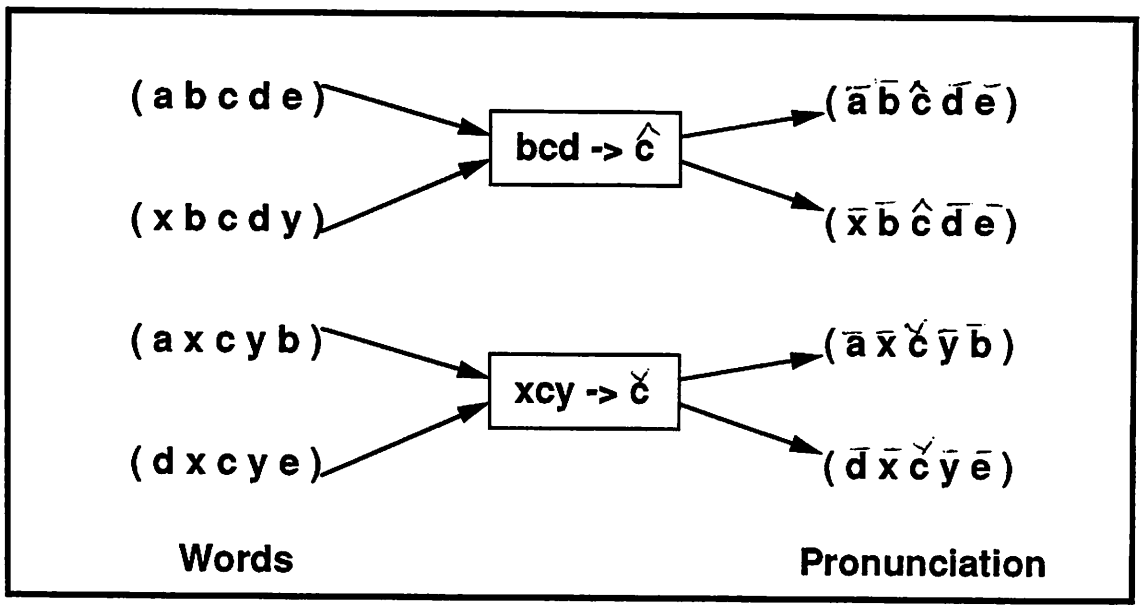


Figure 5.16: Pro's Problem Space

to maximize associativity and discriminability is not a mere psychological fact, but is in fact a design principle for any system that involves a categorization task.

5.6 Summary

Way back, in Section 5.1, we raised two questions about RA's learning strategy. In attempting to answer these questions, we made a foray into the basic-level theory of categorization. Later, in Section 5.3, I showed that RA's generalization strategy can be explained in terms of the category structure of RA's world: RA replaces constants by variables that belong to categories that are both associative and discriminative. I also showed that RA's assumptions about the structure of the **ref** ensures that a research schema, as a whole, is both associative and discriminative. In Section 5.4, I explored the implications of this analysis: in particular, two intriguing speculations emerge from this analysis: Do basic-level schemas exist? Do basic-level categories constitute a universal bias in generalization? Finally, in Section 5.5, we reviewed three CBR systems to see that their design is strongly motivated by their need to cluster their world in categories that are both associative and discriminative.

Did we drift to America because of unprecedented stupidity in handling wooden steering oars, or because of unprecedented skill in sitting on reeds? Here I do have a theory: Perhaps we got across because we sailed on the ocean and not on a map.

—Thor Heyerdahl, The RA Expeditions.

Chapter 6

RA and the Rest of the World

Human beings are so alike all over the world that it is natural for them to have similar notions.

—Thor Heyerdahl, The RA Expeditions.

Representational theories in AI can be classified into ontological or epistemological theories: an ontological theory specifies what kinds of knowledge are involved in what sorts of behavior; in contrast, an epistemological theory specifies how this knowledge is used in actualizing the behavior in a computer system under several pragmatic constraints.

Within the class of ontological theories, we can distinguish between content theories and content-independent theories. A content theory specifies all the various forms in which a particular generic kind of knowledge exists in the world: as such, a content theory is meant not only as a theory of a cognitive behavior, but in some sense, also as a theory of the world.

Further, we can differentiate between theories of knowledge and theories of memory, and correspondingly between knowledge representation and memory organization. This distinction is determined by whether the knowledge access problem in a theory is an epistemological or an ontological level concern.

Under this classification, RA is best understood as an ontological theory, in particular, as a 'content' theory of 'memory' at the ontological level.

RA can also be understood in terms of how it relates to other work in heuristic discovery, memory organization, language processing and machine learning. Table 6.1 contains a guide to chapter 6.

Section	On First reading	Description
1	read	Develops a framework for KR. Defines several terms. Table 6.2 contains a synopsis.
2	read	Fits RA into the framework, and summarizes RA's contributions.
3	skim	Discusses work related to RA along several topical areas.
4	read	Summary.

Table 6.1: Guide to Chapter 6

6.1 A Framework for Knowledge Representational Theories

Research schemas and RA are best understood as an 'ontological' level theory, in particular as a 'content' theory of 'memory' at the ontological level. For most people, the above sentence is pure gibberish: almost all the crucial terms in the sentence, while occasionally used in the literature, are too vague to make any real sense. In this essay, I will develop a framework for classifying and understanding AI theories that normally go under the heading 'knowledge representation.'

Most terms and ideas used in the field of knowledge representation are somewhat ill-defined, but are often used successfully within particular research communities that share similar intuitions. However, these terms often meet with utter incomprehension across communities[1]. In fact, the term 'knowledge representation' itself is used differently in relation to different pieces of work: while Schank's CDs [Schank 73] and Brachman's Klone [Brachman & Schmolze 85] might both be lumped together as techniques for representing knowledge, somehow CDs specify *what* knowledge to represent, whereas Klone specifies *how* to represent your chosen piece of knowledge. Further, most researchers probably don't agree that there is a distinction between knowledge representation and memory organization. In a survey published in the SIGART special issue on knowledge representation, Brachman and Smith noted that there was no consensus on any of the substantive issues pertaining to the field of knowledge representation [Brachman & Smith 80].

In developing a framework for classifying representational theories, it is at first tempting to abandon all the current terms and invent an entirely new set. However, you quickly realize that the existing terms, in spite of their vagueness and ambiguity, reflect some strong and important intuitions about knowledge. Hence I will retain most of the current terms, but I'll clarify them with examples. I will use these terms as if they are my own even if they are traceable to a specific author since my use of the terms is usually

[1] I can personally testify to that. My mention of the term 'episodic memory' has, on more than one occasion, elicited the question "Why don't you just use temporal logic?"

slightly different from their original definition. However, I will indicate where the terms came from, and how my use and the original definitions diverge.

The next several subsections introduce the ideas used to classify representational theories. Each idea is illustrated with examples. Section 6.2 explains the contributions of this work in terms of the framework discussed in this section.

6.1.1 Ontological and Epistemological Levels

While a large body of research contributions in AI go under the general title 'knowledge representation,' one can distinguish between two radically different kinds of contributions: *ontological* level contributions and *epistemological* level contributions:

- A theory (of intelligent behavior) at the *ontological* level specifies (identifies, surmises, vaguely indicates) the kind of knowledge that is required to actualize some cognitive behavior.
- A theory at the *epistemological* level proposes how a computer system (alternately, any intelligent agent) might use this knowledge to realize this behavior under several pragmatic concerns[2].

Thus, an ontological theory specifies *what* kind knowledge is required to elicit what sorts of behaviors, whereas an epistemological theory specifies *how* to realize that behavior in a computer system.[3]

[2]Particularly efficiency, but several other concerns and meta-concerns are also possible: use of the knowledge in a logically correct manner; the ease with which an external observer can prove that the knowledge is used in a logically correct manner, the ease with which a programmer can encode the knowledge, etc.

[3]The distinction between ontological and epistemological levels is strongly motivated by Newell's distinction between knowledge and symbol levels [Newell 82]. In Newell's characterization, the knowledge level pertains to the "knowledge required to solve a problem" and the symbol level pertains to the "processing required to bring the knowledge to bear in real time and real space" (p. 117). This definition thus excludes any processing notion from the knowledge level. However, as we shall see, several theories that propose "the knowledge required to solve a problem" propose "process" knowledge that is naturally described at the knowledge level rather than at the symbol level. Beside this difference, there are two other reasons why I chose to coin new terms rather than use Newell's terms: (1) Newell originally introduced the idea of a knowledge-level as a means of describing AI systems rather than AI theories; in particular, Newell's definition of knowledge level hinges upon the 'principle of rationality' and 'deductive closure.' Neither of these are implied in my definition of the ontological level. (2) Newell's terms suggest that somehow knowledge level is the more substantive one, and the symbol level is mere implementation detail. My terms are more neutral and imply no value judgment.

David Marr distinguished between three levels of description: the computational, the algorithmic and the implementational level [Marr 82]. Ontological level maps rather well on to Marr's computational level and the epistemological level maps on to Marr's algorithmic level. I have decided not to use Marr's terminology since Marr's levels are typically not used in relation to representational theories. Further, the intent behind the idea of the term 'ontological level' is precisely to separate the computational notions from a theory. Hence, calling it the 'computational level' defeats the purpose.

Let's consider a couple of examples: Roger Schank's conceptual dependency theory is a clear example of an ontological theory [Schank 73]. By noticing that people can remember the 'meaning' (of a sentence) while forgetting the specific words used as well as the syntactic structure of the sentence, Schank theorized that sentence understanding should involve a knowledge of the 'conceptual' content of the words. Stated this way, Schank's theory of conceptual dependency (CD) is an ontological theory since it specifies the kind of knowledge required to actualize the behavior of sentence understanding. The original description of CDs did not include an epistemological component at all: i.e., Schank did not specify how this knowledge of 'conceptual' primitives is brought to bear during sentence understanding.

In contrast, Ross Quillian's work on semantic nets is primarily an epistemological theory [Quillian 67]. Quillian proposed a structuring mechanism to represent the 'semantics' of English words. In this model, Quillian attempted to represent the meanings of words by relating them to other words (as in a dictionary) in memory through labelled links. This model derived its power from an intersection search procedure for inference. For example, when given the two words CRY and COMFORT, this procedure would generate the following inference:

> Intersect: SAD
> CRY IS AMONG OTHER THINGS TO MAKE A SAD SOUND.
> TO COMFORT CAN BE TO MAKE SOMETHING LESS SAD.

Quillian's theory of semantic nets does not propose what knowledge to encode in the semantic net to realize any particular behavior. Instead, it proposes a structuring mechanism to encode whatever knowledge you choose to encode. Hence, semantic nets are most usefully understood as an epistemological level theory[4].

In summary, AI theories that normally go under the heading 'knowledge representation' may be one of two kinds: ontological theories specify the kind of knowledge required to actualize particular cognitive behaviors; epistemological theories, in contrast, specify structuring and processing techniques to use this knowledge in actualizing the behavior. However, as we shall see in the next subsection, the situation is not quite so simple.

Here the agreement ends. Here the schism begins among those who have been looking for the answers to the puzzle.

—Thor Heyerdahl, The RA Expeditions.

The term 'epistemology' as I have used it is consistent with Brachman's use of the term to refer to "knowledge structuring" [Brachman 79]; my use of the term 'ontology' is consistent with Lenat and Guha's use of the term to refer to the knowledge commitments of CYC [Lenat & Guha 89].

[4] BTW Quillian's work did include one important ontological level component: the distinction between types and tokens. He identified that the world consisted of two distinct kinds of existents: types that are general categories of objects, and tokens that are specific instances. He noted that, in general, one needs to distinguish between these two kinds of existents.

6.1.2 Content-Theories

As we saw above, ontological theories are concerned with specifying the kind of knowledge required to actualize a behavior. Within the class of ontological theories, we can distinguish between two different kinds:

- *Content-independent* theories specify the generic kind of knowledge required to attain a behavior.
- *Content-dependent* theories (henceforth referred to simply as *content* theories) go further and specify all forms in which this generic knowledge appears in the world. As such, a content theory is meant not only as a theory pertaining to the cognitive behavior, but in some sense, also a theory of the world which the cognitive agent inhabits.

I will first give some examples of both kinds of theories, and then make some general observations about content-theories[5]. Some examples of content theories include Fillmore's work on cases [Fillmore 68], Schank's work on CDs [Schank 73], and Lehnert's work on plot-units [Lehnert 81], summarized below:

- Charles Fillmore claimed that the knowledge of the case structure of sentences was important in sentence understanding (a content-independent ontological claim). He went further to identify a comprehensive set of specific cases for which he claimed some generality, thus making it a content-theory.
- Roger Schank claimed that the knowledge of the conceptual content of words was important in sentence understanding (a content-independent ontological claim). He went further to propose a set of thirteen conceptual primitives for which he claimed generality, thus making CD a content-theory.
- Wendy Lehnert claimed that the knowledge of the affect states of the characters in a narrative was important for narrative summarization (a content-independent ontological claim). She went further to propose a specific set of affect states (and relationships among them) for which she claimed generality, thus making plot-units a content-theory.

In contrast to the content theories considered above, Ed Shortliffe's pioneering work on Mycin embodies a content-independent ontological theory (see [Davis et al 77]). The ontological level contribution of Mycin may be stated as follows: "Expertise in a domain involves considerable knowledge of the domain. This knowledge consists of general

[5]The term *content-theory* has recently started appearing in the CBR literature, see for example [Schank et al 90]. However, at the time of this writing, I do not know of any place where this term has been defined.

semantic knowledge pertaining to the domain."[6]. To demonstrate his claim, Shortliffe built Mycin as an expert system in the domain of medical diagnosis. While Mycin used specific medical notions (predicate-functions, attributes, and objects) to operate, these notions were not meant as part of a theory of expertise in general or even across other medical domains. Thus Mycin did not embody a content theory involving ontological primitives.

To recap, a content-independent ontological theory specifies the generic kinds of knowledge required to actualize some cognitive behavior, whereas a content-theory goes further to identify the particular instances of that knowledge in the world: hence a content theory is not only a theory about a cognitive behavior[7], but is also meant as a theory of the world. The following are some general observations about content-theories:

- Content theories are usually incomplete. In attempting to taxonomize the world, a content theory (typically) does not cover all aspects of the world that it is trying to model. For any given content theory, you might usually be able to come up with new data that the theory does not cover. Thus you could come up with sentences whose conceptual content is not covered by the CD primitives; you could come up with sentences whose cases are not covered by Fillmore's cases; you could come up with stories whose affect content is not covered by Lehnert's affect states.

- Content theories are usually vague about the criteria on which their content primitives are chosen. In CD theory, why is abstract transfer a primitive, but writing a dissertation is not? In Plot-units, why is motivation an affect state link, but inspiration is not? Questions of criteriality are almost never satisfactorily answered by content theories[8].

- A theory is a content-theory only if some generality is claimed across domains. Suppose you build a natural language understanding system to understand newspaper stories in the domain of terrorism. In building such a system, you may introduce notions such as bombing, extortion, ransom notes, and so on. Do these notions constitute a content-theory? Probably not. However, if you introduce notions such as motive, intentional action, goal subsumption etc., and propose a

[6] Btw Contrast this claim with those of Larry Hunter's IVY system [Hunter 88] and Ray Bareiss's Protos system [Bareiss 89]. Like Mycin, both IVY and Protos deal with diagnostic tasks in medical domains. Hunter models expertise using 'paradigm' cases, and Bareiss models expertise using 'prototypical' cases. Their ontological claims might be stated as follows: Expertise in a domain not only involves general semantic knowledge but also specific episodes from one's experience.

[7] When I talk about a theory of cognitive behavior, I do not imply that the theory is a cognitive theory (i.e., it is meant to be cognitively valid).

[8] Wilensky's Kodiak is motivated by this concern for criteriality: Wilensky argues that there is no apparent reason why ATRANS (in CD theory) is a primitive but buying and selling are not. Wilensky's Kodiak makes no commitment to any particular knowledge, but provides a general knowledge structuring mechanism. This makes Kodiak an epistemological, rather than an ontological, theory [Wilensky 87].

theory of story understanding in terms of these notions, your theory will qualify as a content theory for story understanding.

- As you may infer from the last point above, a content theory usually introduces notions of a higher level of abstraction than might be apparent in the phenomena they address. For example, Schank's CDs, Fillmore's cases, and Lehnert's affect states all introduce a set of notions (e.g., abstract-transfer, instrumental case, positive-affect-state) that are not present at the level of words, sentences, and stories. Thus content theories involve a mapping process to translate the surface phenomenon into the concepts proposed by the theory. This mapping is usually not unique, a constant source of criticism against content theories. A notorious feature of CDs is that there is usually no unique translation of a sentence into a CD representation. Similarly, there is usually no unique translation of a story into plot-units.

Despite these problems, content theories have enjoyed considerable success, and have enabled us to understand the notion of knowledge and what forms it might take.

In summary, within the class of ontological theories, one can distinguish between content theories and content-independent theories. In addition to specifying the generic form of knowledge involved in actualizing some cognitive behavior, content theories also attempt to propose a theory of the 'world' in which the cognitive behavior is realized. This is usually done by taxonomizing the various forms in which the generic knowledge might appear in the world. Despite several serious problems in formulation, content-theories continue to be popular, and have contributed considerably to our understanding of the role of knowledge in intelligent behavior[9].

The diffusionists never lacked argument, but they always lacked proof. Therefore, said the isolationists, the oceans had not been crossed.

—Thor Heyerdahl, The RA Expeditions.

6.1.3 The Role of Processing Issues

The ontological theories considered so far were all theories of knowledge without particular commitment to the processes that will use the knowledge. Thus we can talk of a CD expression as standing for the meaning of some sentence, of a case frame as standing for the case-structure of a sentence, or even of a (Mycin) diagnostic rule as standing for some association between a symptom and a disease[10]. The ontological theory is quite

[9] Btw All the theories not considered here are also 'content' theories as they are quite happy at being left alone!

[10] Btw With rules, one has to be careful in distinguishing between the 'knowledge' encoded in a rule and the process implemented by a rule. For example, a rule like, if symptom = ecoli, then diagnosis =

decoupled from the processes that construct or use the knowledge that is specified by these theories[11]. Let me now consider a couple of ontological theories that are strongly tied to some abstract processing notions; these theories, while specifying the knowledge required to realize some behavior, also posit certain abstract processes that will use the knowledge. These abstract processes should be distinguished from the actual algorithms that implement them at the epistemological level.

During the 1920s and '30s, the leading paradigm for memory research in Europe was the Ebbinghaus paradigm that involved presenting subjects with non-sense syllables and testing their rate of learning, retention, recall and so on (see [Gardner 85]). Rejecting this prevailing tradition, an English psychologist, Frederic Bartlett, studied how laboratory subjects understood 'exotic' short stories[12]. Based on these studies, Bartlett theorized that people used structured sets of stereotypical situations in understanding unfamiliar situations [Bartlett 32]. He found that his subjects often incorporated detail that was not present in the original stories, and often omitted unexpected detail that was in fact present in the original stories. Bartlett wrote[13]:

> Remembering is not the re-excitation of innumerable fixed, lifeless, and fragmentary traces. It is an imaginative reconstruction, or construction, built out of the relation of our attitude towards a whole active mass of past experience (p. 213).

Building on Bartlett's work, Marvin Minsky developed his well-known theory of frames [Minsky 75]. Both Bartlett's theory of schemas and Minsky's theory of frames may be stated at the ontological level as follows: understanding requires the knowledge of the structure of a situation to be brought together in toto, rather than in fragments. The point to note is that although this theory (as I have stated it here) does not explicitly specify any particular details on how this knowledge is used, it does make an implicit commitment to a top-down, expectation-driven understanding process. If we disallow all processing notions from the ontological level (as Newell does in his definition of the knowledge-level), Minsky's theory of frames collapses as an ontological theory. For example, Minsky's claim that understanding involves the knowledge of default values for referents left unspecified in a context — an ontological level claim — cannot be stated

bacterial-infection specifies the association between ecoli and bacterial-infection, whereas a rule like, if x = 20, then goto abc is hard to describe at the ontological level. Lenat claims that often knowledge engineers do nothing but assembly language programming with production rules [Lenat & Guha 89]

[11]Hence, my conception of the ontological level, so far, is in agreement with Newell's definition of the 'knowledge-level'.

[12]Btw Bartlett used an experimental technique called the 'Method of Serial Reproduction' based on the parlor game 'Russian Scandal.' In this game, a message is passed around by a set of players to see how the message gets massaged as it gets around. Bartlett used 'exotic' short stories as messages.

[13]I am spending some time to describe Bartlett's work here, since in our later discussion on episodic memory, we will see how similar Bartlett's statements (in 1932) are to recent claims about memory processes, as in [Kolodner 84]

at the ontological level without positing an implicit process that will override the default assignment if the context does specify an explicit value for the referent[14,15].

In passing, I note that the frame theory as proposed by Minsky was quite vague at the epistemological level: while Minsky did specify that a frame might be a data structure that includes a structured set of entities, he was quite vague about how a frame is selected from among many possible ones[16], or how to choose the right value for a referent if the context included multiple values. This has been a constant source of complaint against the frame theory[17,18].

Minsky's frame theory is one example of an ontological theory that subsumes some processing notions. Let me give another example to illustrate how processing notions might be integral to an ontological theory. In their book *Scripts, Plans, Goals and Understanding*, Schank and Abelsen considered the following narrative:

> John was hungry. He reached for the yellow pages.

Most of us understand the causal connection between the above two sentences. What is the knowledge required to achieve this understanding? Schank and Abelson argued that narrative understanding requires a knowledge of plans [Schank & Abelson 77]. Unlike say, conceptual dependency (CD) primitives, plans are not static but dynamic entities. Hence, a theory that posits plan knowledge for narrative understanding subsumes some processing notions on how the plan is to be used. The processing involved in understanding the above narrative can be stated purely at the ontological level as follows: the

[14] Btw Charniak used 'demons' to realize such processes [Charniak 72]. The activation of demons, their life-span, their control structure etc are properly seen as epistemological level concerns.

[15] Btw Bobrow and Winograd's KRL is an epistemological level instantiation of the frame theory [Bobrow & Winograd 77]. KRL provided an elaborate control mechanism — a 'priority-ordered multi-process agenda', no less! — to control the processes one might attach to frames. That should highlight the distinction between processing notions at the ontological and processing notions at the epistemological levels.

[16] Btw The *frame-activation* problem, i.e., the general problem of choosing the right frame from among many (also called the *script-activation* problem as it applies to scripts) has remained quite intractable at the epistemological level. A recent attempt at solving this problem has been to use *connectionist* techniques [Sumida & Dyer 89].

[17] For example, Hayes writes: "Minsky introduced the theory of 'frames' to unify and denote a loose collection of related ideas on knowledge representation: a collection which, since the publication of his paper, has become even looser" [Hayes 79, p. 288]. Brachman and Levesque write: "This problem of general lack of rigor and vagueness has unfortunately followed many of us who pursued the frame ideas, as if the topic itself demanded a certain informal style of research" [Brachman-Levesque 85, p. 245].

[18] Btw It is interesting to note that, in a well-known paper criticizing the 'frame movement,' Pat Hayes misinterprets the frame theory as an epistemological theory. After briefly acknowledging that, at one level, frames may be seen as a thesis about what sort of things a program ought to know, Hayes argues how all frame ideas can be represented in predicate logic with lot more precision. He goes on to provide a predicate calculus representation for some of the 'ontological' claims of the frame theory (such as default values). In conclusion, he writes: "... there are no new insights to be had there [in the frame movement]: no new processes of reasoning, no advance in expressive power" [Hayes 79, p. 294].

first sentence implies a goal state for John, *satisfy-hunger*, and also triggers an expectation that he will execute some plan to fulfill this goal. The second sentence is then recognizable as satisfying the precondition for a plan to find a restaurant where he could eat. Note that the above process level explanation is quite independent of any of the epistemological concerns such as how the plans are represented, accessed, or instantiated. Hence, the theory that narrative understanding requires a knowledge of plans is an ontological level theory even though it is strongly coupled to some (abstract) processing notions[19].

In summary, an ontological theory specifies the knowledge required to actualize a cognitive behavior; often, such a theory of knowledge posits — either explicitly or implicitly — abstract processes that are describable at the ontological level. In contrast, an epistemological theory specifies how this knowledge might be used to obtain the behavior in a computer system under several pragmatic concerns. This specification can take the form of knowledge structuring techniques, and algorithms to operate on these knowledge structures.

6.1.4 Knowledge vs Memory

The term *memory* has been used in computer science for a long time to denote a store of information. Thus we have core and bubble memories, real, virtual and stored-program memory, and so on and on. The term has been used loosely within AI to denote any database of knowledge as in "the system's memory consists of ..." In this subsection, I will discuss a new and emerging use of the term *memory* and distinguish it from the term *knowledge-base*; correspondingly, the phrase *memory organization* is distinguished from the phrase *knowledge representation*. This usage of the term 'memory' is traceable to Endel Tulving [Tulving 72] where he contrasted knowledge as something you *know* from memory as something you *remember*[20]. Within AI, this distinction is most apparent in the work of Roger Schank in his book *Dynamic Memory*.

Before discussing this notion of memory, let me first describe a certain fairly common conception of knowledge in AI systems which I will refer to as the *knowledge-base view*. In this view, an AI system consists of two essentially independent components: a performance program plus a knowledge-base. The knowledge-base, in turn, contains two independent components: a set of data structures that represent the knowledge in whatever form (semantic nets, predicate expressions, rules, schemas etc) plus a general inference mechanism (a marker passer, a theorem prover, etc). The performance pro-

[19]Under Newell's characterization, this theory should be classified as a symbol-level theory since it involves processing notions; however, these processing notions are quite different from Newell's definition of the symbol level as the "processing required to bring the knowledge to bear in real time and real space."

[20]Tulving himself traces this distinction to a monograph by Reiff and Schreerer [Reiff & Schreerer 59] which makes a distinction between *remembrances* and *memoria*.

gram interacts with the knowledge-base by formulating queries such as "Is John a dog?" or "What is the connection between John and Fido?"; the knowledge-base answers these queries as "No, John is not a dog" or "I couldn't prove that John is a dog, therefore John is probably not a dog" or "The connection between John and Fido is that John *owns* Fido." These answers are either declaratively present in the knowledge-base or are inferred by the inference procedure. The performance program operates, broadly, as follows: it is given a problem; it accesses the knowledge from the knowledge-base (the phrase 'Long-term Memory' is occasionally used) and brings it into a working memory. The problem is solved; the working memory is destroyed; and the world is, once again, restored to its original state[21]. While this view is somewhat of a caricature, this kind of interaction is the ultimate dream for some researchers, particularly those who are partial to a declarative logic based formalism.

In contrast, human memory is sensitive to how it is accessed. For example, in a series of experiments with laboratory subjects, Elizabeth Loftus showed that memory access can retroactively change the contents of the memory. She showed a group of subjects a slide of a road intersection that has a STOP sign. After a small time interval, the subjects are asked a question about the YIELD sign, and then asked to describe the scene. As expected, a significant number of described the scene with a YIELD sign at the intersection [Loftus 75].

Further, human memory is reconstructive (see the quote from Bartlett, page 200). When asked a question such as "Where were you on the night of the twenty-fourth?", you have to reconstruct the events of the twenty-fourth and decide where you might have been. This is not knowledge that is declaratively stored and retrieved.

Next, human memory exhibits spontaneous remindings. Reading this dissertation may remind you of the time you saw this play whose plot you couldn't discern. Writing about the knowledge-base view reminded me of Lewis Carroll's Jabberwocky. Broadly, remindings may be of two kinds: semantic or structural. In the semantic case, one thing, say A, reminds you of another thing, B, because they encapsulate a common component. Thus one Star Trek episode reminds you of another because Spock's father Sarek appears in both. In the structural case, A reminds you of B, because in some abstract ways, A and B are similar: for example, even though there are no common components between our knowledge-base view and Jabberwocky, there is a deep structural similarity between the two. Reminding (also known as *indexing* as it applies to case-based reasoning systems) is one of the several major issues that separates research in knowledge representation and research in memory organization.

For the moment, let's focus on the problem of reminding: i.e, the problem of accessing or retrieving some piece of knowledge from memory. Is this an ontological or an epistemological level problem? I will argue that what distinguishes knowledge and

[21] Btw In other words, the world is first brillig and slithy toves; after snicker-snacking the whiffling jabberwock, the world is once again brillig and slithy toves.

memory is whether the access problem is treated as an ontological or an epistemological problem.

The simplest case of knowledge access occurs in any Lisp system where the Lisp reader is given the *pname* (i.e., the string or the 'print-name' by which the symbol is known) and asked to access the data structure corresponding to the symbol. To achieve this mapping between the pname and the data structure efficiently, Lisp systems have traditionally used a hash table as an indexing mechanism. Note that a hash table does not specify any new knowledge, only an efficient mechanism that trades off space for time. As such, a hash table does not embody an ontological theory for knowledge access.

In a surprising number of AI systems, the Lisp symbol-table is the only mechanism needed for knowledge access, either because knowledge is unstructured or because the structured knowledge has a unique name for identification. In such systems, the knowledge access problem is an epistemological level problem.

Consider the access problem for frames represented using KRL[22]. KRL recognizes that knowledge access is not a simple case of presenting a knowledge-base with a pname, and provides extensive facilities for retrieving the right frame to understand a situation. The user can specify arbitrarily complex *description-matchers* as a probe to retrieve a frame from memory. However, KRL does not specify what sorts of knowledge should go into the description matchers to access what kinds of knowledge. KRL only provides an efficient implementation of a matching process for any user specified probe, thereby implementing an epistemological, rather than an ontological, theory of knowledge access.

When you move from knowledge representation to what might be called memory organization, the knowledge access problem becomes an ontological level concern. Most people, it seems, are reminded of Shakespeare's *Romeo and Juliet* when they see the Broadway play *The Westside Story*. Superficially, there is little in common between the two stories: in particular, the names of the characters, their families, their ages etc are not the same in the two stories. As a knowledge access problem, the question is "What kind of knowledge about Westside story accesses the corresponding knowledge about Romeo and Juliet?" Wendy Lehnert, using her plot-unit analysis, showed that both Romeo and Juliet and Westside Story have the same affect structure [Lehnert 81]. While Lehnert's work was not phrased as a theory of memory access, we can imagine a memory for stories organized in terms of their affect structures and plots, and use affect state maps as probes to access similar stories from memory. Such a theory of story memory is ontological since it identifies the kind of knowledge that partakes in the knowledge access problem.

Earlier we saw that ontological theories can subsume some abstract process level notions. This is particularly true of theories in memory. Suppose I ask you "Who is the most famous person whom you've met?," you would probably say something like: "Let's

[22]KRL is a sophisticated frame representation language. See [Bobrow & Winograd 77].

see... have I met any politicians? I met the mayor once in a fundraising campaign; have I met any basketball players? Well, it looked like Larry Bird once waved in my general direction..." The point is that answers to these questions are not declaratively stored in memory, but may have to be inferred by generating a series of probes in order to ask yourself a set of more concrete questions. These kinds of phenomena were considered by Janet Kolodner in her CYRUS system that attempts to model the autobiographical memory of ex-US-secretary-of-state Cyrus Vance [Kolodner 84]. Kolodner proposed that memory retrieval requires a reconstruction of the past: in order to retrieve an item in memory, one has to progressively narrow the context in which the item is likely to be found. Note that these process level claims about memory are ontological in nature since they have nothing to do with pragmatic concerns such as space or time efficiency[23].

Having differentiated between knowledge-bases and memories, we can now differentiate between knowledge representation and memory organization. Knowledge Representation is an epistemological concern; a representation is a structuring mechanism that is motivated by the concern for elegance, efficient implementation of algorithms, requirements for storage space, powerful, clean and correct inference mechanisms and so on. Similarly, memory organization is an epistemological level concern: a particular organization of memory is motivated by the need to minimize redundancy, and implement the memory processes (suggested by the ontological theory) efficiently. In recent years, memory organization languages have started appearing [Cognitive Systems 90].

In summary, the term 'memory' has acquired a new meaning in recent times. This meaning is best understood as a distinction from what I called the 'knowledge-base' view of knowledge. In a memory, among other things, the problem of accessing the right knowledge itself involves knowledge, and hence is an ontological level problem. Further, theories about memory are strongly coupled to ontological level process notions. The goal of research in memory is to identify the various distinct ontological classes of knowledge, their relationships, and the processes subsumed by them.

6.1.5 Semantic and Episodic Memories

The distinction between a knowledge-base and a memory can be traced to the distinction between semantic and episodic knowledge. In March 1971, a conference on *Organizational Processes in Memory* was held at the University of Pittsburgh under the sponsorship of the Office of Naval Research. Two kinds of psychologists participated in this conference: the traditional cognitive psychologists and the new brand of psychologists who used the information processing approach[24]. In organizing the conference proceedings into a book, one of the editors, Endel Tulving, noted that there was a radi-

[23]Kolodner proposes a fast parallel implementation of an indexing algorithm in [Kolodner 88b]. This is properly seen as an epistemological level contribution.

[24]The later category included Rumelhart, Lindsay, Norman, Kintsh, Collins, Quillian, Greeno and others.

cal shift in the meaning of the term 'memory' from the first set of papers to the second. This prompted him to write a paper called *Episodic and Semantic Memory* that was also included in the volume [Tulving 72].

Tulving noted that the term as used by the psychologists involved issues such as retention, recall etc, concepts that we generally associate with information that we say we *remember*. In contrast, the term as used by the information-processing psychologists involved issues having to do with *semantic* information, i.e., information that we generally say we *know*. Tulving coined a new term, *episodic* memory and contrasted it from *semantic* memory. For example, when you say, "Last year, while on summer vacation, I met a retired sea captain," you are referring to an event from your past that you remember. In contrast, when you say, "The value of π is 3.14," you are retrieving a piece of knowledge that you know. In particular, this knowledge is timeless — i.e., it has no temporal reference. Tulving proposed two memory systems to deal with these two kinds of knowledge.

While this distinction, to date, has not been formalized, and the controversy is as yet undecided, this distinction has had a tremendous impact in psychology. In the preface of the proceedings of the second conference on memory organizational processes, the editor, Richard Puff, notes that "The fact that nearly every chapter here makes some descriptive use of these terms [semantic and episodic memory] reflects how enormously useful these terms have been for various kinds of tasks and classes of evidence... There is thus an increasing tendency to talk about semantic and episodic information rather than memory systems." [Puff 77, p. 14, p. 16][25,26].

In AI, the distinction between semantic and episodic memories has met with mixed review: one group of researchers, notably Schank and his colleagues, have embraced the notion of episodic knowledge to the exclusion of semantic knowledge[27]. Several others whose systems have never had to deal with the notion of episodes have completely ignored this distinction. Others have used the notions of episodes, but have not made any particular distinction between episodic knowledge and semantic knowledge: in fact, for most purposes, they can be represented the same way at the epistemological level. For example, "The value of π is 3.14" and "John kissed Mary" can be both written as predicate expressions: (value π 3.14) and (kissed John Mary).

[25] Btw Based on brain studies with accident victims, there also seems to be some evidence to indicate that these two kinds of knowledge may in fact be stored differently at the neurological level: semantic information is less sensitive to interference than episodic information (see [Tulving 83]).

[26] Btw At least one language, Spanish, has this distinction built into the language. The English verb 'to be' stands for two conceptually different relations: When I say, "I am handsome," I refer to a timeless characteristic, whereas when I say "I am fine," I refer to a temporally based state of being. In Spanish, the verb 'to be' has two forms 'ser' and 'estar' to differentiate between these two cases; you would say "Yo soy guapo", i.e., I am (ser) handsome, and "Yo estoy bien", i.e., I am (estar) fine.

[27] Schank argues that all knowledge is gained through personal experience or episodes, hence there is no such thing as semantic knowledge [Schank 75b].

This distinction is properly seen as an ontological level distinction. Suppose we have a knowledge-base into which we can store assertions and later query the system on what it knows. If we give this knowledge base the assertion: (value π 3.14), we have given it some new knowledge. However, giving the knowledge base the same assertion the second time does not add any new knowledge to the knowledge base. Thus semantic knowledge has only one temporal instantiation, which is the same as saying that semantic knowledge has no temporal instantiation. However, the assertion (kissed John Mary) can be presented to the knowledge base any number of times and each is a new piece of knowledge (one more kiss). Thus we can ask the question how many times did John kiss Mary, but not how many times is π 3.14.

Since events can recur any number of times, we can use the knowledge of the event from one occurrence to understand future occurrences of the same event. This statement is best explained in the context of language understanding. Take, for example, the notion of scripts [Schank & Abelson 77]. Scripts are a compilation of the stereotypical actions involved in some activity, the standard example being visits to restaurants. Since restaurant visits can occur any number of times in various stories, one can use a restaurant script to understand various occurrences of this event in a top-down, expectation-driven fashion. Hence, almost all such schema-based theories of language understanding have been in relation to event-based texts or narratives.

In contrast, expository texts (such as this dissertation) are characterized by a preponderance of semantic knowledge. Take that last sentence for example. If you knew that expository texts are characterized by semantic knowledge, there was no new knowledge stated by that sentence; if you didn't, then this knowledge is completely new and has to be understood in a bottom-up, non-expectation-driven way. Thus the distinction between expository and narrative texts stems from the ontological distinction between semantic and episodic knowledge. Episodic knowledge has temporal instantiations, whereas semantic knowledge does not.

In summary, the distinction between semantic and episodic knowledge, while somewhat hazy, recognizes the distinction between knowledge that does not have a temporal reference and knowledge that does. This is an ontological level distinction since the two kinds of knowledge exhibit different properties[28].

In this section, I discussed several broad distinctions among AI theories of knowledge and knowledge representation. The major points of this discussion are summarized in Table 6.2. The next section explains RA's contributions in terms of the ideas introduced above.

[28]The distinction between rule and case-based reasoning can be traced to the distinction between semantic and episodic knowledge. Rules denote timeless semantic association between the 'if' parts and the 'then' parts. In contrast, cases are specific memories for events from one's experience.

- An *ontological* theory specifies the knowledge required to attain some intelligent behavior.
- An *epistemological* theory specifies how some knowledge can be used to achieve a behavior within several kinds of pragmatic concerns.
- A *content-independent* ontological theory specifies the generic kinds of knowledge involved in some intelligent behavior.
- A *content-dependent* ontological theory specifies particular forms in which some generic knowledge appears in the world; as such, content theories are meant not only as a theory of a behavior, but also as a theory of the world.
- Ontological theories can involve abstract processing notions that are integral to the theory. These processes should be contrasted from epistemological level processes which are specific algorithms for realizing an ontological level process.
- A *Memory* may be distinguished from a *Knowledge-Base*. In a memory, the knowledge-access problem is ontological, whereas in a knowledge-base, the knowledge-access problem is epistemological.
- *Semantic* knowledge must be distinguished from *episodic* knowledge. The former does not have a temporal reference, while the latter does. This is an ontological level distinction.

Table 6.2: Synopsis of the Framework

6.2 Fitting RA into the Framework

Just as the bows touched the surface of the water a wide-eyed photographer leaned toward me and said: "What will you say if it goes straight to the bottom now?" There was no time to answer. Ra floated out.

—Thor Heyerdahl, The RA Expeditions.

This Section describes RA in terms of the ideas introduced above. In the first subsection, I discus RA's memory organization and its various functionalities. The learning component, RA II, is discussed separately in Section 6.2.2. The contributions of RA are summarized in Table 6.3.

6.2.1 RA I and Memory Organization

RA is best seen as an ontological theory, in particular, as a content-theory of memory. In this section, I will expand on the above sentence and explain RA's contributions in terms of the classification framework developed above.

This research was motivated by the broad question, "What sorts of knowledge are involved in a researcher's competence in his field?" An obvious answer to the question might be that the researcher has to have a deep knowledge of the research papers in his field. This knowledge — which I called *deep semantic* knowledge — involves the definitions of the various terms, how they relate to each other, the details of the proposed techniques, why they are important, how they are evaluated and so on[29]. While not denying the existence and use of this kind of knowledge, I considered what other kinds of knowledge might be involved by looking at several tasks that researchers typically perform:

- I show that, in addition to the deep-semantic knowledge, there is also another kind of knowledge — what I have called *structural* knowledge — involved. This knowledge suppresses the details, and models the domain in terms of abstract notions such as *problems, techniques,* etc.

- I show that, in addition to the semantic knowledge — what I have called *subject* or S-knowledge — there is also a kind of episodic knowledge involved. I have called this the *evolutionary* or E-knowledge. This knowledge models research papers as events that contribute to the evolution of the field, and shows that there is strong interplay between the two kinds of knowledge.

[29]In fact, some of the so-called *knowledge-based* approaches to Information Retrieval attempt to use precisely this kind of knowledge to index and retrieve research papers (see [Lewis et al 89] for a review.

Let's now consider the support for the above claims. The claim that a researcher ought to see the domain at the structural level obtains from the fact that researchers see analogical connections among research papers that are in fact very different at the deep-semantic level[30]. Secondly, the fact that researchers can discern, list and classify the methods that are used in research also suggests that there should be a level of abstraction that cuts across the deep semantic knowledge. These methods, in order to be general, have to involve higher level notions not apparent at the deep semantic level[31].

The second claim is supported by the observation that researchers can easily provide a historical account of their field. A historical account is not just a list of papers in the field sorted in chronological order, but also an account of how the subject knowledge introduced by the papers relate to each other. Such historical accounts appear in practically every paper to show how the contribution of that paper relates to past papers in the field. Further, historical accounts are also common in various surveys of the field[32,33].

The statements above are all fairly general statements about the kinds of knowledge involved when we consider the various tasks that we attribute to researchers. These statements do not specify what kinds of structural abstraction exist, what is the nature of the research episodes, nor how the S- and E-knowledge are related to each other. This research proposes a small set of structural abstractions, describes the nature of research episodes and the relationship between the subject and evolutionary knowledge. These are discussed below.

Structural Primitives: The structural primitives used in RA include four types and nine relations. The two main types, problem and technique stand respectively for the object and result of a research inquiry. These two notions are minimally required

[30]See for example, Prieditis et al's summary of Machine Learning conference in which summarizes a set of papers in the conference as using 'hybrid methods' [Prieditis et al 87]. See Lehnert's AAAI survey talk on Natural language processing in which she uses abstractions such as weak and strong methods [Lehnert 88a]. Finally the field itself has several terms that denote very high-level abstractions such as the *new-term* problem, the *credit-assignment* problem, the *indexing* problem and so on: two pieces of work may have little in common at the deep semantic level, but may still be different instantiations of, say, the new-term problem — e.g., Lenat's AM [Lenat 76] and Utgoff's STABB [Utgoff 84] have little in common, but are both concerned with the new-term problem.

[31]Several researchers have been concerned with heuristic methods in scientific theory formation. Some well known examples include [Hadamard 54], [Polya 57],[Lakatos 76], [Lenat 76] [Michener 77] and [Langley et al 83].

[32]See for example, Gardner's *Mind's New Science* for an extensive historical account of the various fields that constitute 'Cognitive Science' [Gardner 85]. See Lehnert's survey of NLP in terms of historical trends [Lehnert 88a]. See Groner et al's historical account of the notion of 'heuristics' [Groner et al 83]. Also Lakoff's *Women, fire, and dangerous things* contains extensive historical accounts of prototype theory [Lakoff 87].

[33]BIW Historical surveys may be distinguished from classificatory surveys that provide a semantic classification of the work in the field. Some examples include David Waltz's survey of natural language [Waltz 82], Carbonell et al's survey of Machine Learning [Carbonell et al 83] and Beverly Woolf's survey of intelligent tutorial systems [Woolf 88].

in any conception of of a purposeful activity, i.e., we need a notion of a purpose and a notion of an activity to fulfill the purpose! The other two types, property and concept were chosen to stand for higher and lower order objects with respect to a research field (see Section 2.2.2 for an extensive discussion). The types determine the set of relations chosen: the three epistemological relations (dominates, instantiates and encapsulates) are common, well-known taxonomic relations. Solves and entails are used to state the two possible relations among problems and techniques. Exhibits and involves are used to relate properties and concepts to other kinds of objects. These objects and relations form the core vocabulary that I believe ought to be part of any similar conception of research fields. This is claimed to be a *content* theory in that it is, in some sense, not only a theory of a set of cognitive behavior, but also a theory of the particular existents in the world; further, the types and relations are not tied to any specific ideas in machine learning[34]. While I have tried hard to answer the question of criteriality — i.e., why this specific set of types and relations, some of the other problems of content theories (discussed in Section 6.1.2) apply to this theory as well:

- The problem of coverage: One could always come up with a paper that cannot be described in terms of the abstractions defined by this theory. Interestingly, the problem of criteriality and the problem of coverage seem to be mutually exclusive: as you include more and more primitives (for example, for an AI field, one could introduce primitives to stand for domains, tasks and so on), you can achieve better coverage, but it is harder to define your criteria for the choice of primitives. I have attempted to keep the set of primitives minimal thereby erring in the direction of coverage. However, this theory does provide a generic relation R to stand for any domain-dependent relation.

- The problem of unique translation: Like other content theories discussed in Section 6.1.2 above, the types used in RA (problem, technique etc) are at a higher level of abstraction than is apparent at the textual level of a paper. As such, there is an implicit translation process involved in going from the text of a paper to its representation in terms of problems and techniques. This translation is not unique: a given paper may be translated into our representation in a number of different ways and it is usually not easy to tell what is the problem being addressed by a paper. While I characterized Winston's paper as proposing a technique to solve the concept-learning problem, one could equally well characterize the paper as solving the arch-learning problem (as opposed to the cantilever-beam-learning problem!). However, the existence of multiple translations does not contradict the fact that knowledge at this level of abstraction does exist.

In summary, the structural abstractions used in RA to describe research papers may be seen as a *content* theory of research: the theory shows that, in addition to the deep-

[34]Except the relation *acq*, which is excluded from the core theory.

semantic knowledge, knowledge of the papers at a higher level of abstraction is also involved in several of the tasks attributable to researchers. Further, the theory goes on to provide a taxonomy of types and relations, thereby providing (in some sense) a theory of the particular existents in the world.

Structure of Research Episodes: In this research, I show that conceptualizing the research episodes as research schemas provides considerable explanatory power: if each episode (each paper) is seen as the combination of its immediate research context (i.e., its **ref**) plus the new knowledge added by the paper (i.e., its **def**), we can explain the following phenomena:

- Ability to access the subject knowledge of the field without reference to the papers it came from, e.g., "EBL solves the learning-problem."
- Ability to access the paper that proposed some piece of knowledge, e.g., "version spaces was proposed by [Mitchell 78]."
- Ability to see a paper not as a random collection of semantic information, but as a motivated piece of research, e.g., "Version spaces had a deficiency in that its inductive bias was fixed. In [Utgoff 81], Utgoff proposed a technique to solve this problem."
- Ability to suggest general research directions, e.g., "Maybe you could see how EBL can be used to solve the problem of learning control-knowledge."
- Ability to detect chronological relationships among papers, e.g., "Winston's depth-first technique had an emergent problem of backtracking. Vere proposed a breadth-first technique that partially solved the backtrcking problem." (see [Mitchell 78]).
- Ability to detect analogical relationships among papers, e.g., "Several papers in this conference use hybrid techniques to solve the various emergent problems of EBL." (see [Prieditis et al 87]).

Note that all the above explanations can be stated at the ontological level, without concern for particular representations or pragmatic concerns such as efficiency. For example, we can say that suggesting research directions involves the knowledge of the domain plus the knowledge of general research methods; understanding the motivation behind a particular paper requires a knowledge of the current state of the field *plus* the research contributions of that paper; the ability to find the chronological connections between two papers involves a knowledge of the contribution of the earlier paper and how the contributions of the later paper relates to those of the earlier one. The RA program is simply an obvious, straight-forward implementation of these ontological level explanations. Several of the epistemological level problems were either ignored or finessed. In particular, some important epistemological level concerns are the following: (1) in what

order should the research heuristics be accessed?[35] (2) how many generations into the past should a chronological summary go?[36] and so on.

The theory of research episodes in terms of research schemas is a theory of 'memory' in that the knowledge access problem is treated as an ontological level problem. The theory is motivated by concerns for what kinds of knowledge is involved in accessing what kinds of knowledge in memory. For example, when I say that version-spaces ought to retrieve [Mitchell 78], I specify that the S-knowledge entities ought to access the episodes (E-knowledge) that introduced those entities into the memory; when I say that [Utgoff 81] has a **ref** pointer to a relation defined by [Mitchell 78], I specify that the motivation for Utgoff's work cannot be understood without accessing the knowledge defined by [Mitchell 78].

In summary, the conception of the E-knowledge of a research field in terms of research schemas has considerable explanatory power. Such explanations are ontological level explanations in that they specify what kinds of knowledge are involved in what kinds of behaviors. The RA program is simply a straightforward implementation of these explanations. Further, the theory of research embedded in research schemas is a theory of 'memory' since it treats the knowledge access problem at the ontological level.

6.2.2 RA II and Schema Acquisition

The contributions we discussed in the last subsection were concerned with the various tasks performed by RA using research schemas. In Chapter 4, we considered how one could learn the research schemas within the same memory organizational framework. In some broad ways, the learning strategy used by RA is similar to EBL: RA learns a schema from a single instance; RA's assimilation phase results in the instantiated schema, which may be seen as an explanation; RA's generalization phase generalizes the instantiated schema into a skeletal schema like in EBL systems. However, at the ontological level, RA's learning strategy is significantly different from the EBL strategy. Typically, EBL systems learn by deriving an explanation as to why a given instance belongs to a target concept. This explanation is constructed by using *causal* knowledge. For example, Mitchell et al consider how one could learn the concept of when it is safe to stack on object x on another object y. Their EBG algorithm [Mitchell et al 86] uses causal rules such as the following:

$$\textit{safe-to-stack}(x,y) \Longleftrightarrow \textit{fragile}(x) \vee \textit{lighter}(x,y)$$

In contrast, when we consider learning heuristic methods, it is hard to talk about absolute causality. For example, one of RA's research heuristics is shown below:

[35] Lenat's AM had an involved procedure, called *rippling*, to determine the order in which heuristics are accessed [Lenat 76].

[36] RA goes, arbitrarily, two generations into the past, see Section 3.5.

Major Claims

- In addition to the deep-semantic level, scientific research is also understood at a structural level of abstraction.
- In addition to the subject knowledge, scientific research is also understood in terms of the evolutionary knowledge.
- Heuristic knowledge is learned by relating a new input to the most specific existing knowledge in memory.

Some Perspectives

- This work proposes a taxonomy of structural abstractions, making it a content theory.
- It proposes that the evolutionary knowledge consists of research schemas, i.e., a combination of a paper's immediate context (**ref**) plus the new knowledge added by the paper (**def**).
- It is a theory of memory in that the knowledge access problem is treated at the ontological level.
- It integrates semantic and episodic knowledge into a single memory.
- RA's learning may be seen as a memory-based learning strategy as opposed to EBL's knowledge-based learning strategy.

Table 6.3: Summary of RA's contributions

> If there some problem P1 has a sibling problem P2, and if there is a technique T2 to solve P2, then suggest, "You could try to solve P1 using a technique similar to T2."

If hard pressed, one could give a spirited argument for the reasoning behind this heuristic such as "sibling problems are likely to have a core or an essence that is similar; a technique that solves one problem somehow addresses this core; it might be possible to adapt it slightly to address the core of the other problem." However this explanation is hardly causal[37]. The claim associated with RA's learning strategy may be stated as follows:

- Heuristic knowledge is learned by relating a new input to the most specific existing knowledge in memory.

Note that this is an ontological level claim in that identifies the knowledge that partakes in the learning of heuristic knowledge. In addition, I also proveded an analysis of the learning strategy in terms of basic level categories (Chapter 5). While this analysis is somewhat speculative, it raises several interesting questions about the intuitions behind the ideas of schemas and speculates that when the world is in fact well-structured, maximally associative and discriminative categories may provide a universal bias for generalization.

6.2.3 Why Is This Interesting?

This work is interesting for a number of reasons that are closely intertwined with each other. The following list summarizes some of these reasons:

- Almost all work in memory organization has been concerned with what might be called "episodic" domains, where there is a strong notion of events (e.g., MOPS [Schank 82], IPP [Lebowitz 83] Cyrus [Kolodner 84], Chef [Hammond 89]). In this work, I have taken a "semantic" domain that deals with "timeless" scientific knowledge, and discern the episodic structure of this knowledge. I propose a memory organization that integrates both the semantic knowledge (S-knowledge) and the episodic knowledge (E-knowledge) pertaining to the domain.
- There has been recent controversy between what is called Case-Based Reasoning, i.e., using particular episodes for reasoning, and Rule-Based Reasoning, i.e., using 'semantic' rules for reasoning. RA uses skeletal schemas as both case indices and as if-then rules.

[37]You see this kind of "intuitive" reasons next to most of AM's heuristics as well [Lenat 76]. For example, one of AM's heuristics is "A non-constructive existence conjecture is interesting." Lenat writes: "Thus the unique factorization theorem is judged to be interesting because it merely guarantees that some factoring will be into primes. If you give an algorithm for that factoring, then the theorem actually loses its mystique and (according to this rule) some of its value" (p. 241).

- Even though this work does not provide explicit mechanisms for language processing, it should be of interest to researchers concerned with what kinds of knowledge are involved in understanding natural language. Schema based representations, typically, are the forte of event-based narratives. This work considers a large class of expository texts, models them as episodes, and provides a schema-based representation.

- This work is also interesting from the point of view of its learning scheme. Broadly, this work can be contrasted from EBL learning in that RA's learning strategy might be called "memory-based" learning, whereas EBL strategy might be called "knowledge-based" learning. Further, the analysis of the learning strategy in Chapter 5 raises several interesting questions about knowledge representation and heuristic generalization.

- Finally, unlike most work in AI that proposes some theory of knowledge in order to explain one behavior, this work explains several conceptually different behaviors using a single theory of memory.

In Summary, RA is best understood in terms of its ontological level contributions. In this section, I explained RA's contributions and discussed why these contributions are interesting. Table 6.3 provides a summary of RA's contributions.

6.3 Related Research

This section discusses research that is related to RA along the following topical areas: heuristic discovery, memory organization, language processing, and machine learning. I will provide a short description of the related work and discuss their relation to RA. Since RA does not address the issues of control and analogical reasoning in any significant way, this dissertation does not contain a review of the literature in these areas. For a discussion on control issues in rule based systems, see [Davis & King 77] and [Corkill et al 82]. Also see the discussion on production systems in Chapter 3 of [Barr & Feigenbaum 81]. For a discussion on the representation of control knowledge, see [deKleer et al 77] [Cohen et al 88a] and [Clancey 85]. Ortony's book [Ortony 79] is a good treatment of analogy and analogical reaoning. [Hall 89] is an excellent survey of research in analogical reasoning and contains several pointers into the literature in this area.

6.3.1 Heuristic Discovery

There is a significant amount of work in heuristics and scientific theory formation. Perhaps one of the first to explore the idea was Descartes. He proposed 21 heuristics to direct the mind (regulae ad directionem ingenii) to reduce any problem to algebraic

equations (see [Groner et al 83]). Another early explorer in heuristics, Leibniz, criticized Descartes's heuristics as too vague that basically amounted to the statement, "Take what you have to take, and work the way you have to, and you will get what you are looking for!" (Sume quod debes et operare ut debes, et habebis quod optas). Leibniz devised a system of primitives and a mapping of these primitives onto natural numbers. The ultimate goal of Leibniz's approach was to derive a mapping such that a statement like "All As are Bs" can be verified by simply checking if the number corresponding to B is a factor of the number corresponding to A. For an excellent historical survey of heuristics, see [Groner et al 83]. In recent times, several researchers, both from mathematical and computational backgrounds, have been interested in heuristic approaches to theory formation. Polya [Polya 57] and Hadamard [Hadamard 54], coming from a mathematical tradition, detailed several heuristic approaches to problem solving in mathematics. In her work on developing a conceptual framework for communicating mathematical knowledge, Edwina (Rissland) Michener used three broad categories (or types) of mathematical knowledge: concepts, results, and examples. She used the idea of *duals* to relate the three spaces corresponding to these three types. For example, the dual of a result item consists of examples motivating it, concepts and results needed to state and prove it, and concepts and results derived from it. These ideas were used in a computer-based tutorial system called Grokker [Michener 77] [Michener 78]. Doug Lenat used a heuristic-based approach for machine discovery in his AM system [Lenat 76] which we will review below. Pat Langley, in his Bacon system, used what might be called data-driven heuristics to perceive patterns in experimental data (see [Langley et al 83]).

Even though the goal of this work is not to build a discovery system, the suggestion component of RA has some interesting relationships to AM's approach to heuristic discovery. The AM program was built by Douglas Lenat as part of his doctoral dissertation [Lenat 76] and was later expanded into the Eurisko system[Lenat 83]. In this section, I will first describe these systems in some detail, and then discuss RA's relationship to them.

6.3.1.1 AM

AM is a machine discovery system that models discovery as a heuristic search process. For concreteness, AM operates within the domain of elementary arithmetic. The initial knowledge encoded into the system consists of 100-odd 'pre-numerical' concepts, e.g., concepts such as 'bags' (unordered sets), operations such as compose, relations such as list-equal etc. The following is a frame defining the active concept LIST-EQUAL (taken from [Lenat & Brown 84]):

NAME:	LIST-EQUAL
IS-A:	(PREDICATE FUNCTION OP BINARY-PREDICATE BINARY-FUNCTION BINARY-OP ANYTHING)
...	...

```
RECUR-ALG:    (LAMBDA (X Y)
                      (COND ((OR (ATOM X) (ATOM Y)) (EQ X Y))
                            (T (AND
                                 (LIST-EQUAL (CAR X) (CAR Y))
                                 (LIST-EQUAL (CDR X) (CDR Y))))))
DOMAIN:       (LIST LIST)
RANGE:        TRUTH-VALUE
```

In addition, AM is also provided with a set of 200-odd heuristic rules. Few of AM's heuristics are listed below:

1. If very few examples of X are found,
Then add the following task to the agenda: "Generalize the concept X", for the following reason: "X's are quite rare; a slightly less restrictive concept might be more interesting."
2. To fill in generalizations of concept X,
Take the definition e and replace it by a generalization of e.
If e is a conjunction, then remove a conjunct or generalize a conjunct.
If e is a disjunction, then add a disjunct or generalize a disjunct[38].
3. Any entity X is interesting if it is related (via a rare, interesting relation) to another entity which arose in a very different way and is not obviously tied to X.
4. After working on Operation F, give a slight, ephemeral boost to tasks involving Range(F).

Heuristics such as (1) above suggest tasks to perform; heuristics such as (2) provide syntactic mutations to synthesize new Lisp code from existing code; heuristics such as (3) collectively define the notion of 'interestingness'; finally, heuristics such as (4) deal purely with controlling AM's agenda mechanism.

Starting with its initial base of concepts, AM uses these heuristics to discover new, interesting concepts. AM discovered, by itself, several well-known results in elementary arithmetic, including the unique factorization theorem, Goldbach's conjecture, and prime numbers. Let's quickly consider how AM discovered the concept of equality. AM sought for examples of LIST-EQUAL by picking random lists to see if they are equal. Not surprisingly, few of these lists turned to be equal to each other. So the first heuristic above suggested that, since very few examples of identical lists were found, a more general concept might be more interesting. Now the second heurisic above fires. This heuristic takes the definition of LIST-EQUAL (i.e., the RECUR-ALG slot) and attempts to generalize this: since the COND part of the lisp definition of LIST-EQUAL contains a conjunction, this heuristic generalizes the conjunction in each of the following three ways: removal of the first conjunct, removal of the second conjunt, and replacement of the conjunction by a disjunction. The first of these three generalizations is shown below:

[38]This is a part of a larger heuristic listed in [Lenat 76] (heuristic 89, p. 244).

```
L-E-1: (LAMBDA (X Y)
           (COND (OR (ATOM X) (ATOM Y)) (EQ X Y))
                 (T (L-E-1 (CDR X) (CDR Y))))
```

With the test in the CAR direction removed, all that this function checks for is whether the two input lists become null at the same time; hence it returns true if the two input lists have the same length and false otherwise. Since AM had developed a unary notation for numbers, the same-length function above became the defining fuction for equality.

After an initial burst of discoveries that included prime-numbers and Goldbach's conjecture (among others), AM's performance tapered off. Realizing that the major limitation of AM was its inability to discover new heuristics, Lenat considered this problem in his Eurisko system (described below).

Before I come to discussing the interesting relationships between RA and AM (in Section 6.3.1.4), let me note some of the differences between the the two systems. While AM used heuristics for discovery, AM was not intended as a model of memory. In particular, AM did not consider the relationship between the knowledge of arithmetic (stored in frames) and the heuristic knowledge stored in the heuristic rules. In AM, the knowledge access problem was dealt with as an epistemological problem solved by a procedure called rippling: this procedure ensures that the more specific heuristics are found before the more general heuristics. The heuristics themselves combine ontological notions with the epistemological notions. For example, the first heuristic above pertains to what kinds of knowledge might be used to generate interesting new concepts, whereas the third heuristic is purely a control heuristic that is strongly coupled to AM's agenda mechanism.

Also, AM's approach requires an internally formalizable domain. While RA's suggestion component is much weaker than AM's discovery component, RA's heuristics see the field from a general (subject independent) set of notions such as problems, techniques and so on. A fruitful future research topic might be to combine RA's suggestions with AM's discovery heuristics such that RA provides top-down research suggestions that are accomplished by AM-like heuristics in a bottom-up fashion (see Chapter 7).

One issue that was considered in great detail by AM but was ignored in RA was the issue of control. AM used its notion of interestingness to control which heuristics to apply. While the control issue is irrelevant to RA's ontological level claims, it is an important issue that remains to be addressed from the point of view of building practical computer-aided research systems (see Chapter 7).

6.3.1.2 Eurisko

The Eurisko system treats the learning of new heuristics as a discovery problem in itself. Lenat reports that his initial attempts at this problem met with little success until he

realized why AM was succesful in the first place [Lenat & Brown 84]. Before we come to that, let me first describe how Eurisko was used in the domain of naval fleet design.

Design of futuristic ships to compete in a wargame called 'Traveller Trillion Credit Squadron' was one of the tasks performed by Eurisko. Each participant in this game is given a budget of a trillion credits to design a fleet of futuristic ships. The game provides over a hundred pages of rules that specify the costs (and constraints) of various features that could be used in a ship, e.g., the armor plating, the weapon systems, size, maneuverability etc. The battle itself is (supposedly) tactically trivial. The goal of a participant is to design a fleet that wins the battles against one's opponents. For details, see [Lenat 83].

Eurisko is first given a fleet design (as a seed, say), and is also provided a simulator that simulates the wargame. Eurisko repeatedly designs new fleets — generated as mutations of known designs — and sees how these fleets perform in simulated battles. Fleets are allowed to fight, and Eurisko determines which design policies are winning. The design decisions that led to winning fleets are abstracted into new heuristics. Viewed in search terms, AM searches a space of concepts (in arithmetic) guided by a set of heuristics; Eurisko, however, conducts a search in two distinct search spaces, the space pertaining to concepts in naval fleet design, as well as the space pertaining to heuristics.

In reminiscing about AM and Eurisko, Lenat and Brown provide some interesting perspective on why AM was successful and why (initially) Eurisko met with rather poor results[Lenat & Brown 84]. This is summarized below.

6.3.1.3 Analysis of AM and Eurisko

Let's consider how AM arrived at the concept 'equality' from the concept 'list-equal': AM used one of its mutation operators — i.e., to generalize a concept definition that contains a conjunction, eliminate one of the conjuncts — to mutate the Lisp code that defined the concept 'list-equal.' Interestingly, this happened to correspond to a meaningful concept in arithmetic, namely, 'equality.' Lenat and Brown concluded that the success of AM was due to the close relationship between Lisp (more generally, lambda expressions) and arithmetic: simple syntactic mutations of small Lisp expressions turned out to be important arithmetic concepts as well. AM worked because, as an automatic programming system, AM contained several heuristics that were essentially syntactic mutators of Lisp code. Fortuitously, these mutations produced new code that — due to the close correspondence between Lisp and arithmetic — happened to correspond well with new concepts in arithmetic. In more abstract terms, AM was searching a space of Lisp expressions using syntactic operators; because of the close correspondence between Lisp (the syntax) and arithmetic (the semantics), AM was, fortuitously, also searching the space of arithmetic concepts.

In moving from AM to Eurisko, there was no such correspondence between the concepts relevant to the domain — i.e., the domain of naval fleet design or the domain of heuristics — and the mutation operators encoded in the heuristics. While a mutation operator that generalizes a Lisp expression (by eliminating a conjunct) also turns out to generalize the corresponding arithmetic concept, such a mutation operator may produce a useless mutation of a heuristic. Lenat and Brown note that the Lisp code corresponding to a typical arithmetic concept in AM may be a few lines long; hence a mutation such as replacing an AND by an OR results in a significant, and semantically valid, new concept. However, the Lisp code corresponding to a heuristic may be a couple of pages long; a point mutation of this code results in garbage that is of no semantic value.

The denouement of Lenat and Brown's analysis was that any approach to discovery by syntactic mutation can and does succeed only when there is a close correspondence between form and meaning: in the case of AM, there was a strong correspondence between the world of Lisp expressions and arithmetic, so that mutations of Lisp programs corresponded, fortuitously, to new arithmetic concepts. With the other domains considered by Eurisko (such as battleship design) there was no such correspondence between the world of syntax and semantics; hence, Eurisko had a hard time until a mapping between form and meaning was re-established. The successful versions of Eurisko represented both domain concepts and heuristic concepts using several new slots, with each slot containing just a few lines of code. Therefore, once again, as in AM, simple syntactic mutations gave rise to meaningful semantic concepts.

6.3.1.4 Relation to RA

In some ways, RA's approach is orthogonal to that of AM and Eurisko; in particular, the view of heuristics as syntactic mutations is inapplicable in RA's case. To understand this, let's first look at AM's problem space, depicted in Figure 6.1. The left half of the figure depicts the world corresponding to the problem domain, i.e., the world of arithmetic concepts. The right half of this figure depicts the internal world of AM — the world corresponding to Lisp expressions. The heuristic opertors are shown as curved arrows. A heuristic takes the Lisp code or the meaning representation of a concept — i.e., a representation of the *intension* of the concept, which also corresponds to what I have called the deep semantic knowledge (Section 2.3.2) — and obtains a new chunk of Lisp code. Lenat and Brown's analysis shows that if there is a good match between the world at the left and the world at the right — i.e., the world of intensions and their representations — then this newly synthesized piece of Lisp code will probably turn out to be the meaning representation of an interesting concept in the world of arithmetic.

Figure 6.2 depicts the problem space of RA. The left half of the figure shows the world of EBL along with the relationships among the EBL items. The right half shows RA's S-knowledge: the S-knowledge objects are atomic symbols connected to each other by various relations. The biggest difference between an AM-heuristic and a research schema

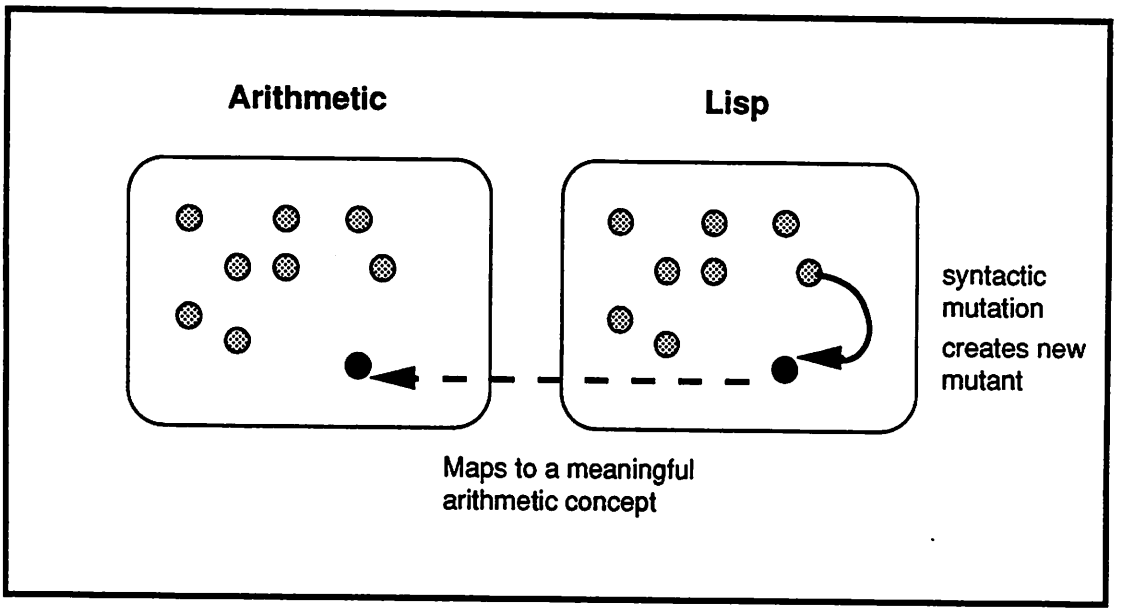


Figure 6.1: The Problem Space of AM

is that the former is a mutator of the (representation of the) *intension* of a concept: such a heuristic looks at the details of the concept's definition — i.e., its deep semantics — and determines how this definition can be mutated to obtain a new definition in the hope that the new definition will correspond to the intension of an interesting arithmetic concept. By contrast, research schemas are not mutation operators at all. In the case of RA, what is represented is not the *intension* of the EBL objects: all EBL objects are essentially atomic symbols whose meaning obtains solely from their structural relationship to other objects. When a research schema suggests a research direction as a transformation from some existing set of objects, it does so not because it has a representation of the *intension* of the objects, but because it has a representation of the *structural relationship* of these objects to each other. For example, consider the research schema shown below:

ref: {(solves T1 P1)}

def: {(entails T1 P2)
(solves T2 P1)
(not-entails T2 P2)}

This schema corresponds to the following heuristic rule:

> If there exist T1, P1 such that T1 solves P1, the suggest, "You could look for an emergent problem P2 of T1. Then you could propose a new technique T2 to solve P1 while avoiding P2."

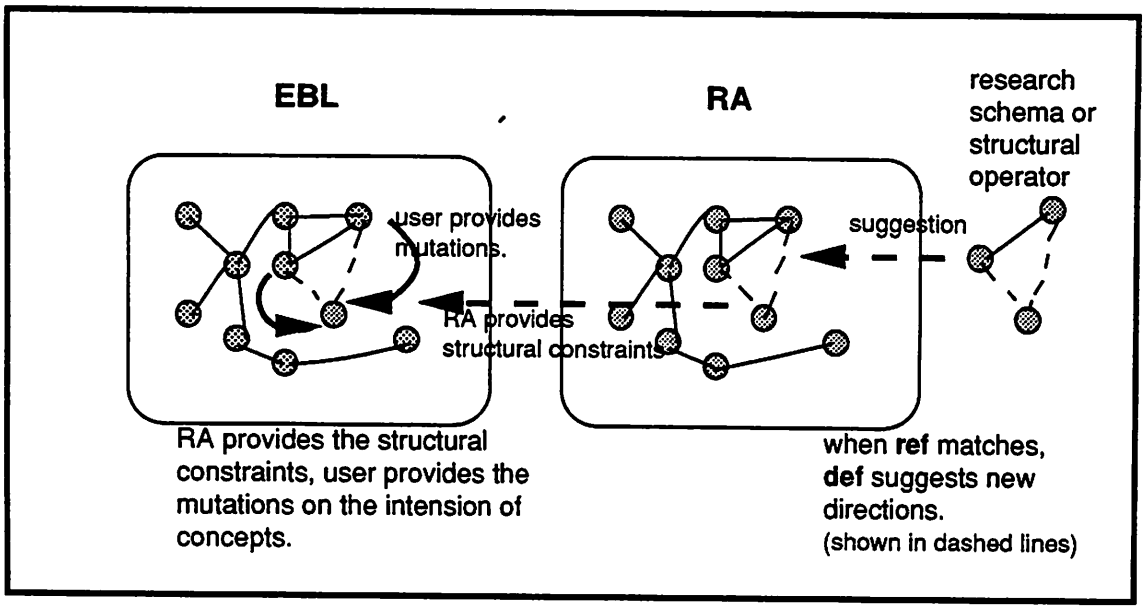


Figure 6.2: The Problem Space of RA

Since RA does not know the internal details of any of the objects in its domain, the only thing that the condition part of the rule looks for is structural relations among objects: in this case, RA looks for two objects, one of type technique and and the other of type problem connected to each other by a solves relation. Thus, the schema is totally oblivious to the intension of the two objects. This is precisely why RA cannot discover anything, but can only make suggestions. It is the user who is asked to provide the actual mutations of the intensions of the objects, i.e., mutations that will satisfy the 'specifications' (so to say) suggested by the research schemas[39]

Since the analogy between syntactic mutations (in AM) and biological evolution has been drawn before [Lenat 82], let's extend this analogy to RA. The concepts in AM (as well as Eurisko) may be likened to, say, stable biological species in an environment. The heuristic operators create new mutants from old ones. The heuristics that evaluate the interestingness of concepts determine which mutants live and which will die, some-

[39] Btw This is also the reason why this RA does not have much of a symbol level component. Since RA does not purport to represent the intension of the EBL concepts, RA does not need AM's (and Eurisko's) stringent requirement that there be a good correspondence between the domain concepts and their representation at the *operational* level. Since RA's representation of EBL entities are simply atomic symbols, all that RA requires is that its vocabulary (types and relations) be well-chosen so that there is a good correspondence between the EBL concepts and RA's representation at the *denotational* or knowledge level.

thing like the process of natural selection[40]. In contrast, research schemas may be seen as encoding the knowledge about the interactions among various species and also the knowledge about what kinds of mutants will exhibit interesting relationship to existing ones. While AM creates a mutant and sees if it will survive natural selection, RA's suggestions provide the specifications for the mutants that will survive and asks the user to provide the actual mutations[41].

So far, we have discussed the relationship between AM and RA I. Let's now consider the relationship between Eurisko and RA II. Broadly, AM uses heuristics to discover new concepts, and Eurisko attempts to learn these heuristics; similarly, RA I uses research schemas to suggest new research directions, and RA II attempts to learn these research schemas. However, the similarity between RA II and Eurisko ends there. The first major difference between RA II and Eurisko is that, unlike Eurisko, RA II does not model the process of learning heuristics as just another discovery problem in a different space (the space of heuristic concepts). This is illustrated in Figures 6.3 and 6.4. Figure 6.3 illustrates the two search spaces of Eurisko, the search space corresponding to the concepts in, say, the domain of fleet design, and the search space corresponding to the heuristics pertaining to the domain. In the latter search space, each heuristic is a point or an interesting heuristic concept. A syntactic mutation of a heuristic concept results in a new heuristic concept. This new heuristic concept, when mapped back into the space of fleet-design concepts, result in a new heuristic operator (mutator), shown as a squiggly arrow. To evaluate the new operator, Eurisko conducts simulated battles to see which ones design winning fleets and which ones design losing fleets. The good heuristics are retained, and the bad ones discarded[42].

Let's now consider how RA II acquires a new research schema. Figure 6.4 shows RA II's space of EBL entities. When given a new input or the **def** of a paper (shown in dashed lines), RA II finds a set of existing relations as the **ref** of the paper. As discussed

[40]We should note that AM's mutations are not completely random; some local criteria are used to generate only (locally) interesting mutations; however, the ultimate criterion for the worth or survival of a mutant is how well it performs in the long run.

[41]Btw A somewhat far-fetched analogy is that while AM's heuristics can be likened to biological evolution under natural selection, RA's research schemas can be likened to James Lovelock's Gaia hypothesis [Lovelock 87]. The Gaia hypothesis is that the Earth, as a whole, is a living thing called Gaia, whose goal is its own long-term survival. Hence, this hypothesis attributes intentionality to the planet as a whole: the evolution and extinction of various species is not random natural phenomena, but are in fact well-motivated actions of Gaia. Gaia determines what ecological niches need to be filled or emptied, and natural selection obliges by providing the mutants to accomplish that goal. See [Joseph 90]. Perhaps this also provides a metaphor for combining the approaches of RA and AM. See Chapter 7 (Section 7.2).

[42]Btw While Lenat sees the discovery of new heuristics as analogous to the discovery of domain concepts, I see a certain asymmetry that has, apparently, gone unnoticed: while the domain concepts are evaluated (for their worth) in their own space, the heuristic concepts are not evaluated in their own space; instead, they also seem to be evaluated in the space of the domain concepts. In other words, a heuristic's worth is determined not because it is an interesting heuristic concept in its own right, but because it is capable of generating interesting domain concepts.

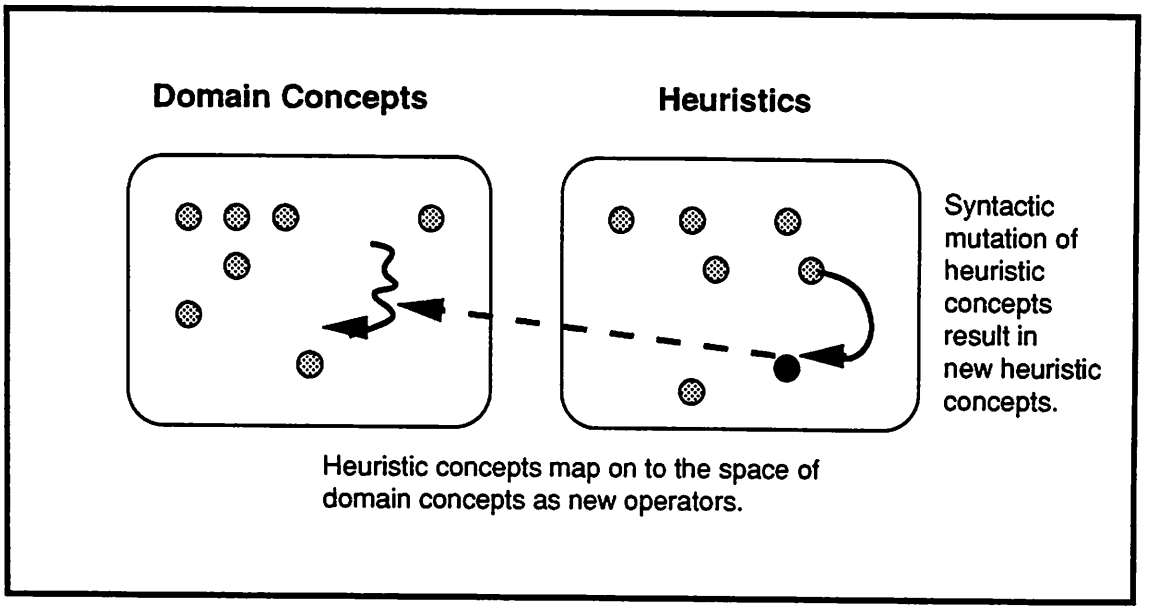


Figure 6.3: The Problem Space of Eurisko

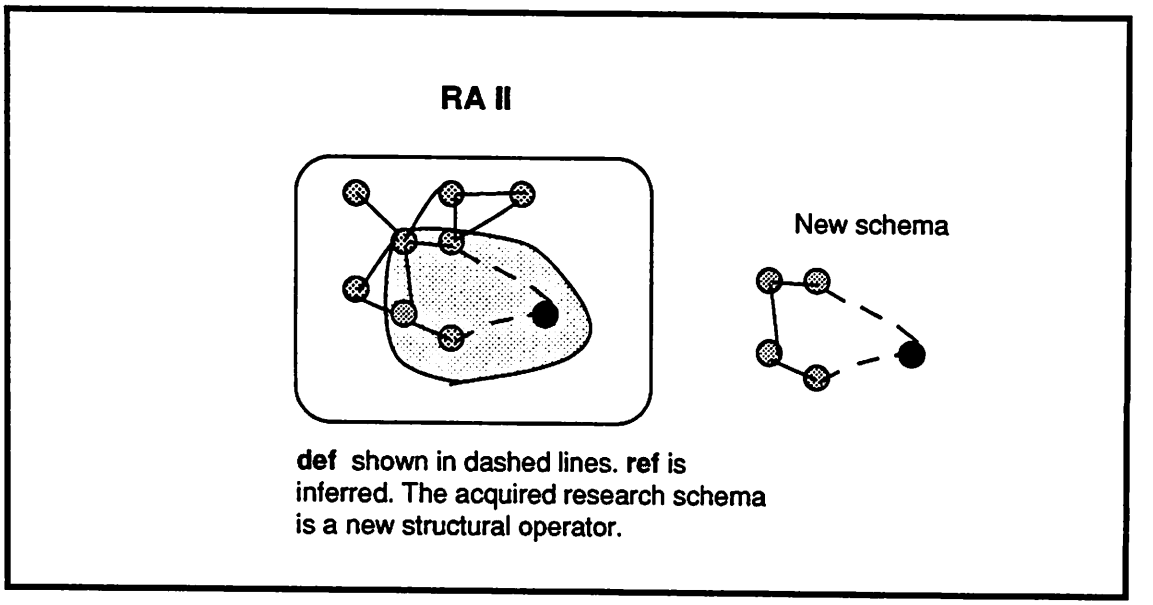


Figure 6.4: The Problem Space of RA II

in Chapter 4, and later justified in Chapter 5, the **ref** of the paper is assumed to be the set of most specific relations that just connect the pre-objects in the **def**. This **ref**, **def** pair results in a new structural operator. Thus, neither RA I nor RA II deal with the intension of the objects in their problem space, but only with structural relationships. Hence, the equation of heuristics to syntactic mutators is not a fruitful way to visualize RA's research schemas.

In order to map Lenat and Brown's analysis of AM and Eurisko onto RA I and RA II, let's assume a scenario in which research schemas are learned as variations of other research schemas (i.e., as mutants of extant existents). First, let's assume, without much ado, that RA I's internal representation is a good one to capture the *structural relations* among EBL entities in the same way that AM's internal representation (as Lisp expressions) is a good one to capture the *intension* of arithmetic functions. Now let's consider the following mapping:

AM/Eurisko	RA/RA II
AM creates new domain concepts as variations of the *intension* of existing domain concepts. There is a good mapping between the intension of arithmetic concepts and their representation in AM.	RA suggests new research directions as transformations based on the *structural relationship* among domain objects. By assumption, there is a good mapping between the structural relationship among EBL entities and their representation in RA.
Eurisko models the learning of new heuristics as a discovery problem, by discovering new heuristics as variations of the *intension* of existing ones.	A corresponding approach to learn new research schemas would be to find new research schemas as variations of the *structural relationship* among existing ones. Let's call a system that uses such an approach as RA II*.

How would RA II* work? Figure 6.5 shows the problem space of RA II*. The left side of the figure corresponds to the space of EBL objects and the right side corresponds to the space of research schemas. Points in the latter space are research schemas, and the lines connecting different points correspond to an (as yet unknown) vocabulary of the structural connections *among research schemas.* For example, consider the following two schemas:

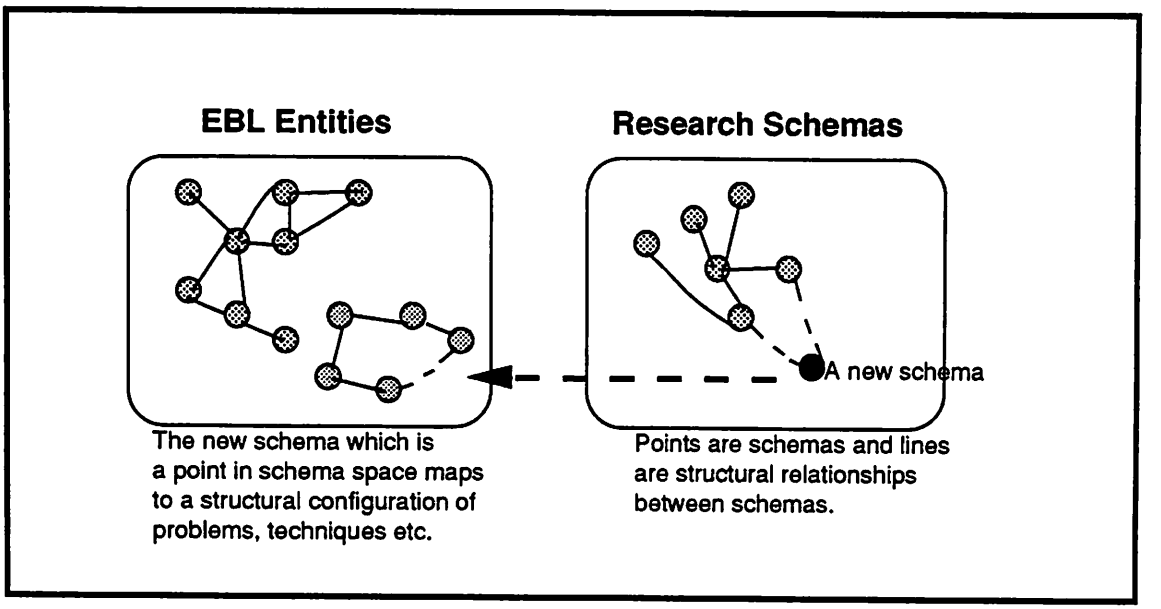


Figure 6.5: The Problem Space of RA II*

ref: {(solves T1 P1)}

def: {(entails T1 P2)
(solves T2 P1)
(entails T2 P2)}

ref: {(solves T1 P1)}

def: {(entails T1 P1)
(solves T2 P2)}

There is obviously some relationship between these two schemas: the first one solves P1 by avoiding the emergent problem P2, whereas the second one simply fixes the emergent problem. In common sense terms, if your roof leaks, the first one asks you to replace the roof and the second one asks you to find a tarp. To state such relationship between schemas, we need either a new vocabulary (both types and relations) or redefine the notions of problems, techniques, solves, entails etc. As a very simplistic solution, let's define two new types called 'elegant schemas' and 'brute-force schemas'; the first schema above is an elegant schema and the second schema is a brute-force version of the first. Let's also define a new relation called 'brutifies' and use this relation to link the second to the first. Once such a vocabulary exists to state the structural relationship between research schemas (which themselves stand for structural relationship between domain objects), then a suggestion component like RA's could in fact be used to suggest new

research schemas using meta-schemas: for example, whenever RA finds another 'elegant' schema, it could suggest that one way to find a new structural variation of this schema is by 'brutifying' it.

Now let's consider how Lenat and Brown's analysis of Eurisko would apply to RA II*. Lanat and Brown found that AM's representation, while adequate for discovering new domain concepts, was inadequate for discovering new heuristics; eventually, Eurisko's representation evolved such that the mapping between syntax and semantics that was fortuitously satisfied in AM was explicitly re-established between the space of heuristic concepts and the world of syntactic mutations. In the same way, one would find that the representation of RA (in terms of problems, techniques, solves etc) would be inadequate to learn new research schemas using an approach like that of RA II*. One would need a new representation (a new vocabulary) that is suited not to state relationships between problems and techniques, but to state relationships between different kinds of research schemas.

However, RA II (the real system) does not use a new representation to acquire new research heuristics; this is because RA II does not model the acquisition of new research schemas as generating structural variations of old schemas; hence, the analogy with Eurisko breaks at this point. The schema acquisition strategy in RA II involves an entirely different procedure with a different set of assumptions (see Chapter 4). In more abstract terms, Eurisko retained the *same process* (as that of AM) for the acquisition of new heuristics and found that it needed a *different representation*; in contrast, RA II retains the *same representation* for the acquisition of new research schemas because it uses a *different process* for schema acquisition. As Raj Reddy's quote goes, there are many paths to nirvana [Reddy 88].

6.3.2 Memory Organization

In this section, I discuss two AI systems, IPP [Lebowitz 83] and Cyrus [Kolodner 84], that deal with a memory organization for episodic knowledge[43]. This section also discusses plot-units [Lehnert 81]; even though Lehnert's work was not primarily concerned with memory organization per se, it proposes a representation involving structural primitives as in RA.

6.3.2.1 IPP

Lebowitz's IPP is a story understanding system that understands short newspaper stories on international terrorism [Lebowitz 83]. Unlike its predecessors such as Sam

[43]For readers interested in this general area, Roger Schank's *Dynamic Memory* [Schank 82] and Endel Tulving's *Episodic Memory* [Tulving 83] contain several interesting ideas for the organization of episodic memory.

[Cullingford 78], Pam [Wilensky 83], and Frump [DeJong 79] which processed each story in isolation, IPP used a so-called 'generalization-based memory' to process a given story in the context of the generalizations learned from processing past stories. When IPP is given a new story to process, it looks for similar stories in its memory. If such a story is found, IPP creates a generalization that reflects the common parts of the two stories. These generalizations are stored in memory in terms of 'specialized Memory Organization Packets' or spec-Mops. Spec-Mops capture generalizations such as "The victims of kidnappings in Italy are usually businessman." As IPP processes several stories, IPP's memory gets organized as a hierarchy of spec-Mops. Each level of the hierarchy is characterized by a set of features that define generalizations at that level. Lower level generalizations specify newer features. For example, the generalization that businessmen are kidnapped in Italy is organized as one of the generalizations stored under the spec-Mop that stands for kidnappings in Italy, with the distinguishing feature 'businessmen.' In turn, the generalizations about the kidnappings in Italy is stored under the generalizations about kidnappings, with the distinguishing feature 'Italy' as the place where such kidnappings take place.

IPP and RA are related in the sense that they are both concerned with how a memory grows in response to new input. However, IPP and RA deal with very different kinds of knowledge and use their memory for very different purposes. For IPP, a story does not provide new semantic knowledge; instead each story is a variation of similar themes IPP has seen before. IPP's motivation is to find generalizations among the input stories. In contrast, RA's motivation is to store both the subject and the evolutionary knowledge in the input. Several of the questions addressed in RA, such as what is the relationship between S- and E-knowledge, what accesses what, what is the relationship between rules and case-indices and so on are not the focus in IPP.

6.3.2.2 Cyrus

The Cyrus system was built by Janet Kolodner as a model of one's autobiographical memory for events [Kolodner 84]. Cyrus uses newspaper stories about the ex-US-secretary-of-state Cyrus Vance to model his autobiographical memory of diplomatic initiatives while he was in office. One of the major claims of Cyrus is that memory is reconstructive — i.e., often finding the right information from memory requires a reconstruction of the events to decide what might have happened rather than retrieving assertions stating what did happen. For example, when Cyrus is asked "Has your wife ever met the first lady of Egypt," it might have to reconstruct possible scenarios in which such a meeting might have taken place, e.g., state dinners. Then it has to check if Vance's wife accompanied him on those trips in which he was given a state dinner and if the first lady of Egypt was present at that event. If so, Cyrus infers that Vance's wife should have met the first lady of Egypt. Kolodner considers a memory organization that indexes particular events in memory in terms of the features that distinguish it

from other events, and proposes several kinds of reconstructive processes for retrieval from this memory.

Cyrus is concerned with a memory for events, and as such does not consider the relationship between semantic and episodic knowledge that might co-exist in memory. Suppose, while on a diplomatic trip to Egypt, Cyrus Vance had learned that camels are not native to Egypt but were in fact brought there by traders from Central Asia. There is no easy way for Cyrus to represent this kind of semantic knowledge in memory, except as encapsulations of particular episodes. In this case, Cyrus's new knowledge about camels is accessible only through the particular episode (his trip to Egypt) in which it was acquired. In contrast, RA is concerned with knowledge evolution episodes — episodes that lead to the acquisition of new semantic knowledge. Thus, in RA, the knowledge that EBL solves the learning-problem is accessible both independently as well as through the papers (episodes) through which this knowledge was acquired.

6.3.2.3 Plot Units

The theory of plot-units was proposed by Wendy Lehnert to account for people's ability for narrative summarization [Lehnert 81]. While several theories of narrative understanding are based on the actions and events in a story (e.g., scripts [Cullingford 78], plans [Wilensky 83], story-trees [Rumelhart 75] etc.), the basis of the plot-unit theory is not events per se, but how the events impact on the affect (emotional) states of the characters in the narrative. Lehnert defines three kinds of affect states (positive, negative, and neutral) and four relations (motivation, equivalence, termination, and actualization) to state the relationships among the affect states. For example, if you win a million dollars in a lottery and then lose it in a fool-hardy enterprise, then the first event (winning the money) results in a positive mental state, and the second event (losing the money) results in a negative affect state; further, the second state *terminates* the first. Proceeding this way, Lehnert constructs the affect state map of an entire narrative. Lehnert also defines a set of *plot-units* as stereotypical configurations of affect states. Some examples of plot-units are mixed-blessing, unexpected-dividend, and so on. An affect state map of a narrative is converted into a plot-unit graph that reflects the plot structure of the narrative. A summary of the narrative is constructed in terms of the the node that has the highest connectivity in the plot-unit graph. Lehnert also shows that affect state maps can be used to understand the analogy between narratives: e.g., Romeo and Juliet and the Westside story have the same affect state map in this theory.

Research schemas and plot-units have some deep similarities: both view their respective domain from a high level of abstraction. Plot-units attempt to capture the plot of narratives and research schemas are used to capture the plot (so to say) of research papers. However, plot-units are strongly tied to the notion of events and human affect states, whereas research schemas are tied to the notions of problems and techniques.

Further, unlike research schemas, plot-units were not meant as a theory of memory or memory retrieval, nor were they meant to integrate multiple kinds of knowledge.

6.3.3 Language

Although RA does not explicitly process language, this work is related to several representational theories that deal with language processing. Most work in language processing have used various kinds of knowledge structures (schemas) to process event-based texts or narratives (e.g., [Rumelhart 75] [Schank & Abelson 77] [Cullingford 78] [DeJong 79] [Wilensky 82] [Wilensky 83]) A nice feature of narratives is that they have a strong notion of events that occur in a temporal sequence. These events tend to be related to each other through strongly predictive episodic relations such as causality, intentionality, enablement and so on. Compiled structures of such event sequences and event relationships have been used as schemas for processing narratives in a top-down predictive fashion.

In contrast to narrative texts, research papers are expository in nature. Expository texts, as I discussed in Chapter 1, are characterized by a predominance of semantic relations. Semantic relations, in general, are not predictive. The few schema-based representations that have been proposed for expository texts (see [Voss & Bisanz 85]) have been based on discourse structure and discourse expectations, and not on the content of the text. In this work, I conceptualize research papers, not as isolated pieces of natural language text, but as research episodes that have predictable patterns with strong relationship to other research episodes in the field. These patterns of research provide a schema-based representation. While it remains to be seen if these schemas can be used predictively for processing research papers (in a manner similar to, say, FRUMP's processing of short stories with sketchy scripts [DeJong 79]), this work suggests that an output representation of research papers in terms of research schemas might be a useful one.

6.3.4 Learning

In Chapter 5 (Section 5.4) I explained what I consider to be a fundamental distinction between the generalization strategy in RA and that in other learning paradigms. This section discusses some other differences between RA's learning strategy and EBL, and points out an interesting relationship between the two techniques.

Explanation-based Learning (EBL) is a a class of learning techniques that use considerable domain knowledge to achieve learning (e.g., [Mitchell et al 86] [DeJong & Mooney 86] [Silver 83] [Mooney 88] [Minton 88] [Keller 87]). Typically, the domain knowledge expresses the causal association among the concepts in the domain. This knowledge is stated as general *uninstantiated* rules. For example, in Mitchell et al's EBG approach

[Mitchell et al 86], one uses causal rules like the one shown below:

$$safe\text{-}to\text{-}stack(x,y) \Longleftrightarrow fragile(x) \vee lighter(x,y)$$

The learning system is given a target concept to learn, plus an instance of that concept. In the EBG approach, the learning process involves two phases: during the first phase, also called the explanation phase, the system constructs an explanation as to why the example belongs to the target concept. This explanation is constructed by instantiating the domain theory rules, i.e., variables in the rules are instantiated to specific constants to construct a specific explanation about the given example. Thus if the system is given a 'red-block' and an 'end-table' and told that it is safe to stack the red-block on the end table, then the variable x is instantiated to the red-table and the variable y is instantiated to the end-table. The second phase involves the generalizaton of the explanation into a concept descrription. During the generalization process, the domain rules are 'regressed' through the explanation in order to obtain the generalized constraints on the constants in the explanation. A related strategy is used by DeJong and his colleagues in acquiring schemas from input short stories. The EGGS algorithm used by DeJong and Mooney is similar in spirit to the EBG algorithm but uses unification instead of goal regression [DeJong & Mooney 86]. For a comparison of various generalization algorithms, see [Mostow 87] and [Mooney & Bennett 86].

In some ways, RA's schema-acquisition strategy is similar to EBL. Given the relations belonging to the **def** of a paper, RA acquires the schema of the paper. This is done, as in EBL, by a two step process: during the assimilation phase, RA obtains the instantiated schema of the paper; during the generalization phase, RA converts this into a skeletal schema by replacing the constants in the schema by typed variables. Unlike EBL systems, however, RA does not use any domain theoretic rules[44]. Instead the assimilation process involves accessing the memory directly to find the most specific (instantiated) knowledge that connects the various *pre-objects* in the input (**def**). In that sense, RA's strategy may be contrasted as a memory-based learning process from EBL which may be seen as a knowledge-based learning process.

The generalization strategy of RA is also very different from the generalization strategy of EBL. Both have the problem of determining the category affiliation of the constants in their respective explanations (i.e., the instantiated schemas in RA's case) in order to generalize the constants into variables constrained to those categories. EBL obtains the constraints from the domain theory rules. In contrast, RA has no basis for generalization. As we saw in Chapter 5, RA's strategy of simply replacing the constants by variables of types Problem, Technique, Concept, and Property is justified for a deeper reason: these types are both associative and discriminative and hence are the most differentiated categories in RA's world.

[44]It is not even clear what kinds of causal knowledge would go into acquiring the kind of heuristics that RA acquires.

While EBL is obviously a general learning technique, it appears that RA's learning technique might be something quite specific to RA and research schemas. Is there any generality behind this technique?

First of all, let me point out that the assimilation phase is in fact a well-known weak method for natural-language inference; hence, one could make a spirited claim about the assimilation phase that is completely independent of learning: assimilating a piece of new knowledge into memory involves the ability to relate this knowledge to the most specific knowledge in one's current ontology. As an example, consider the seemingly random pieces of information put out by Harper's Index (in Harper's Magazine). Typically, Harper's index lists random statistics that are cleverly juxtaposed with each other. A couple of examples are shown below:

Price of "Video Dog," a tape offering "the experience of owning a pet without the mess and inconvenience": $19.95.
Price of "Video Baby": $19.95.
Percentage of Americans earning less than $15,000 a year who say they have achieved the American dream: 5
Percentage of Americans earning more than $50,000 a year who say they have achieved the American dream: 6

Let's take the first two pieces of information above. In isolation, these are two new facts, somewhat like the relations in a **def**. However, most of us do not perceive these as completely unrelated facts about pets and babies; to 'really' assimilate this information into your memory, you not only have to connect these facts to your ontology, but you also have to find a connection between pets and babies — i.e, you have to construct something like a **ref**. As such, this is an age old claim about inference that can be attributed to several others, including Quillian [Quillian 67], Rieger [Schank & Rieger 74], and Norvig [Norvig 87].

What is new in RA is that such inferences are used for learning. An input, plus the inference (i.e., a **def** plus the **ref**) is simply converted into a general rule. In RA's case, this is possible because of RA's category structure: each instantiation (i.e., each constant) has a clear category of primary affiliation, so RA generalizes a fact about a specific object into a general fact about all intrinsically similar objects[45].

If RA's assimilation phase is essentially the same as finding a connection between two pieces of input (similar to natural language inference), then the question of whether this is a general learning technique boils down to the question of whether general inference techniques exist[46]. In RA's case, the inference was straight-forward — we did a simple

[45]In the case of the Harper's Index, in contrast, there is no 'obvious' generalization, even though one could posit an assimilation process similar to RA's

[46]Biw In the natural language context, this question is as yet unsettled. In her survey of natural language inference, Wendy Lehnert shows how the field has been shuttling back and forth between

intersection search to find a connection among the pre-objects in the **def**; further, we used a simple total ordering of RA's relations to choose one connection among many. A crucial question that was swept under the rug (in RA) was whether a *single* total ordering is enough for acquiring *any* schema. For the examples considered in RA II, this turned out to be adequate. Since RA's vocabulary was quite small, I used the simple strategy of using a total ordering of the relational predicates, with the domain-dependent relations having priority over the epistemological ones. I suspect that if the memory is richer with more types and relations, the process of finding a connection among the pre-objects in the **def** will in fact turn out to be an explanation process: since a simple ordering of the relations is likely to be inadequate with respect to all schemas, the learning system may have to 'understand' a schema (in some sense) in order to choose a set of relations that is most appropriate for connecting the pre-objects in the **def** of *that* particular schema. The only way to verify this speculation is to build a learning system with a richer set of primitives and by considering a much larger number of examples than was attempted in RA II. Let's assume that such a system, called RA II**, can be built.

This raises the question as to whether the learning technique in RA II** will simply turn out to be EBL. If RA II** needs explanations in order to choose one set of relations (to include in the **ref**) over others, doesn't that mean that RA II** will simply be doing EBL? The answer is no. In EBL, an explanation is used to provide a basis for generalization: in particular, an explanation isolates certain properties of the *intension* of the input example as relevant to the target concept. Hence, for example, an explanation may decide that given a particular green cup, the handle is an important property of the cup, but the color is not. The domain theory and the explanation provide a way for a learning system to isolate relevant from irrelevant properties of a concept, so that the system can learn a general concept description that inlcudes only the relevant properties. In contrast, RA II** sees the intension of the concepts as unanalyzable atomic entities; the category structure of the world provides a simple and unique generalization for each object. However, there can be many structural relationships between an object and other objects. If RA II* needs an explanation, this explanation is not for choosing the right set of properties belonging to the intension of an object (as in EBL), but for choosing the right set of *structural relationships* among the objects.

Thus, in mapping RA II** onto EBL, the assimilation phase of RA II** corresponds to EBL's explanation phase; further, the instantiated schema of RA II** corresponds to EBL's explanation. If RA II** turns out to require an explanation in order to find the **ref** of a paper, then it requires an explanation for finding the instantiated schema, which corresponds to an explanation in EBL. In other words, EBL, in its purest form[47],

strong and weak methods for inference and how the pundulum has, in recent years, swung in favor of weak methods [Lehnert 88a].

[47]With assumptions of correct, complete, and consistent theories. See, for example, [Mitchell et al 86]. Several researchers have since addressed the problem of incorrect, incomplete and inconsistent theories. See [Ellman 89] for a survey.

assumes that there is one relationship among the objects (i.e., one explanation) and uses the explanation to choose one among many generalizations. RA II** assumes that there is one generalization (determined by the category structure) but uses an explanation to choose one among many structural relationships (instantiated schemas). Hence, we are still far from being able to choose one among many paths to nirvana!

6.4 Summary

In this chapter, I first defined several terms to classify AI theories concerned with knowledge and knowledge representation. This classification identified several dichotomies: ontological vs epistemological theories, content-dependent vs content-independent theories, knowledge-base vs memory, and semantic vs episodic knowledge. After developing this framework, I discussed how RA can be understood in terms of the ideas in this framework. RA is best seen as a content theory of memory for scientific research at the ontological level. I also discussed several closely related pieces of work along several topical areas such as heuristic discovery, memory, language, and learning.

Chapter 7

Wrapping Up

During the morning of the seventh day, Ra seemed, oddly enough, a little less disjointed; the ropes seemed to be tauter.

—Thor Heyerdahl, The RA Expeditions.

What is the research schema of this dissertation? Research schemas and RA's memory can be seen in terms of at least three different research schemas, each focusing on a different contribution of this work (a different **def**) and relating that contribution to a different body of knowledge (a different **ref**). These schemas pertain to memory, language, and heuristic discovery.

Could RA have suggested this work? This is like the question, "If God is so powerful, can he make a stone so heavy that even he can't lift it?" Either way, God loses. If RA, a menial computer program, could have suggested this work, there is nothing interesting about this work. However, if RA could not have suggested even this work, there is nothing interesting about RA. Seriously, could RA have suggested this work? We will see.

We have covered a lot of ground in the last six chapters: we have seen the RA program, its memory, how it works, how it can acquire its research schemas, and why the schema acquisition strategy works. But there are still promises to keep and miles to go. Several important problems pertaining to this work still remains to be addressed. Research. Table 7.1 contains a guide to Chapter 7.

7.1 The Schemas of This Dissertation

After having talked at length about research schemas and how they model knowledge evolution, it is only fair to ask for the research schemas of this dissertation. The main ideas of this work can be described in terms of at least three different research schemas.

Section	On first reading	Description
1	read	Describes the schemas of this work and discusses whether RA could have this work.
2	read	Suggests several topics for future research.
3	skim	Reviews the various topics covered in this thesis. and points to chapters where they are discussed.

Table 7.1: Guide to Chapter 7

Each of these schemas focuses on a different statement of RA's contributions (i.e., a different **def**); correspondingly, each schema accesses a different body of background knowledge (i.e., a different **ref**). The schemas are listed below:

> [**Schema 1**]: The problem of knowledge-representation consists of two kinds of problems, the problem of semantic-knowledge-representation and the problem of episodic-knowledge-representation [ref]. In this work, I propose the problem of scientific-knowledge-representation that is an instantiation of both these problems. I propose a technique called research-schemas to solve this problem.

> [**Schema 2**]: The problem of text-representation consists of two kinds, the problem of narrative-text-representation and the problem of expository-text-representation [ref]. The technique of schema-representation solves the narrative-representation problem [ref]. In this work, I propose the problem of scientific-papers-representation which is an instantiation of expository-text-representation. I propose a technique called research-schemas to solve this problem. Research-schemas are R-related to (analogous to) schema-representation technique that solves the narrative-text-representation problem.

> [**Schema 3**]: Discovery-in-arithmetic is one kind of (i.e., an instantiation of) discovery-problem [ref]. AM-heuristics technique solves this problem [ref]. In this work, I propose research-schemas technique to solve the discovery problem. Research-schemas technique is R-related to (weaker than) AM-heuristics technique.

These schemas are somewhat convoluted because of RA's impoverished vocabulary, but their implications are clear. The first schema focuses on the episodic-semantic distinction, and explains how this work relates to these two kinds of knowledge. The

second schema focuses on the expository-narrative distinction, and explains how research schemas are analogous to the kind of schema representations that have been used proposed for narrative texts. Finally, the third schema focuses on the use of research schemas as heuristics and relates this work to AM.

Now let's consider the question, "Could RA have suggested this work?" Each of these schemas is fairly simple; if RA's knowledge-base were right, RA could make each of the following suggestions:

> Can you find a problem P1 that is an instantiation of both episodic and semantic knowledge representation problems? Then you could propose a technique T1 to solve it.
>
> Can you propose a representational technique T1 for expository texts that is analogous to schema-representations for narrative texts?
>
> Can you find a technique related to AM's heuristic technique but to solve discovery problems for real research fields?

Each of these suggestions is within the realm of possibilities. However, what is interesting about this work is that these three schemas constrain each other. For example, the third schema constrains the second by restricting expository texts to a particular class of expository-texts, namely research papers. In turn, the second schema restricts the heuristics of the third schema to be the schema-representation of the research papers. Finally, the domain of research also turns out to be a problem that involves both semantic and episodic components as required by the first schema. The ability to see such mutual constraints among multiple schemas requires significant amount of domain knowledge; thus, sadly, no simple-minded extension to RA could suggest this work.

Let's stretch our imagination and assume that such a system does in fact exist. This system, say RX, prescribes only 'interesting' suggestions, where interesting suggestions are those that are obtained from the mutual interactions among multiple research schemas. Let's assume that RX can come up with the following suggestion: "Heuristic rules have been used to solve the problem of discovery in the context of internally formalizable domains such as arithmetic. You could consider what heuristics exist in real contemporary research domains. Research results are written up in research papers, which are, by the way, expository texts. Let's see how we handle the sibling problem, narrative texts. Narrative texts are represented using schemas which are a compiled set of stereotypical event configurations. Do such event configurations occur in research? If so, how would you characterize them as heuristic rules? Now, expository texts are related to semantic knowledge, whereas event configurations are related to episodic knowledge. Hence a representation of research papers as event configurations integrates semantic and episodic knowledge." Suppose RX had come up with this suggestion a couple of years back. Would I have found it interesting? Definitely. Well, maybe...

7.2 Future Work

"Then perhaps we can agree that the problem is still unsolved?"
He hesitated barely a second.
"Yes," he said with conviction. "That's exactly what I think."

—Thor Heyerdahl, The RA Expeditions.

The work reported in this dissertation touches upon several interesting problems in AI and suggests several important issues for future research. These issues are discussed in this section, grouped under the following three categories: (1) Representation and Memory, (2) Learning and (3) Technological Issues. Figure 7.1 summarizes this section.

7.2.1 Representation and Memory

7.2.1.1 Structural Representations

Throughout this dissertation, I made a distinction between *deep-semantic* knowledge and *structural* knowledge. While I made some effort to clarify this distinction in Chapter 2 (Section 2.3.2) and again in Chapter 6 (Section 6.3.1), I have *not defined* the notion of structure in any clear or unambiguous manner. The idea of structure and structural representation seems important and needs further articulation. More specifically, does this kind of knowledge exist in other domains? Can we recognize a structural representation if we see one? What are the properties of such a representation? Is 'structure' just a new name for knowledge at high levels of abstraction? In particular, what is the relationship between structural knowledge and basic-level categories?

In my analysis of the relationship between RA and AM (Section 6.3.1), I referred to AM as using the knowledge pertaining to the *intension* of its concepts, and RA as using the knowledge pertaining to the *structural relationships* among its concepts. Perhaps this suggests a way of combining the approach of AM and that of RA. For example, one approach to building discovery systems might be to use RA's approach to formulate research directions in a top-down manner (by looking only at the structural relationships) and use AM's approach to provide the actual syntactic mutations in a bottom-up manner (by looking at the intensions of objects).

Researchers in the area of analogical reasoning have been concerned with the notion of 'structure' (see for example, [Gentner 83]). In this dissertation, I have not explored the relationship between the use of this term in RA and the use of the same term in theories of analogical reasoning. An exploration of this relationship might be an interesting one. For a comprehensive survey of analogical reasoning and for pointers into the literature in this area, see [Hall 89].

7.2.1.2 Memory Organization

In this work, I considered how episodic and semantic knowledge interact; to do this, I used a very stylistic problem, knowledge evolution, in a very stylistic domain, scientific research. While I believe that this work demonstrates the conceptual distinction between the two kinds of knowledge and that they ought to interact, this is by no means a general solution to the problem of integrating semantic and episodic knowledge in memory. At this point, it is not clear how this organizational scheme applies to other domains. Evaluating the generality of this organizational scheme with respect to other problems is a fruitful direction for further research.

The memory organization in RA views a new piece of knowledge (**def**) as a transformation from an existing ontology. The acquisition of this new knowledge is conceptualized as an episode and is represented in terms of the **def** plus a small slice of the ontology to which the **def** directly relates (**ref**). This **ref**, **def** pair somehow stands for some underlying (research) strategy. It is not clear how this relates to research in failure driven learning as in Chef or impasse-driven learning as in Soar. Exploring these connections is an interesting topic for research. Some relevant reading in this area include [Schank 82], [Riesbeck & Martin 85], [Laird et al 87] and [Hammond 89]. For several interesting ideas about episodic memory, but from a psychological viewpoint, see [Tulving 83].

7.2.1.3 Concept Drift

Perhaps the most serious problem in building a large memory to describe any interesting domain is the problem of 'concept drift': as the memory grows, concepts in the knowledge base evolve to mean different things in different neighborhoods. While I do not believe that it is possible or even desirable to build absolutely consistent (in a sterile formal-logic sense) knowledge-bases for any interesting domain, it is desirable, even from a cognitive viewpoint, to eliminate concept drift as far as possible. First of all, people do manage to operate in the world with some reasonable efficiency, suggesting that their ontologies cannot be totally random. Secondly, different people, despite embodying different world views, do manage to make themselves intelligible to each other. Hence, there should be some core of cognates that different people understand in some mutually consistent ways[1]. Finally, the basic-level effect suggests that, at some level of abstraction, even different (and isolated) populations of people might have consistent ontologies.

In Chapter 5, I outlined some possible experiments to address the problem of concept drift. Briefly, the experiment was to see if there is at all a correlation between consistency and the level of abstraction: if different people were to build a knowledge-base to describe, say, a house, are their descriptions likely to be most consistent at the basic level of description? I believe that such an experiment would be of great interest not only

[1] [Biw] This is the idea called 'consensus reality' [Lenat & Guha 89].

from the point of view of building large memories, but also from a psychological point of view. It is also likely to have tremendous implications for knowledge representation and man-machine communication. For some relevant reading, see [Lenat & Guha 89], [Suthers 89], [Cohen & Stanhope 86], and [Schank et al 90]. In recent years, several researchers have claimed that the view of representations as static existents in human memory is simply the wrong way to think about representation. These researchers have claimed that representations are dymanically constructed in a *functional* and *indexical* manner. For some reading in this area, see [Agre 88], [Clancey 89], and [Suchman 87]. One might reconcile between this view and basic-level effect by speculating that perhaps representations *exist* at the basic-level, but are dynamically *constructed* at other levels. Finally, to throw a glitch in the whole works, Rosch herself believes that the basic-level effect (and the prototypicality effect) should *not* be interpreted as saying anything about representations. Rosch's views are described in detail in [Lakoff 87][2].

7.2.1.4 Functional Flexibility

Research schemas have been shown to exhibit a considerable degree of functional flexibility — i.e., they support a number of conceptually different functions, and stand for different kinds of knowledge with respect to different functions. With respect to the chronological summarization component, they act as a description of papers; with respect to the suggestion component, they act as intentional rules for action; with respect to the analogical summarization component, they stand for the underlying research strategy of a paper. In Chapter 5 (Section 5.4.3), I made an attempt to explain why this might be so. This explanation needs to be evaluated. One possible approach to evaluate this explanation is to design knowledge structures for other domains to see what properties of the representation support what kinds of functionalities. Alternately, given a knowledge structure, can we predict its properties? One kind of knowledge structure that seems to have the potential for such an analysis is the idea of 'abstract strategies' proposed in [Collins & Birnbaum 88].

From a methodological viewpoint, I believe that my explanation of the functional flexibility of research schemas (in Section 5.4.3) suggests a new methodology for the explanation and evaluation of representational schemes. Despite knowing that my explanation was purely speculative and possibly incorrect, I chose to write it down because I believe that the time is ripe for AI to engange in such abstract analyses of knowledge representations.

In building AI systems, it is all too easy to focus on what the system does and how well it does it, without worrying about why it works. The obsession with performance

[2] BTW To help make sense of all these conflicting views, see Fodor's analogy between scientific research and the frame problem [Fodor 87]. Fodor states that science solves the problem of conflicting views the same way AI systems solve the frame problem — ignore everything that is not relevant to what you are doing!

and performance evaluation, it appears to me, tends to direct a researcher (particularly, a young researcher) to engineer a system without worrying about whether this engineering contributes in any fundamental way to the principles underlying the system[3]. I believe that abstract analyses of representations and their properties is almost virgin territory and contitutes a very fruitful direction for exploration. For somewhat related ideas about methodology in the design of intelligent 'agents', see [Cohen et al 89]. Also see [Kintch et al 84] for several interesting papers on methodology in AI and cognitive science.

7.2.2 Learning

RA II's learning strategy consists of two phases: during the assimilation phase, RA acquires the instantiated schema of a paper; during the generalization phase, the instantiated schema is generalized by replacing the constants by typed variables. This results in the skeletal schema of the paper. Chapter 5 contains an extensive analysis of RA's learning technique and its underlying assumptions.

A crucial question about this technique is whether it has any generality: is this something specific to RA and research schemas or can it be applied to other domains? In Chapter 6 (Section 6.3.4), I showed that the assimilation phase of RA is in fact a well-known weak method for natural language inference traceable to Quillian [Quillian 67]. Hence, the question of whether this is a general technique for learning boils down to whether there is a weak method for inference. In Chapter 6, I also pointed out that, if we build a learning system, say RA II**, with a larger vocabulary of types and relational predicates, we would expect that the problem of choosing one among many paths (to connect the pre-objects in the **def**) will become a significant problem to address. There I suggested that one may need an explanation process (that reasons about the particular schema being learned) to choose among the many possible paths to infer the **ref**. An alternative approach might be to use syntactic criteria in choosing one path over others. For an example of this approach, see [Norvig 87]. Norvig models natural language inference as marker passing, and uses syntactic criteria, such as the shape of the markers, to choose valid inferences. Also see [Lehnert 88a] for a survey of several approaches to natural language inference.

One interesting direction for future work would be to take something like DeJong and Mooney's domain of stories in kidnapping [DeJong & Mooney 86], and sanitize the domain such that generalization is straight-forward; this can be done by defining the well-differentiated categories in the world as the types to generalize to. Hence, with the generalization problem finessed, we can attempt to model schema acquisition as a

[3] BTW My proposal for this dissertation, in retrospect, seems to embody this compulsion. In the proposal, I had level 1 schemas, level 2 heuristics etc; the sole aim of such monsters was to squeeze some performance out of RA.

memory-based inference process as opposed to a knowledge-based explanation process. Addressing this question will involve tracking two parameters. The first is the tradeoff between accuracy of the acquired schemas and the effort in learning. Obviously, the schemas learned through a causal analysis (as in DeJong's and Mooney's approach) are likely to be more accurate than those learned through an inference process that simply tries to find some connections among the various pieces of an input story. However, the former involves considerably more work in finding a fully causal explanation than the latter. A second parameter to track is whether, as the size of the memory grows, the inference process itself needs considerable domain knowledge to choose one inference over others (see the discussion on RA II** in Section 6.3.4) or whether simple syntactic criteria can be used (as in [Norvig 87]) for inference and hence learning.

All of the above, however, rests on the assumption that the world is in fact structured, and generalization is simply the process of going from a specific instance to the category of primary affiliation of the given instance. While not a very new position within psychology[4], this is somewhat of a new and radical position with respect to machine learning. Any research leading toward either the substantiation or the refutation of this position would be an interesting extension of this work. In particular, a substantiation of this position would challenge the established notion that induction is search: if generalization, particularly, heuristic generalization, is simply the process of going from an instance to a basic-level category[5], then there are precious few concept hypotheses considered during generalization; therefore, the view that generalization is a search process [Mitchell 81] will need to be discarded.

Having said that, let me point out how my position can be refuted. There are at least two ways: the first involves a demonstration that the basic-level effect does not easily extend outside the world of natural and man-made objects; in particular, one can show that there are several interesting natural worlds (say, the world of ideas) which have no discernable structure. In such a case, the view that the world is structured is not be a very interesting view for anybody to lose sleep over.

The second, and I believe, a better strategy to refute my position is that, with respect to learning and generalization, it does not matter whether the world is structured or not. That is, the assumption that the world is structured does not and need not significantly influence how we learn and generalize. Chapter 5 (Section 5.4.4), discusses some experiments and possible research directions. For some relevant reading, see [Rosch et al 76], [Adelson 85], [Lakoff 87], [Mitchell 81], [Oshersen et al 90], [Murphy & Medin 85], [Langley 86], [Gardner 85] (Chapter 12), and [Fisher & Langley 90].

[4]One can surmise this position, with some mental effort, in [Rosch et al 76].

[5]Taking into account the various tradeoffs with respect to the amount of work involved in learning, accuracy of the generalization, amount of work involved in applying the generalization, and the number of different situations in which the generalization is applicable etc.

At a more philosophical level, my speculations about learning and generalization is related to the position that the structure of the environment ought to constrain the design of an intelligent agent. For an articulation of this position, see [Cohen et al 89]. This position also seems to be related to the idea of *situated actions*, i.e., the richness and structure of the environment is at least partly responsible for the richness and structure of intelligence. For some relevant reading in this area, see [Suchman 87] and [Agre 88].

7.2.3 Technological Issues

Coming to somewhat lighter topics, let me discuss some future directions with respect to RA as a purely technological artifact. During the evolution of this dissertation and the RA program, the technological goals were sacrificed somewhere along the way to move on to other topics such as the acquisition of research schemas and the analysis of the representation in terms of basic-level categories. Hence, this dissertation does not answer questions regarding whether RA, as a computer-aided research system, has any credibility. In particular, did any of RA's suggestions lead to any interesting discoveries?[6,7]. In this work, no effort was made to answer any concrete questions about RA as a per-

[6] BIW There was one, based on a particularly trivial suggestion by RA. In [Segre 87], Segre had shown that there is a tradeoff between operationality and generality in EBL. Later, Richard Keller argued that operationality and generality are two distinct dimensions [Keller 88]. In RA, both these statements had been represented as properties called segre-property and keller-property, attached to EBL and operationality. Since Keller had refuted Segre, using the same heuristic, RA made the incredibly trivial suggestion that I could refute Keller. Stimulated by this (for some unknown reason), I was in fact able to find a flaw in Keller's argument.

[7] ML In his paper, Keller confuses among at least three different interpretations of the term 'generality.' One interpretation involves the generality of a concept in the world without reference to a learning system. For example, he says that, for the purposes of teaching, a more general concept might be more understandable, and hence more operational. However, if the system were to *learn* this concept by truth preserving transformations from the goal "useful-concept-for-teaching," then the final concept description, i.e., the more operational one, is more specific than "useful-concept-for-teaching." Then he shifts his position, forgets about the world, and talks about generality with respect to *his* system, MetaLex. Keller argues that his MetaLex system is a counter-example to the belief that there is a tradeoff between operationality and generality. MetaLex takes a heuristic operator and a set of symbolic integration problems, and *operationlizes* the operator by using two transformations called *truify* and *falsify* [Keller 87]. Keller's argument is that these transformations result in a more operational *and* more general concept. The flaw in the argument is a confusion between a syntactic interpretation and a common-sense interpretation of the term 'generality.' Briefly, assume that the statement $A(x) \wedge B(x)$ is true in the world. In a technical or syntactic sense, removal of one conjunct from the statement is a generalization. Suppose you construct toy worlds $W1$ and $W2$ so that $A(x)$ holds in $W1$ but not in $W2$. From within $W1$, $A(x)$ appears, syntactically, like a generalization of $A(x) \wedge B(x)$ because it has less constraints. However, $A(x) \wedge B(x)$ is really the more general statement because it holds for all x's, both $x \in W1$ and $x \in W2$. Similarly, Keller's operationalizing transforms only *appear* to be more general because they cater to a smaller world in which the eliminated conjuncts were always true. Thus the operationalized operator (that MetaLex learns) is not *really* more general outside of the problem set (in the same sense of generality as with the first interpretation) for which MetaLex performs the transformations. In a nutshell, if operationality determines how far we continue the explanation from the target concept, and if the explanation is constructed from a target concept using truth preserving

formance system. Part of the reason was that RA was built in much too hap-hazard a manner to be able to support any experiments with the system. Its knowledge-base had several inconsistencies as my ideas regarding the primitive relations and objects evolved. All the different versions of RA, in combination, have considered about 60-odd papers in total. While I believe that this dissertation provides support for its ontological level claims, RA, as a computer system remains to be evaluated. This section discusses how work along these lines could proceed.

To realize computer-aided research systems for real contemporary research disciplines requires a much bigger and better vocabulary than was attempted in RA. For a discussion on why RA's vocabulary was kept small, see Section 6.2.1. While the primitives in RA might suggest some criteria for the choice of new ones, it is quite unclear how the vocabulary should be chosen. What are the tradeoffs between a large vocabulary and a smaller one? I would speculate that a larger vocabulary will provide a more natural way to map papers into schemas, but is likely to result in a higher degree of concept drift. In contrast, a smaller vocabulary might exhibit the reverse problem. Such experiments provide interesting directions to explore.

Another problem to address on the road to converting these ideas into a technology is the problem of handcoding RA's memory. Can we devise a method to acquire the schema representation of a research paper automatically? Addressing this problem was one of the original goals of this work, but, somewhere along the way, it was sacrificed in the interest of finishing and leaving town. However, some of my colleagues did address this problem and developed the novel idea of *conceptual references* [Lehnert et al 90].

Since research schemas attempt to capture the relationship between a given paper and other papers in a field, one could look for these relationships in reference sentences, i.e., sentences in a paper that contain an explicit reference to other papers. Based on a small corpus of papers from EBL, they show that there are just a few conceptual reasons why papers reference each other; further, they also show that it might be possible to map reference sentences onto these conceptual reference types without recourse to any knowledge specific to EBL.

Conceptual references are related to work in citation indexing as follows: citation indexing notes that papers reference each other and these reference graphs have interesting properties. The idea of conceptual references notes that papers reference each other for a specific set of reasons, thereby putting labels on the arcs of these reference graphs. This is similar to Quillian putting arc labels on associative nets to obtain semantic nets [Quillian 67]. Typing the arc labels can now enable typed inferences. Further, the idea of conceptual references is also a content theory: it does not just say there ought to be arc labels, but provides a specific set of labels with some claims of generality.

---

transformations, the final operational concept cannot be any general — under the first interpretation of generality — than the target concept.

Suppose we have a memory that consists of the research schemas of several papers in a field. When we are given a new paper to process, suppose we can find its conceptual references — this tells us the specific relations through which this paper is related to the other papers, whose research schemas we already know. The question is, based on this information, can we infer the instantiated schema of the given paper?

At this point, there is no reason to believe one way of the other, and hence it is an interesting problem to address. I do know that I can often tell what a paper is about by simply looking through its references; I am also told that journal editors skim through references to decide what a paper is about and who should review it. So here is a problem that is worth addressing, not only as a technological problem, but also as a cognitive problem. For some reasding in this area, see [Lehnert et al 90], [Garfield 79] [Garfield 64].

Some readers might have realized that the major question with RA-like computer-aided research systems is not whether they can be built, but whether they are useful. This is a fair criticism. While I am convinced that RA's ideas of chronological and analogical summarization (plus the hypertext interface) can all be extended into useful technological tools, I am much less convinced about the usefulness of RA's suggestion component[8]. While I do believe that the suggestion component can be engineered so that it achieves a high hit rate (i.e., a high percentage of good suggestions), my expressed doubts are not so much technological but sociological. Systems like RA — unlike problem-solving systems that work by themselves and in isolation — involve a vision of computers, not as a slave, but as a collaborator. Hence, the feasibility of such systems is more than a technological issue: it also depends on how the human half will interface with the system. Let's assume that RA can be engineered to achieve a particularly high hit rate of good research suggestions. Will this result in good research?

The first problem is the mismatch between good research and good research suggestions. Showing that $P = NP$ is good research, but suggesting that you show $P = NP$ is not a particularly bright suggestion. From the human side, the goodness of a suggestion depends on the confidence the human has on the source that makes the suggestion. In a university setting, if a student takes his advisor's suggestion seriously enough to spend some time thinking about it, it is not necessarily due to any inherent worth of the suggestion, but because of the student's confidence in the advisor. For example, Marvin Minsky's frame theory [Minsky 75] stimulated a tremendous amount of research in AI not because the theory had very many concrete suggestions but because of the stature of the person making the suggestions[9]. If it is a computer system that suggests research directions, the question of the human's confidence in the system (despite the

[8]I should emphasize that this ambivalence about the suggestion component stems purely from an application viewpoint. With respect to the ontological level claims of this work, the suggestion component served a very important purpose: it illustrated how schemas can also serve as intentional rules for action.

[9]See Brachman and Levesque's introduction to Minsky's paper [Brachman & Levesque 85, p. 245].

merits of the suggestions) cannot be ignored. Further, in a human setting, a research advisor might come up with one suggestion per month, so that a student can take it seriously enough to explore. If a computer system can come up with a suggestion a minute, there is a serious mismatch between the human part and the machine part of the overall interaction[10]. Finally, although systems like RA are interesting and fun to think about, one may wonder whether they are an attempt by scientists to further their own careers at the cost of suggesting applications that might, if realized, *destabilize* and dehumanize well-established human systems. I believe that an inquiry into such questions is integral to evaluating AI technologies in general and human-computer interaction in particular. For an argument that a computer-based tutorial system can only enhance human systems (like classrooms) that are already quite inhuman anyway, see [Papert 80]. For an entirely different viewpoint, see [Weizenbaum 76]. This section is summarized in Figure 7.1

7.3 Summary and Overview

This section provides a summary of the material covered in this dissertation. This summary is organized in terms of several topical areas. The summary also provides pointers to the chapters and sections in which the relevant material is discussed.

7.3.1 Knowledge and Memory

Broadly, this dissertation is concerned with knowledge and memory. While several researchers have recognized the distinction between semantic and episodic knowledge, few have addressed the interaction between the two. In this dissertation, I address this squarely by focusing on knowledge evolution, i.e., the evolution of semantic knowledge through discrete episodes. This sort of knowledge evolution is illustrated within the context of scientific research.

Distinction between a memory and a knowledge-base is discussed in Section 6.1.4. Semantic-Episodic distinction is discussed in Section 6.1.5. Related work is discussed in Section 6.3.

The domain of scientific research was chosen as a paradigm example of knowledge evolution. Scientific knowledge that is supposed to be timeless 'eternal' truths about the nature of the world is still revealed only through discrete events. Hence I chose this domain to study the episodic nature of essentially semantic knowledge. This domain also presented a set of simple tasks that required reasoning about both the knowledge of the subject and the knowledge of the evolution of the subject.

[10] BIW Perhaps this is a good reason why we need not worry about efficiency with respect to RA-like systems!

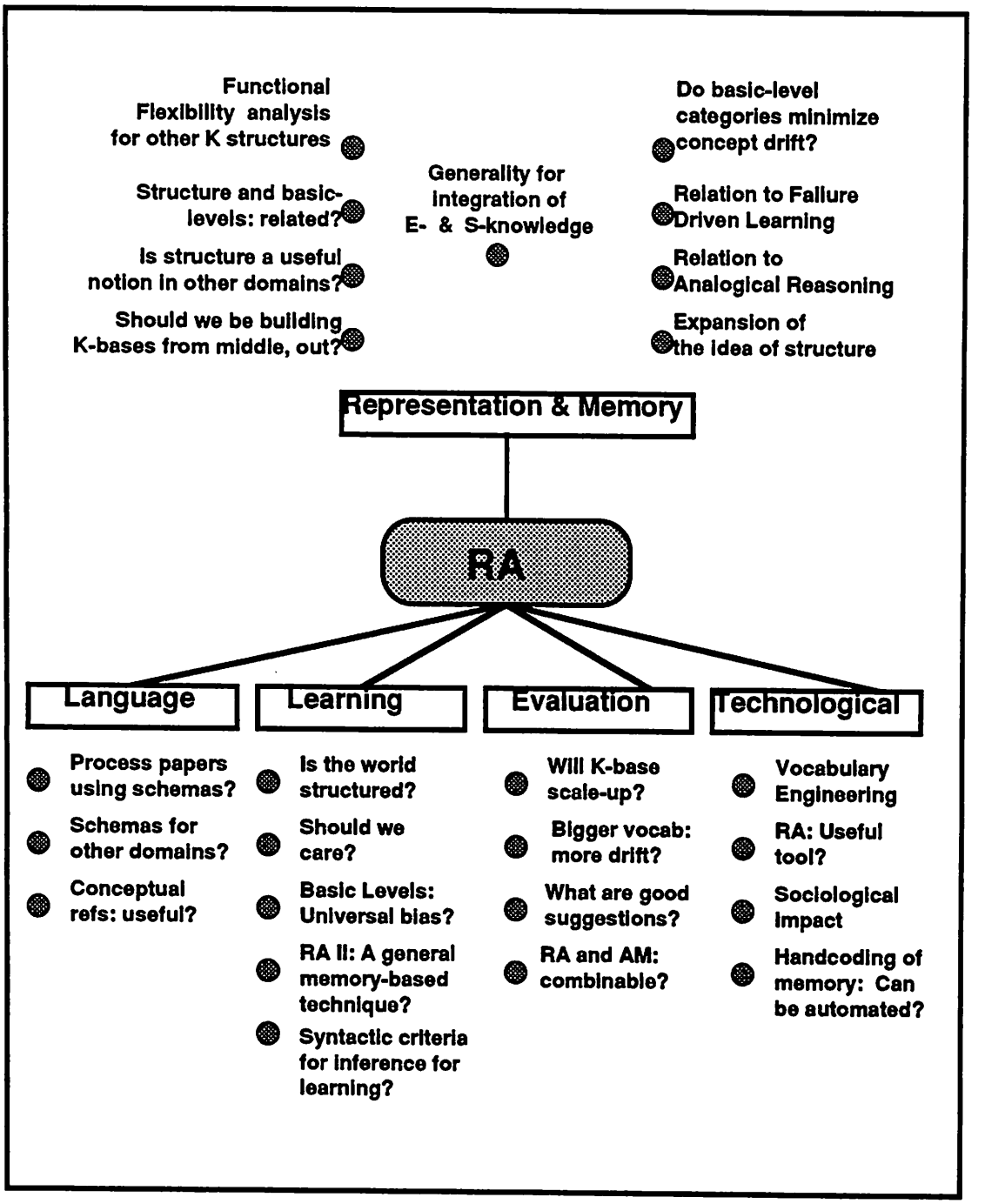


Figure 7.1: Future Directions

Further, scientific research papers reflect this 'knowledge evolution' aspect rather nicely. A paper usually states, through references, what knowledge existed before its time as well as the new knowledge contributed by that paper. Hence, the standard practice of writing research papers by using references to past work has turned out to be a wonderful device for illustrating knowledge evolution.

Motivations for the work are discussed in Sections 1.2 and 1.3.

The research claims of this work are stated below. The first two relate to knowledge and memory, and the third to RA's learning.

- In addition to the deep-semantic level, scientific research is also understood at the structural level.
- In addition to *subject* knowledge, scientific research is also understood in terms of *evolutionary* knowledge.
- Heuristic knowledge is learned by relating a new input to the most specific existing knowledge in memory.

These claims are best seen as ontological level claims since they specify the kinds of knowledge involved in a certain set of behaviors.

Ontological theories are distinguished from epistemological theories in Section 6.1. The claims and why they are interesting are discussed in Section 6.2.

These claims are quite general and are supported by the kinds of tasks typically performed by researchers. For example, when you talk about general scientific methods and heuristics, you are referring to some set of abstract types. Mathematicians normally use types such as sets, functions, and relations; even with mathematics, when you move to vector calculus or differential geometry, you need different kinds of types to state your 'research methods.' With other fields of research, while research methods exist, it is not obvious what types you need to describe your research methods with. The second claim is supported by the observation that researchers can provide historic accounts of their fields when they write surveys and historical reviews.

Support for these claims is provided in Section 6.2.

In this work, I propose a set of structural abstractions (types and relations) to model a scientific field. This is best seen as a content theory. I also propose a particular way to conceptualize research episodes as research schemas. As I pointed out in my discussion on content theories, it is hard to validate content theories. Content theories are meant

not only as a theory of what knowledge is involved in some behavior, but also as a theory of the world. This is precisely why such theories are not evaluable through performance statistics on computer programs. Content theories, in general, have two problems, the problem of criteriality (i.e., the criteria for your choice of primitives), and the problem of coverage (i.e., do these primitives cover all possible instances of the phenomena?). These two are usually in opposition. I have given spirited arguments for my choice of primitives, thereby leaning toward criteriality, perhaps erring in the direction of coverage. Thus, I would claim that any similar conception of a research domain will minimally require these kinds of primitives, but possibly more.

Content theories are discussed in Section 6.1.2. RA's choice of primitives is justified in Section 2.2.2 and reviewed again in Section 6.2.1. Section 7.2 suggests some future work.

The idea of research schemas is validated by the fact that the model of memory suggested by research schemas can support a wide variety of semantic and episodic tasks; further, the same memory model can also support the acquisition of research schemas. I believe that this is a much more substantive evaluation for an ontological theory than a performance evaluation with respect to a single task.

7.3.2 Learning

Despite apparent similarity, RA is different from EBL. In EBL-style learning, one uses general (uninstantiated) semantic knowledge of the domain to explain an example, whereas RA's learning strategy directly accesses (instantiated) knowledge from memory in order to find the relationship between the current state of the field and the new knowledge introduced by a paper. One could see EBL as knowledge-based learning, and RA's learning as memory-based learning. RA's generalization strategy is also different from EBL's. In EBL-style generalization, the domain knowledge provides a basis for generalization. Since RA does not use any domain knowledge in constructing an instantiated schema, it has no apparent basis for generalization.

RA's schema acquisition strategy is described in Chapter 4. Its relation to EBL (and SBL) is discussed in Sections 5.4.4 and 6.3.4.

A close look at RA reveals that RA's world consists of inherently well-differentiated categories. RA generalizes a specific instance into a variable of a category that is both associative and discriminative. This is related to the idea of 'basic-levels' of abstraction that has been demonstrated by Eleanor Rosch and other cognitive psychologists.

Further, the assumptions behind RA's assimilation phase are also explained in terms of associativity and discriminability. These assumptions ensure that the structure of the research schemas is also both associative and discriminative.

Categorization and basic levels are described in Section 5.2 and RA's generalization is related to these in Section 5.3.1. RA's assimilation phase is analyzed in Section 5.3.2.

While several aspects of this analysis are speculative, they raise some interesting questions about knowledge representation, and heuristic generalization.

These speculations are discussed in Section 5.4.

7.3.3 Language

The distinction between semantic and episodic memory gives rise to a corresponding distinction between expository and narrative texts. Episodic relations, in general, are predictive, hence it is possible to compile stereotypical configurations of episodic relations into various kinds of schemas. Such schemas have been used to process narrative texts predictively. In contrast, expository texts are characterized by a preponderance of semantic relations which are, in general, not predictive. In this work, I characterize a large class of expository texts, namely research papers, not as isolated pieces of natural language texts, but as research episodes that are related to each other. This provides a schema-based representation for research papers.

Expository-narrative distinction is discussed in Sections 1.3.2 and 6.1.5.

This begs the question: Can we process research papers using research schemas? Currently, RA's memory is hand-coded. The begging question is: Can we process the text of research papers using a library of skeletal research schemas in order to obtain the (instantiated) schema representation of a paper? As you can see, this is a hard problem. Nevertheless, some of my colleagues considered this problem and came up with an interesting idea called *conceptual references.* This seems like a promising idea to explore.

Conceptual References are discussed in Section 7.2.3

7.3.4 Computer-Aided Research

The idea of Computer-Aided Research is just an interesting idea in order to put all this work within a computational framework. This idea was almost a literal translation of an ideal interaction between a research advisor and a student. While I believe that this might in fact be a practical and useful idea, there are several pragmatic issues that remain to be addressed, particularly, the issue of control (i.e., how to identify interesting suggestions), the issue of building a knowledge base and so on.

The motivation for computer-aided research is discussed in Section 1.1.1. RA's interface and its capabilities are described in Chapter 3. Future work is discussed in Section 7.2.

In passing, I would note that if one were to build a Computer-Aided Research system for a specific field purely as an application of these ideas, certain things might actually become easier. In RA, I was interested in seeing what kinds of knowledge are *minimally* required to achieve certain kinds of behavior. Hence, I was very concerned with some of the theoretical goals: whether my primitives were general, whether I could state the criteria behind their choice, whether the single memory organization could support all of RA's capabilities (without extra frills specific to particular tasks) and so on. RA's capabilities were implemented in order to demonstrate that they require, at a minimum, certain kinds of knowledge, but not to show that RA can do certain wonderful things very well. When these ideas are translated into an application, one can unshackle oneself from this mode of self-deprivation. Hence, in some ways, building a real computer-aided research system might actually be easier.

7.3.5 Scientific Theory Formation

RA follows a long tradition of researchers who have addressed the problem of scientific theory formation. In RA's case, however, the memory organization was treated as a more important issue than theory formation. In particular, RA's suggestion capability is much weaker than AM's discovery capabilities: RA does not 'discover' anything, it only suggests possible research directions. However, RA's approach appears more general than that of AM. Despite Lenat's claims about AM and discovery, I do not believe that AM's approach is applicable to anything like a contemporary research discipline. One reason is that AM requires a strong model of the domain — i.e., it requires what might be called an internally formalizable domain. Even assuming that we can build huge knowledge-bases that can formalize all knowledge pertaining to a research field like EBL, this is not enough. RA shows that, in addition to this kind of deep-semantic knowledge, you also need interesting levels of abstraction for conceptualizing this knowledge in order to discern the research strategies in a field. In terms of building a practical discovery system for a real research field, I would speculate that we should consider a combination of RA-like heuristics to formulate research directions, and AM-like heuristics to actually perform the research.

Related work in theory formation is discussed in Section 6.3. RA's relation to AM is described in Section 6.3.1.

7.3.6 Miscellaneous

The RA system was built to demonstrate, in broad brush strokes, that RA's memory organization can and does support a number of different capabilities. RA works for numerous examples, but has several inconsistencies in its knowledge-base. It needs orders of magnitude more engineering to go past the demo stage.

Based on the experience with this work, I believe that there are several interesting problems in memory organization that need to be addressed. In RA, I considered the interaction between semantic and episodic knowledge in the context of a very stylistic domain. However, it is not clear how these two kinds of knowledge ought to interact in general. It is also not clear as to how general RA's learning scheme is. Is it something very specific to RA and research domains, or is it widely applicable? Finally, the idea of basic level abstractions and their relevance to knowledge representation is very speculative and needs articulation.

Future work is discussed in Section 7.2.

Fifty-seven days. Fifty-seven thousand years. Has mankind changed? Nature has not. And man is nature.

—Thor Heyerdahl, The RA Expeditions.

Appendix A

Research Schemas

This appendix lists several research schemas. Each schema is listed as a **ref def** pair followed by a short description. The schemas that can be derived from the type constraints (e.g., (solves T P)) are not listed here. The relation R denotes any domain-dependent relation (e.g., extends, generalizes, has-less-restrictive-assumptions-than etc). Thus, research schemas involving R stand for a number of different schemas that correspond to different interpretations of R.

Research Schema 1

ref: {(solves T1 P1)}

def: {(solves T2 P1) (R T2 T1)}

For a given technique to solve some problem, propose a new technique that is R-related to the original technique.

Research Schema 2

ref: {(solves T1 P1)}

def: {(entails T1 P2) (solves T2 P2)}

For a given technique to solve some problem, show that the technique has a deficiency. Propose a technique to fix this deficiency.

Research Schema 3

ref: {(solves T1 P1)}

def: {(entails T1 P2) (solves T2 P1)
(not-entails T2 P2)}

For a given technique to solve some problem, show that the technique has a deficiency. Propose a new technique that avoids this deficiency.

Research Schema 4

ref: {(dominates P P1) (dominates P P2)
(solves T1 P1) (instantiates T T1)}

def: {(solves T2 P2) (instantiates T T2)}

If there are two sibling problems, and one of them is solved by some generic technique, show that this technique can also solve the other sibling problem.

Research Schema 5

ref: {(dominates P P1) (dominates P P2)
(solves T1 P1) (solves T2 P2)}

def: {(solves T P)
(instantiates T T2) (instantiates T T1)}

If the children of some problem are all solved by some set of techniques, provide a general framework to solve the parent technique. Show that the individual techniques to solve the children all fit into this framework.

Research Schema 6

ref: {(solves T1 P) (solves T2 P)}

def: {(solves T3 P)
(encapsulates T3 T31) (encapsulates T3 T32)
(instantiates T1 T31) (instantiates T2 T32)}

If there are two different techniques to solve some problem, propose a new technique that is a hybrid of the two techniques to solve the problem.

Research Schema 7

ref: {(dominates P P1) (dominates P P2)
(solves T1 P1) (solves T2 P2)}

def: {(solves T3 P) (encapsulates T3 T31) (instantiates T1 T31)
(encapsulates T3 T32) (instantiates T2 T32)}

If a problem consists of two children and each is solved by a different kind of generic technique, then propose a hybrid technique to solve the parent problem.

Research Schema 8

ref: {(dominates P P1) (dominates P P2)
(solves T1 P1)}

def: {(solves T2 P2) (R T2 T1)}

If one of two siblings is solved by some technique, then propose an R-related technique to solve the other sibling problem.

Research Schema 9

ref: {(dominates P P1) (dominates P P2) (dominates P P3)
(solves T P1) (solves T P2)}

def: {(solves T P3)}

For a technique to solve two sibling problems, use it to solve a third.

Research Schema 10

ref: {(solves T1 P1) (dominates P P1)}

def: {(solves T2 P) (R T2 T1)}

If a problem is solved by a technique, propose an R-related technique to solve the parent problem.

Research Schema 11

ref: {(solves T1 P) (solves T2 P)
(involves T1 C1)}

def: {(involves T2 C2) (R C2 C1)}

If there are two techniques to solve a problem, and one of them involves some concept, propose an R-related concept for the other technique.

Research Schema 12

ref: {(dominates P P1) (dominates P P2)
(involves P1 C1)}

def: {(involves P2 C2) (R C2 C1)}

If there are two sibling problems and one of them involves some concept, propose an R-related concept for the other.

Research Schema 13

ref: {(R P1 P2) (involves P1 C1)}

def: {(involves P2 C2) (R C2 C1)}

If two problems are R-related and one of them involves some concept, propose an R-related concept for the other.

Research Schema 14

ref: {(involves P C1) (involves P C2)}

def: {(R C2 C1)}

If a problem (or technique) involves two concepts, show that these two concepts are R-related.

Research Schema 15

ref: {(solves T1 P) (solves T2 P)
(exhibits T1 Pr1)}

def: {(R Pr2 Pr1)
(exhibits T1 Pr2) (exhibits T2 Pr2)}

If there are two techniques to solve a problem and one of them exhibits a property, propose an R-related property that both techniques exhibit.

Research Schema 16

ref: {(dominates P P1) (dominates P P2)
(exhibits P1 Pr1)}

def: {(R Pr2 Pr1)
(exhibits P1 Pr2) (exhibits P2 Pr2)}

If there are two sibling problems, and one of them exhibits a property, propose an R-related property that applies to both siblings.

Research Schema 17

ref: {(exhibits T Pr) (instantiates T T11)
(encapsulates T1 T11)}

def: {(R Pr T1)}

If a technique is a hybrid technique that encapsulates another technique, and the latter technique exhibits a property, then show that this property is R-related to the hybrid technique.

Research Schema 18

ref: {(entails T P1) (entails T P2)}

def: {(instantiates P1 P12) (instantiates P2 P12)
(solves T12 P12)}

If a technique has two emergent problems, propose a problem that is an instantiation of both these emergent problems. Propose a technique to solve it.

Research Schema 19

ref: {(dominates P P1) (solves T1 P1)
(dominates P P2) (solves T2 P2)
(entails T1 P11)}

def: {(entails T2 P22) (R P22 P11)}

If there are two sibling problems each solved by a technique and one of them has an emergent problem, propose an R-related emergent problem for the other.

Research Schema 20

ref: {(solves T1 P) (solves T2 P)
(entails T1 P1) (entails T2 P2)}

def: {(R P2 P1)}

For two techniques to solve some problems, if each of them has an emergent problem, show that these two emergent problems are R-related.

Research Schema 21

ref: {(dominates P P1) (solves T1 P1)
(entails T1 P11) (solves T11 P11)
(dominates P P2) (solves T2 P2)
(entails T2 P22)
(instantiates T T11)}

def: {(solves T22 P22)
(instantiates T T22)}

If there are two sibling problems each solved by a technique and each of these have an emergent problem each and one of them is solved by some generic technique, solve the other also with the same generic technique.

Research Schema 22

ref: {(solves T P)}

def: {(dominates P P1) (dominates P P2)
(solves T P1) (not-solves T P2)
(not-solves T P) (entails T P2)}

If there is some technique to solve some problem, show that the problem consists of two classes of problems. Show that the technique solves one class but not the other, hence the latter is an emergent problem of the technique.

Research Schema 23

ref: {(encapsulates P P1) (solves T1 P1)
(encapsulates P P2) (solves T2 P2)}

def: {(*solves* T P)
(R T T1) (R T T2)}

If a problem consists of two parts (encapsulations) each solved by separate techniques, propose a general technique that solves the original problem.

Research Schema 24

ref: {(solves T1 P1)}

def: {(solves T2 P2)
(R P2 P1) (R T2 T1)}

If some technique solves some problem, propose an R-related technique to solve an R-related problem.

Research Schema 25

ref: {(exhibits T Pr1) (exhibits T Pr2)}

def: {(exhibits T Pr3)
(R Pr3 Pr1) (R Pr3 Pr2)}

If some technique has two properties, propose a new property that is R-related to both these properties.

Research Schema 26

ref: {(encapsulates P P11) (encapsulates P P22)
(solves T11 P11) (solves T22 P22)
(instantiates T1 T11) (instantiates T2 T22)}

def: {(encapsulates T T11) (encapsulates T T22)
(solves T P)}

If a problem encapsulates two problems each of which are of two generic kinds, and if each of these generic kinds of problems are solved by two generic kinds of techniques, propose a technique that combines these two generic kinds of problems to solve the original problem.

Research Schema 27

ref: {(solves T1 P1) (entails T1 P11)
(dominates P P1) (solves T P)}

def: {(solves T11 P11)
(encapsulates T11 T111) (encapsulates T11 T112)
(instantiates T1 T111) (instantiates T2 T112)}

For a given technique to solve some problem, if the technique has an emergent problem, and if there is a technique to solve the parent problem of the original problem, then combine the two techniques to solve the emergent problem.

Research Schema 28

ref: {(solves T1 P) (solves T2 P)
(exhibits T1 Pr1)}

def: {(R Pr1 T2)}

If a problem is solved by two techniques and if one of them exhibits a property, see how it relates to the other technique.

Research Schema 29

ref: {(dominates performance-problem P1)
(solves T1 P1)}

def: {(acq T1 P2)
(dominates concept-learning-problem P2)
(solves T2 P2)}

If there is a performance problem that is solved by some technique, propose a concept learning problem (or control-learning problem) that emerges from the performance technique. Propose a technique to solve this learning problem.

Research Schema 30

ref: {(dominates performace-problem P1)
(dominates performance-problem P2)
(solves T1 P1) (solves T2 P2)
(acq T1 P11) (solves T11 P11)
(dominates concept-learning-problem T1)}

def: {(acq T2 P22) (solves T22 P22)
(R T22 T11) (dominates concept-learning-problem T22)}

If there are two R-related performance problems that are solved by two techniques, and if there is a technique to solve the (concept or control) learning problem that emerges from the first technique, propose an R-related technique to solve the learning problem that emerges from the second.

Research Schema 31

ref: {(solves T1 P) (solves T2 P) (solves T3 P)
(exhibits T1 Pr) (exhibits T2 Pr)}

def: {(exhibits T3 Pr)}

If there are two techniques to solve some problem, and both of them exhibit a property, show that a third technique also exhibits that property.

This list could provoke conclusions. It was in fact intended to do so.

—Thor Heyerdahl, The RA Expeditions.

Appendix B

Example Summary

Section 3.5 described RA's chronological summarization component. This appendix gives an example of a chronological summary that involves tracing through multiple **refs**. First I introduce a sequence of papers and their research schemas in order to illustrate a paper whose **ref** contains relations defined by several prior papers.

In Chapter 2, we saw the schema of [Mitchell et al 86]. In addition to providing a general framework for EBL, this paper also defined three emergent problems of EBL: incomplete theory problem, intractable theory problem, and inconsistent theory problem. The following is the research schema of this paper.

ref: {(dominates learning-problem concept-learning-problem)
(dominates learning-problem control-learning-problem)
(instantiates concept-learning-problem schema-acquisition-problem)
(instantiates control-learning-problem LP-control-learning-problem)
(solves explanatory-schema-acquisition schema-acquisition-problem)
(solves LP-learning-technique LP-control-learning-problem)}

def: {(solves EBL learning-problem)
(instantiates EBL explanatory-schema-acquisition)
(instantiates EBL LP-learning-technique)
(entails EBL incomplete-theory-problem)
(entails EBL intractable-theory-problem)
(entails EBL inconsistent-theory-problem)}

Lebowitz's paper [Lebowitz 86] proposed that for problems that involve a large domain theory, one needs some way of controlling the inferences generated by a learning system while constructing an explanation. Lebowitz proposed the notion of 'interestingness' as a mechanism to control inferences in such a theory. One schema for this paper is the following:

ref: {(entails EBL intractable-theory-problem)}

def: {(solves interestingness intractable-theory-problem)}

Tadepalli's paper [Tadepalli 85] also considers intractable theory problems such as chess. Even though the rules of chess constitute a correct and complete theory of chess, the combinatorics of the domain make it intractable. Tadepalli proposes that, for such theories, one needs a smaller and hence an approximate theory of the domain that is more tractable[1]. The schema of this paper is shown below:

ref: {(entails EBL intractable-theory-problem)}

def: {(solve approximation intractable-theory-problem)}

In a 1988 paper, I considered the problem of epidemiological diagnosis [Swaminathan 88a] This diagnosis problem may be seen as a concept learning problem — the diagnosis identifies the set of features that is common to all victims of an epidemic, which is the same as learning the concept "epidemic victim." The problem was shown to be both an incomplete- and an intractable-theory problem. The diagnostic technique I proposed used a hybrid technique combining EBL and SBL to solve the incomplete-theory aspect of the problem, and a combination of the ideas of an approximate theory and interestingness to solve the intractable theory aspect of the problem. The schema of this paper is shown below:

ref: {(entails EBL incomplete-theory-problem)
(entails EBL intractable-theory-problem)
(solves approximation intractable-theory-problem)
(solves interestingness intractable-theory-problem)
(solves EBL learning-problem)
(solves SBL learning-problem)}

def: {(instantiates incomplete-theory-problem epidemiology-problem)
(instantiates intractable-theory-problem epidemiology-problem)
(solves diagnostic-technique epidemiology-problem)
(encapsulates diagnostic-technique component1)
(encapsulates diagnostic-technique component2)
(encapsulates diagnostic-technique component3)
(instantiates SBL component1)
(instantiates approximation component2)
(instantiates interestingness component3)}

[1]Some readers may be surprised by a 1985 paper by Tadepalli referencing a 1986 paper by Mitchell et al. Tadepalli attributes the intractable theory problem to Mitchell without explicitly referencing any papers.

Let's now assume that RA is asked to generate a chronological summary of this paper. Tracing the **ref** from this paper, we arrive at [Mitchell et al 86], [Tadepalli 85], [Lebowitz 86], and [Mitchell 83][2]. Tracing the **ref** from [Tadepalli 85] and [Lebowitz 86], we arrive at [Mitchell et al 86]. Since we now have a common root, learning-problem, and all **refs** have been covered, we generate the following summary:

[MITCHELL-83] showed: SBL solves LEARNING-PROBLEM.
[MITCHELL-86] showed: EBL solves LEARNING-PROBLEM.
EBL entails INCOMPLETE-THEORY-PROBLEM.
EBL entails INTRACTABLE-THEORY-PROBLEM.
[LEBOWITZ-86] showed: INTERESTINGNESS solves
INTRACTABLE-THEORY-PROBLEM.
[TADEPALLI-85] showed: APPROXIMATION solves
INTRACTABLE-THEORY-PROBLEM.
[SWAMINATHAN-88] showed: INCOMPLETE-THEORY-PROBLEM
instantiates EPIDEMIOLOGY-PROBLEM.
INTRACTABLE-THEORY-PROBLEM
instantiaties EPIDEMIOLOGY-PROBLEM.
DIAGNOSTIC-TECHNIQUE solves EPIDEMIOLOGY-PROBLEM.
DIAGNOSTIC-TECHNIQUE encapsulates COMPONENT1, COMPONENT2, COMPONENT3.
SBL instantiates COMPONENT1.
APPROXIMATION instantiates COMPONENT2.
INTERESTINGNESS instantiates COMPONENT3.

In English, this summary is equivalent to the following: [Mitchell 83] proposed SBL to solve the learning problem and [Mitchell 86] proposed EBL to solve the learning problem, and showed that EBL has two emergent problems called incomplete theory problem and intractable theory problem. There are two different solutions to the intractable theory problems: the technique called interestingness is due to [Lebowitz 86] and the technique called approximation is due to [Tadepalli 85]. [Swaminathan 88a] considers a problem that involves both an incomplete theory and an intractable theory and proposes a technique that combines SBL, approximation, and interestingness.

[2]There is no obvious paper to attribute the relation (solves SBL learning-problem) to. Let's assume that this is attributable to Mitchell's *Computer and Thought* award lecture [Mitchell 83].

Bibliography

[Abelson & Sussman 85] Abelson, Harold, and Sussman, Gerald J., *Structure and Interpretation of Computer Programs*, MIT Press, Cambridge, MA, 1985.

[Adelson 85] Adelson, Beth, "Comparing Natural and Abstract Categories: A Case-Study From Computer Science," *Cognitive Science*, 9: 417–430, 1985.

[Agre 88] Agre, Philip E., "The Dynamic Structure of Everyday Life," Ph.D Dissertation, Techical Report 1085, MIT AI Laboratory, Cambridge, MA, 1988.

[Aha & Kibler 89] Aha D., and Kibler D., "Noise Tolerant Instance-based Learning," Proc. of the International Joint Conference on AI, Detroit, MI, 1989.

[Ashley & Rissland 88] Ashley, Kevin D., and Rissland, Edwina R., "Waiting on Weighting: A Symbolic Least Commitment Approach," Proc. of the National Conference on AI St. Paul, MN, 1988.

[Ashley 88] Ashley, Kevin, "Modelling Legal Argument: Reasoning With Cases and Hypotheticals," Ph.D Dissertation, University of Massachusetts, Amherst, MA, 1988.

[Bareiss 89] Bareiss, Ray, *Exemplar-Based Knowledge Acquisition*, Academic Press, New York, NY, 1989.

[Barr & Feigenbaum 81] Barr, Avron, and Feigenbaum, Edward, *The Handbook of Artificial Intelligence, Volume 1*, Addison Wesley, Reading, MA, 1981.

[Bartlett 32] Bartlett, Frederick C., *Remembering*, Cambridge University Press, Cambridge, England, 1932.

[Berlin 78] Berlin, Brent, "Ethnobiological Classification," in Rosch E. and Lloyd B., (eds) *Cognition and Catgeorization*, Lawrence Erlbaum, Hillsdale, NJ, 1978.

[Birnbaum & Selfridge 81] Birnbaum, Lawrence, and Selfridge, Mallory M., "Conceptual Analysis of Natural Language," in Schank R.C., and Riesbeck C.K., (eds) *Inside Computer Understanding*, Lawrence Erlbaum, Hillsdale, NJ, 1981.

[Bobrow & Winograd 77] Bobrow, Daniel G., and Winograd, Terry, "An Overview of KRL, A Knowledge Representation Language," *Cognitive Science*, 1: 3–46, 1977.

[Brachman & Levesque 85] Brachman, Ronald J., and Levesque, Hector J., *Readings in Knowledge Representation,* Morgan Kaufman, Los Altos, CA, 1985.

[Brachman & Schmolze 85] Brachman R.J., and Schmolze J.G., "An Overview of the KL-ONE Knowledge Representation System," *Cognitive Science* 9: 171–216, 1985.

[Brachman & Smith 80] Brachman R.J., and Smith B.C., "Special Issue on Knowledge Representation," *SIGART Newsletter*, 70: 5-25, 1980.

[Brachman 79] Brachman, Ronald J., "On the Epistemological Status of Semantic Networks," in Findler N.V., (ed) *Associative Networks: Representation and Use of Knowledge by Computers*, Academic Press, New York, NY, 1979. (Reprinted in [Brachman & Levesque 85]).

[Britton & Black 85] Britton, Bruce, and Black, John, "Understanding Expository Text: From Structure to Process and World Knowledge," in Britton B., and Black J., (eds.) *Understanding Expository Text*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1985.

[Brown 58] Brown, Roger, "How Shall A Thing Be Called?," *Psychological Review*, 65: 14–21, 1958.

[Bundy & Welham 81] Bundy, A., and Welham, B., "Using Meta-level Inference for Selective Application of Rewrite Rules in Algebraic Manipulation," *Artificial Intelligence*, 16: 189–212, 1981.

[Carbonell et al 83] Carbonell, Jaime G., Mitchell, Thomas M., and Michalski, Ryzard S., "An Overview of Machine Learning," in Michalski R.S., Carbonell J.G., and Mitchell T.M., (eds) *Machine Learning: An Artificial Intelligence Approach*, Morgan Kaufmann, Los Altos, CA, 1983.

[Charniak 72] Charniak, Eugene C., "Toward a Model of Children's Story Comprehension," Ph.D Dissertation, Massachusetts Institute of Technology, Cambridge, MA, 1972.

[Charniak 86] Charniak, Eugene C., "A Neat Theory of Marker Passing," Proc. of the National Conference on AI, Philadelphia, PA, 1986.

[Clancey 85] Clancey, William J., "Representing Control Knowledge As Abstract Tasks and Metarules," Technical Report KSL 85-16, Computer Science Department, Stanford University, Stanford, CA, 1985.

[Clancey 88] Clancey, William J., "Detecting and Coping with Failure," Proc. of the Spring Symposium on Explanation-Based Learning, Stanford, CA, 1988.

[Clancey 89] Clancey, William J., "The Frame of Reference Problem in Cognitive Modeling," Proc. of the Cognitive Science Conference, Ann Arbor, MI, 1989.

[Cognitive Systems 90] "Case-Based Retrieval Shell," User's Manual, Cognitive Systems, New Haven, CT, 1990.

[Cohen & Stanhope 86] Cohen, Paul R., and Stanhope, Philip, "Finding Research Funds with the GRANT System," Proc. of the Sixth International Workshop on Expert Systems and Their Applications, Avignon, France, 1986.

[Cohen et al 88a] Cohen, Paul R., DeLisio, Jefferson L., and Hart, David, "A Declarative Representation of Control Knowledge," *IEEE Transactions on Systems, Man and Cybernetics*, 2: 17–29, 1988.

[Cohen et al 88b] Cohen, William, Mostow, Jack, and Borgida, Alex, "Generalizing Number in Explanation-based Learning," Proc. of the Spring Symposium on Explanation-Based Learning, Stanford, CA, 1988.

[Cohen et al 89] Cohen, Paul R, Greenberg, Michael L., Hart, David M., and Howe, Adele E., "Trial by Fire: Understanding the Design Requirements for Agents in Complex Environments," *AI Magazine*, 10: 34–48, (Fall) 1989.

[Collins & Birnbaum 88] Collins, Gregg, and Birnbaum, Lawrence, "An Explanation-Based Approach to the Transfer of Planning Knowledge Across Domains," Proc. of the AAAI Spring Symposium on Case-Based Reasoning, Stanford, CA, 1990.

[Collins & Quillian 72] Collins, Allan M., and Qullian, R.M., "How To Make a Language User," in Tulving E., and Donaldson W., (eds) *Organization of Memory*, Academic Press, New York, NY, 1972.

[Corkill et al 82] Corkill, Daniel D., Lesser, Victor R., and Hudlicka, Eva, "Unifying Data-directed and Goal Directed Control: An Example and Experiments," Proc. of the National Conference on AI, Pittsburgh, PA, 1982.

[Cruse 77] Cruse D.A., "The Pragmatics of Lexical Specificity," *Journal of Linguistics*, 13: 153–64, 1977.

[Cullingford 78] Cullingford, Richard E., "Script Application: Computer Understanding of Newspaper Stories," Technical Report No. 16, Computer Science Department, Yale University, New Haven, CT, 1978.

[Davis & King 77] Davis, Randall and King, Jonathan, "An Overview of Production Systems," *Machine Intelligence*, 8: 300–332, 1977.

[Davis et al 77] Davis, Randall, Buchanan, Bruce, and Shortliffe, Edward, "Production Rules as a Representation for a Knowledge-Based Consultation Program," *Artificial Intelligence*, 8: 15–45, 1977.

[DeJong & Mooney 86] DeJong, Gerald F., and Mooney, Raymond, "Explanation-Based Learning: An Alternative View," *Machine Learning*, 1: 145–176, 1986.

[DeJong 79] DeJong, Gerald F., "Skimming Stories in Real Time: An Experiment in Integrated Understanding," Technical Report No. 158, Yale University, New Haven, CT, 1979.

[DeJong 83] DeJong, Gerry F., "Acquiring Schemata through Understanding and Generalizing Plans," Proc. of the International Joint Conference on AI, Karlsruhe, Germany, 1983.

[deKleer et al 77] deKleer J., Doyle J., Steele, G., and Sussman G., "Amord: Explicit Control of Reasoning," *SIGART Newsletter*, 64: 116–125, 1977.

[Dietterich 86] Dietterich, Thomas, "Learning at the Knowledge Level," *Machine Learning*, 1: 287–315, 1986.

[Dijkstra 68] Dijkstra, Edsgar, "Goto Considered Harmful," *Communications of the ACM*, 11: 147–148, 1968.

[Dyer 83] Dyer, Michael, 1983. *In-Depth Understanding: A Computer Model of Integrated Processing for Narrative Comprehension,* MIT Press, Cambridge, MA, 1983.

[Ellman 89] Ellman, Thomas, "Explanation-Based Learning: A Survey of Programs and Perspectives," *ACM Computing Surveys*, 21: 163–222, 1989.

[Fikes et al 72] Fikes R.E.,"Learning and Executing Generalized Robot Plans," *Artificial Intelligence,* 3: 251–288, 1972.

[Fillmore 68] Fillmore, Charles, "The Case for Case," in Bach E., and Harms R., (eds) *Universals in Linguistic Theory,* Hold, Reinhart & Winston, New York, NY, 1968.

[Fisher & Langley 90] Fisher, Douglas, and Langley, Patrick, "The Structure and Formation of Natural Categories," Technical Report CS-90-05, Department of Computer Science, Vanderbilt University, Nashville, TN 1990.

[Fisher 86] Fisher, Douglas H., "Conceptual Clustering, Learning from Examples, and Inference," Proc. of the Fourth International Workshop on Machine Learning, Irvine, CA, 1986.

[Fisher et al 90] Fisher D., Yoo J., and Yang H., "Case-based and Abstraction-based Reasoning," Proc. of the AAAI Spring Symposium on Case-Based Reasoning, Stanford, CA, 1990.

[Fodor 83] Fodor, Jerry A., *The Modularity of Mind*, Bradford Press, Cambridge, MA, 1983.

[Fodor 87] Fodor, Jerry A., "Modules, Frames, Fridgeons, Sleeping Dogs, and the Music of the Spheres," in Garfield, Jay L., (ed) *Modularity in Knowledge Representation and Natural Language Understanding,* MIT Press, Cambridge, MA, 1987.

[Fricke 88] Fricke, Hans, "Coelacanths: The Fish That Time Forgot," *National Geographic*, 173: 824–838, 1988.

[Galambos et al 86] Galambos, James A., Abelson Robert P., and Black, John B., *Knowledge Structures*, Lawrence Erlbaum, Hillsdale, NJ, 1986.

[Gardner 85] Gardner, Howard, *The Mind's New Science,* Basic Books, New York, NY, 1985.

[Garfield 64] Garfield E., "Can Citation Indexing Be Automated?," Proc. of the Symposium on Statistical Association Methods for Mechanized Documentation, Washington D.C., 1964.

[Garfield 79] Garfield E., *Citation Indexing — Its Theory and Applications in Science, Technology and Humanities,* John Wiley, New York, NY, 1979.

[Gentner 83] "Structure-Mapping: A Theoretical Framework For Analogy," *Cognitive Science,* 7: 155–170, 1983.

[Gershman 79] Gershman, Anatole V., "Knowledge Based Parsing," Ph.D Dissertation, Research Report No. 156, Computer Science Department, Yale University, New Haven, CT, 1979.

[Gluck & Corter 85] Gluck, M.A., and Corter, J.E., "Information, Uncertainty, and the Utility of Catgeories," Proc. of the Cognitive Science Conference, Irvine, CA, 1985.

[God 08] God, My O, "The Genesis of Machine Learning," Technical Report No. TR1, Heaven, Above, 0008.

[Gould 83] Gould, Stephen J., "What If Anything Is A Zebra?," reproduced in Gould S.J., (ed) *Hen's Teeth and Horse's Toes,* Norton Press, New York, NY, 1983.

[Groner et al 83] Groner, Marina, Groner, Rudolf, and Bischof, Walter, "Approaches to Heuristics: A Historical Review," in Groner R., Groner M., and Bishof W., (eds) *Methods of Heuristics,* Lawrence Erlbaum, Hillsdale, NJ, 1983.

[Hadamard 54] Hadamard, J., *The Psychology of Invention in the Mathematical Field,* Dover Publishers, New York, NY, 1954.

[Hall 89] Hall, Rogers P., "Computational Approaches to Analogy," *Artificial Intelligence,* 39: 39–120, 1989.

[Hammond 89] Hammond, Kristian J., *Case-Based Planning: Viewing Planning as a Memory Task,* Academic Press, San Diego, CA, 1989.

[Hayes 79] Hayes, Patrick J., "The Logic of Frames," in Metzing D., (ed) *Frame Conceptions and Text Understanding,* Walter de Gruyter and Co., Berlin, Germany, 1979. (Reprinted in [Brachman & Levesque 85]).

[Hirsh 88] Hirsh, Haym, "Empirical Techniques for Repairing Imperfect Theories," Proc. of the AAAI Spring Symposium on Explanation-Based Learning, Stanford, CA, 1988.

[Hunter 88] Hunter, Lawrence, "Explanation Based Discovery," Proc. of the AAAI Spring Symposium on Explanation-Based Learning, Stanford, CA, 1988.

[Jacobs 86] Jacobs, Paul S., "Language Analysis in Not-so-limited Domains," Proc. of the Fall Joint Computer Conference, Dallas, TX, 1986.

[Joseph 90] Joseph, Lawrence E., *Gaia: The Growth of an Idea*, St. Martin's Press, New York, NY, 1990.

[Kass & Owens 88] Kass, Alex, and Owens, Christopher C., "Learning New Explanations by Incremental Adaptation," Proc. of the AAAI Spring Symposium on Explanation-Based Learning, Stanford, CA, 1988.

[Keller 87] Keller, Richard M., "The Role of Explicit Contextual Knowledge in Learning Concepts To Improve Performance," Ph.D Dissertation, Rutgers University, New Brunswick, NJ, 1987.

[Keller 88] Keller, Richard, "Operationality and Generality in Explanation-Based Learning: Separate Dimensions or Opposite Endpoints?," Proc. of the AAAI Spring Symposium on Explanation-Based Learning, Stanford, CA, 1988.

[Kintch et al 84] Kintsch, Walter, Miller, James R., and Polson, Peter G., *Method and Tactics in Cognitive Science,* Lawrence Erlbaum, Hillsdale, NJ, 1984.

[Knuth 84] Knuth, Donald E., *The TEXbook*, Addison-Wesley, Reading, MA, 1984.

[Kolodner 84] Kolodner, Janet L., *Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model*, Lawrence Erlbaum, Hillsdale, NJ, 1984.

[Kolodner 88a] Kolodner, Janet L., "Background Reading," Proc. of the Case-Based Reasoning Workshop, Clearwater, FL, 1988.

[Kolodner 88b] Kolodner, Janet L., "Retrieving Events from a Case Memory: A Parallel Implementation," Proc. of the Case-Based Reasoning Workshop, Clearwater, FL, 1988.

[Kuhn 70] Kuhn, Thomas, *The Structure of Scientific Revolutions,* 2nd edition, The University of Chicago Press, Chicago, IL, 1970.

[Laird et al 87] Laird, J.E., and Newell A.,and Rosenbloom P.S., "SOAR: An Architecture for General Intelligence," *Artificial Intelligence*, 33: 1–64, 1987.

[Lakatos 76] Lakatos, Imre, *Proofs and Refutations*, Cambridge University Press, Cambridge, England, 1976.

[Lakoff & Johnson 80] Lakoff, George, and Johnson, Mark, *Metaphors We Live By*, University of Chicago Press, Chicago, IL, 1980.

[Lakoff 87] Lakoff, George, *Women, Fire and Dangerous Things,* The University of Chicago Press, Chicago, IL, 1987.

[Lamport 86] Lamport, Leslie, *LATEX: A Document Preparation System,* Addison-Wesley, Reading, MA, 1986.

[Langley 86] Langley, Patrick, "Human and Machine Learning," Editorial, *Machine Learning,* 1: 243–248, 1986.

[Langley et al 83] Langley P., Bradshaw G.L., and Simon H.A., "Rediscovering Chemistry with the Bacon System," in Michalski R.S., Carbonell J.G., and Mitchell T.M., (eds) *Machine Learning: An Artificial Intelligence Approach*, Tioga, Palo Alto, CA, 1983.

[Leake 88] Leake, David B., "Using Explainer Needs To Judge Operationality," Proc. of the AAAI Spring Symposium on Explanation-Based Learning, Stanford, CA, 1988.

[Lebowitz 83] Lebowitz, Michael M., "Memory-Based Parsing," *Artificial Intelligence*, 21: 363–404, 1983.

[Lebowitz 86] Lebowitz, Michael, "Integrated Learning: Controlling Explanation," *Cognitive Science,* 10: 219–240, 1986.

[Lebowitz 86] Lebowitz, Michael M., "Concept Learning in a Rich Input Domain: Generalization-Based Memory," in Michalski R.S., Corbonell J.G., and Mitchell T.M., (eds) *Machine Learning: An Artificial Intelligence Approach, Vol 2,* Morgan Kaufman, Los Altos, CA, 1986.

[Lehnert 78] Lehnert, Wendy G., *The Process of Question Answering,* Lawrence Erlbaum Associates, Hillsdale, NJ, 1978.

[Lehnert 81] Lehnert, Wendy G., "Plot Units and Narrative Summarization," *Cognitive Science*, 5: 293–331, 1981.

[Lehnert 87a] Lehnert, Wendy G., "Case-Based Problem Solving with a Large Knowledge Base of Learned Cases," Proc. of the National Conference on AI, Seattle, WA, 1987.

[Lehnert 87b] Lehnert, Wendy G., "Case-Based Reasoning as a Paradigm for Heuristic Search," Technical Report, University of Massachusetts, Amherst, MA, 1987.

[Lehnert 88a] Lehnert, Wendy G., "Knowledge-based Natural Language Understanding," in Shrobe H.E., (ed) *Exploring Artificial Intelligence*, Morgan Kaufmann, San Mateo, CA, 1988.

[Lehnert 88b] Lehnert, Wendy G., "Symbolic/Subsymbolic Sentence Analysis: Exploiting the Best of Two Worlds," in Barnden J., and Pollack J., (eds) *Advances in Connectionist and Neural Computation Theory, Vol. 1*, Ablex Publishers, in press. (Also available as COINS Technical Report 88-99, Department of Computer and Information Science, University of Massachusetts, Amherst, MA, 1988.)

[Lehnert et al 83] Lehnert, Wendy G., Dyer M., Johnson P., Yang C., and Harley S., "BORIS — An Experiment in In-depth Understanding of Narratives," *Artificial Intelligence*, 20: 15–62. 1983.

[Lehnert et al 90] Lehnert, Wendy G., Cardie, Claire, Riloff, Ellen M., "Analyzing Research Papers Using Citation Sentences," Proc. of the Cognitive Science Conference, Cambridge, MA, 1990.

[Lenat & Brown 84] Lenat, Douglas B., and Brown, John S., "Why AM and Eurisko Appear To Work," *Artificial Intelligence*, 23: 269–294, 1984.

[Lenat & Guha 89] "The World According To CYC," MCC Technical Report, MCC, Austin, TX, 1989.

[Lenat 76] Lenat, Douglas B., "AM: An Artificial Intelligence Approach to Discovery in Mathematics as Heuristic Search," Ph.D Dissertation, Stanford University, Stanford, CA, 1976.

[Lenat 82] Lenat, Douglas B., "Learning By Discovery: Three Case Studies in Natural and Artificial Learning Systems," in Michalski R.S., Mitchell T.M., and Carbonell J.G., (eds) *Machine Learning: An Artificial Intelligence Approach*, Tioga, Palo Alto, CA, 1982.

[Lenat 83] Lenat, Douglas B., "Eurisko: A Program That Learns New Heuristics and Domain Concepts: The Nature of Heuristics III: Program Design and Results," *Artificial Intelligence*, 21: 61–98, 1983.

[Lewis et al 89] Lewis, David D., Croft, Bruce W., and Bhandaru, Nehru, "Language-Oriented Information Retrieval," *International Journal of Intelligent Systems*, 4: 285–318, 1989.

[Loftus 75] Loftus, Elizabeth F., "Leading Questions and the Eyewitness Report," *Cognitive Psychology*, 7: 560–572, 1975.

[Loiselle & Cohen 89] Loiselle, Cynthia L., and Cohen, Paul R., "Explorations in the Contributors to Plausibility," Proc. of the Cognitive Science Conferenence, Ann Arbor, MI, 1989.

[Lovelock 87] Lovelock, James, "Gaia: A Model for Planetary and Cellular Dynamics," in Thompson, W.I., (ed) *Gaia: A New Way of Knowing*, Lindisfarne Press, Great Barrington, MA, 1987.

[Lytinen 86] Lytinen, Steven L., "Dynamically Combinining Syntax and Semantics in Natural Language Processing," Proc. of the National Conference on AI, Philadelphia, PA, 1986.

[Mahadevan et al 88] Mahadevan, Sridhar, Natarajan, B.K., and Tadepalli, Prasad, "A Framework for Learning as Improving Problem-solving Performance," Proc. of the AAAI Spring Symposium on Explanation-Based Learning, Stanford, CA, 1988.

[Marr 82] Marr, David., *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* W.H. Freeman, San Francisco, CA, 1982.

[Martin 86] Martin, James H., "Polysemy," Proc. of the Workshop on Theoretical Issues in Conceptual Information Processing, Philadelphia, PA, 1986.

[Mayr 84] Mayr, Ernst, "Species Concepts and Their Applications," in Sober E., (ed) *Conceptual Issues in Evolutionary Biology*, MIT Press, Cambridge, MA, 1984.

[McCarthy 80] McCarthy, John, "Circumscription — A Form of Non-monotonic Reasoning," *Artificial Intelligence*, 13: 27–39, 1980.

[McCulloch & Pitts 43] McCulloch, W., and Pitts, W. "A Logical Calculus of the Ideas Immanent in Nervous Activity," *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

[McDermott 76] McDermott, Drew, "Artificial Intelligence Meets Natural Stupidity," *SIGART Newsletter*, 57: 4-12, 1976.

[Michalski et al 83] Michalski, Ryzard S., Carbonell, Jaime G., and Mitchell, Thomas M., *Machine Learning: An Artificial Intelligence Approach*, Morgan Kaufmann, Los Altos, CA, 1983.

[Michener 77] Michener, Edwina R., "Epistemology, Representation, Understanding and Interactive Exploration of Mathematical Theories," Ph. D. Dissertation, MIT, Cambridge, MA, 1977.

[Michener 78] Michener, Edwina R., "Understanding Understanding Mathemetics," *Cognitive Science*, 2: 361–383, 1978.

[Minsky & Papert 69] Minsky, Marvin, and Papert, Samuel, *Perceptrons*, MIT Press, Cambridge, MA, 1969.

[Minsky 75] Minsky, Marvin, "A Framework for Representing Knowledge," in Winston P.H., (ed) *The Psychology of Computer Vision*, MGraw Hill, New York, NY, 1975.

[Minton 88] Minton, Steven, "Learning Effective Search Control Knowledge: An Explanation-Based Approach," Ph.D Dissertation, CMU-CS-88-135, Computer Science Department, Carnegie-Mellon University, Pittsburgh, PA, 1988.

[Mitchell 78] Mitchell, Thomas M., "Version Spaces: An Approach to Concept Learning," Ph.D. Dissertation, Stanford University, Stanford, CA, 1978.

[Mitchell 80] Mitchell, Thomas M., "The Need for Biases in Learning Generalizations," Technical Report CBM-TR-117, Department of Computer Science, Rutgers University, New Brunswick, NJ, 1980.

[Mitchell 81] Mitchell, Thomas M., "Generalization as Search," in Webber B.L., and Nilsson N.J., (eds) *Readings in Artificial Intelligence* Morgan Kaufmann, Los Altos, CA, 1981. (also appears in *Artificial Intelligence*, 18: 203–336, 1982.)

[Mitchell 83] Mitchell, Thomas M., "Learning and Problem Solving," Proc. of the International Joint Conference on AI, Karlsruhe, Germany, 1983.

[Mitchell et al 83] Mitchell T. M., Utgoff P.E., and Banerji R.B., "Learning by Experimentation: Acquiring and Refining Problem-Solving Heuristics," in Michalski R.S., Corbonell J.G., and Mitchell T.M., (eds) *Machine Learning: An Artificial Intelligence Approach*, Morgan Kaufmann, Los Altos, CA, 1983.

[Mitchell et al 86] Mitchell, Thomas M., Keller, Richard M., and Kedar-Cabelli, Smadar, "Explanation-Based Generalization: A Unifying View," *Machine Learning*, 1: 47–80, 1986.

[Mooney & Bennett 86] Mooney, Raymond, and Bennett, Scott, "A Domain Independent Explanation-Based Generalizer," Proc. of the National Conference on AI, Philadelphia, PA, 1986.

[Mooney & DeJong 85] Mooney, Raymond, and DeJong, Gerald, "Learning Schemata for Natural Language Processing," Proc. of the International Joint Conference on AI, Los Angeles, CA, 1985.

[Mooney 88] Mooney, Raymond, "A General Explanation-Based Learning Mechanism and Its Application to Narrative Understanding," Ph.D Dissertation, University of Illinois, Urbana, IL, 1988.

[Mostow 87] Mostow, Jack, "Searching for Operational Concept Descriptions in BAR, MetaLEX, and EBG," Proc. of the Machine Learning Conference, Irvine, CA, 1987.

[Murphy & Medin 85] Murphy, Gregory L., and Medin, Douglas L., "The Role of Theories in Conceptual Coherence," *Psychological Review*, 92: 289–316, 1985.

[Newell 82] Newell, Allen, "The Knowledge Level," *Artificial Intelligence*, 18: 87–127, 1982.

[Newport & Bellugi 78] Newport, Elissa L, and Bellugi, Ursula, "Linguistic Expression of Category Levels in a Visual-gestural Language: A Flower Is a Flower Is a Flower," in Rosch E., and Lloyd B.,(eds) *Cognition and Catgeorization*, Lawrence Erlbaum, Hillsdale, NJ, 1978.

[Norvig 87] Norvig, Peter, "Unified Theory of Inference for Text Understanding," Ph.D Dissertation, University of California, Berkeley, CA, 1987.

[Ortony 79] Ortony, A., *Metaphor and Thought*, Cambridge University Press, London, England, 1979.

[Oshersen et al 90] Oshersen D.N., Smith E.E., Wilkie O., Lopez A., and Shafir E., "Category-Based Induction," *Psychological Review*, 97: 185–200, 1990.

[Papert 80] Papert, Seymour, *Mindstorms*, Basic Books, New York, NY, 1980.

[Peters & Rapaport 90] Peters, Sandra, and Rapaport, William, "Superordinate and Basic Level Categories in Discourse," Proc. of the Cognitive Science Conference, Cambridge, MA, 1990.

[Polya 57] Polya, G., *How To Solve It*, 2nd edition, Doubleday Anchor Books, Garden City, NY, 1957.

[Prieditis et al 87] Prieditis, Armand E., et al., "AAAI-86 Learning Papers: Developments and Summaries," *Machine Learning*, 2: 83–95, 1987.

[Puff 77] Puff, Richard C., *Memory Organization and Structure*, Academic Press, New York, NY, 1977.

[Quillian 67] Quillian, Ross M., "Word Concepts: A Theory and Simulation of Some Basic Semantic Capabilities," *Behavioral Science*, 12: 410–430, 1967. (Reprinted in [Brachman & Levesque 85]).

[Rajamoney 88] Rajamoney, Shankar, "Experimentation-Based Theory Revision," Proc. of the AAAI Spring Symposium on Explanation-Based Learning, Stanford, CA, 1988.

[Reddy 88] Reddy, Raj, "Foundations and Grand Challenges of Artificial Intelligence," *AI Magazine*, 9:4, 9–24, 1988.

[Reiff & Schreerer 59] Reiff, R., and Schreerer, M., *Memory and Hypnotic Age Regression*, International Universities Press, New York, NY, 1959.

[Reimer 87] Reimer U., and Hahn U., "Text Condensation as Knowlede Base Abstraction," Proc. of the IEEE Conference on AI Applications, San Diego, CA, 1988.

[Riesbeck & Martin 85] Riesbeck, C. and Martin C., "Direct Memory Access Parsing," Research Report 354, Yale University, New Haven, CT, 1985.

[Riesbeck 75] Riesbeck, Christopher C., "Conceptual Analysis," in Schank R.C., (ed) *Conceptual Information Processing,* North Holland, Amsterdam, 1975.

[Rissland 87] Rissland, Edwina L., "Research Initiative in Case-Based Reasoning," Councelor Project Technical Memo, Computer and Information Science Department, University of Massachusetts, Amherst, MA 1987.

[Rosch & Lloyd 78] Rosch, Eleanor, and Lloyd, Barbara, *Cognition and Categorization*, Lawrence Erlbaum, Hillsdale, NJ, 1978.

[Rosch & Mervis 75] Rosch, Eleanor, and Mervis, Carolyn, "Family Resemblances: Studies in the Internal Structure of Categories," *Cognitive Psychology* 7: 573–605, 1975.

[Rosch 78] Rosch, Eleanor, "Principles of Catgeorization" in Rosch E., and Lloyd B., (eds) *Cognition and Categorization*, Lawrence Erlbaum, Hillsdale, NJ, 1978.

[Rosch et al 76] Rosch, Eleanor, Mervis, Carolyn, Gray, Wayne, Johnson, David, Boyes-Braem, Penny, "Basic Objects in Natural Categories," *Cognitive Psychology* 8: 382–439, 1976.

[Rosenblatt 59] Rosenblatt F., "Two Theorems of Statistical Separability in the Perceptron," Proc. of the NPL Symposium on Mechanization of Thought Processes, Washington, D.C., 1959.

[Rosenbloom 88] Rosenbloom, Paul, "Beyond Generalization as Search: Towards a Unified Framework for the Acquisition of New Knowledge," Proc. of the AAAI Spring Symposium on Explanation-Based Learning, Stanford, CA, 1988.

[Rumelhart 75] Rumelhart, David E., "Notes on a Schema for Stories," in Bobrow D.G, and Collins A., (eds) *Representation and Understanding*, Academic Press, New York, NY, 1975.

[Salton & McGill 83] Salton, Gerard and McGill, Michael J., *Introduction to Modern Information Retrieval*, McGraw-Hill, New York, NY, 1983.

[Salton 88] Salton G., "Syntactic Approaches to Automatic Book Indexing," Proc. of the ACL Conference, Buffalo, NY, 1988.

[Samuel 59] Samuel, Arthur, "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal*, 3, 211–229, 1959.

[Samuel 67] Samuel, Arthur, "Some Studies in Machine Learning Using the Game of Checkers: Recent Results," *IBM Journal*, 1967.

[Schank & Abelson 77] Schank, R.C., and Abelson, R.P., *Scripts, Plans, Goals and Understanding*, Lawrence Erlbaum, Hillsdale, NJ, 1977.

[Schank & Rieger 74] Schank, Roger C., and Rieger, Charles J., "Inference and the Computer Understanding of Natural Language," *Artificial Intelligence*, 5: 373–412, 1974.

[Schank 73] Schank, Roger C., "Identification of Conceptualizations Underlying Natural Language," in Schank R.C., and Colby K.M., (eds) *Computer Models of Thought and Language*, W. H. Freeman Co, San Francisco, CA, 1973.

[Schank 75a] Schank, Roger C., "The Primitive ACTs of Conceptual Dependency," Proc. of the Theoretical Issues in Natural Language Processing, Cambridge MA, 1975.

[Schank 75b] Schank, Roger C., "The Structure of Episodes in Memory," in Bobrow D.G., and Collins A., (eds)*Representation and Understanding*, Academic Press, New York, NY, 1975.

[Schank 79] Schank, Roger C., "Interestingness: Controlling Inference," *Artificial Intelligence*, 12: 273–298, 1979.

[Schank 82] Schank, Roger C., *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*, Cambridge University Press, New York, NY, 1982.

[Schank 86] Schank, Roger C., *Explanation Patterns: Understanding mechanically and creatively*, Lawrence Erlbaum, Hillsdale, NJ, 1986.

[Schank et al 90] Schank Roger C. et al., "Towards a General Content Theory of Indices," Proc. of the AAAI Spring Symposium on Case-Based Reasoning, Stanford, CA, 1990.

[Scholnick 83] Scholnick, Ellin K., *New Trends in Conceptual Representation: Challenges to Piaget's Theory?*, Lawrence Erlbaum, Hillsdale, NJ, 1983.

[Searle 80] Searle, J., "Minds, Brains, and Programs," *Behavioral and Brain Sciences* 3: 417–457, 1980.

[Segre 87] Segre, Alberto M., "On the Operationality/Generality Tradeoff in Explanation-Based Learning," Proc. of the International Joint Conference on AI, Milano, Italy, 1987.

[Shavlik 88] Shavlik, Jude, "Generalizing the Structure of Explanations in Explanation-based Learning," Ph.D Dissertation, University of Illinois, Urbana, IL, 1988.

[Silver 83] Silver, Bernard, "Precondition Analysis: Learning Control Information," in Michalski R.S., Carbonell J.G., and Mitchell, T.M., (eds) *Machine Learning: An Artificial Intelligence Approach, Vol 2*, Morgan Kaufman, Los Altos, CA, 1983.

[Skalak & Rissland 90] Rissland, E. L. and Skalak, D. B., "CABARET: Statutory Interpretation in a Hybrid Architecture," *International Journal of Man-Machine Studies*, 1990 (to appear).

[Smith 85] Smith, Edward, "Cognitive Psychology: Correspondent's Report," *Artificial Intelligence*, 25: 247–253, 1985.

[Smith & Medin 81] Smith, Edward and Medin, Douglas, *Categories and Concepts,* Harvard University Press, Cambridge, MA, 1981.

[Soloway 78] Soloway E.M., "Learning = Interpretation + Generalization: A Case-study in Knowledge Directed Learning," Ph.D Dissertation, Computer and Information Science, Technical Report 78-13, University of Massachusetts, Amherst, MA, 1978.

[Stanfill & Waltz 86] Stanfill, C. and Waltz D., "Toward Memory-Based Reasoning" *Communications of the ACM*, 29: 1213–1228, 1986.

[Stansfield 77] Stansfield W.D., *The Science of Evolution,* MacMillan, New York, NY, 1977.

[Subramanian 89] Subramanian, Devika, "A Theory of Justified Reformulations," Ph.D Dissertation, STAN-CS-89-1260, Computer Science Department, Stanford University, Stanford, CA, 1989.

[Suchman 87] Suchman, Lucy A., *Plans and Situated Actions*, Cambridge University Press, Cambridge, England, 1987.

[Sumida & Dyer 89] Sumida, Ronald A., and Dyer, Michael G., "Storing and Generalizing Multiple Instances While Maintaining Knowledge-level Parallelism," Proc. of the International Joint Conference on AI, Detroit, MI, 1989.

[Suthers 89] Suthers, Daniel D., "Perspectives in Explanation," COINS Technical Report 89-24, University of Massachusetts, Amherst, MA, 1989.

[Swaminathan 88a] Swaminathan, Kishore, "Integrated Learning With an Incomplete and Intractable Domain Theory: The Problem of Epidemiological Diagnosis," Proc. of the AAAI Spring Symposium on Explanation-Based Learning, Stanford, CA, 1988.

[Swaminathan 88b] Swaminathan, Kishore, "Properties of an Indexing Scheme," Proc. of the Case-Based Reasoning Workshop, Minneapolis, MN, 1988.

[Tadepalli 85] Tadepalli, Prasad, "Learning in Intractable Domains," Proc. of the Machine Learning Workshop, Skytop, PA, 1985.

[Tulving 72] Tulving, Endel, "Episodic and Semantic Memory," in Tulving E., and Donaldson W., (eds) *Organization of Memory*, Academic Press, New York, NY, 1972.

[Tulving 83] Tulving, Endel, *Elements of Episodic Memory*, Oxford University Press, Oxford, England, 1983.

[Utgoff 84] Utgoff, Paul E., "Shift of Bias for Inductive Concept Learning," Ph.D Dissertation, Rutgers University, New Brunswick, NJ, 1984.

[Valiant 84] Valiant, Leslie G., "A Theory of the Learnable," *Communications of the ACM*, 27: 11, 1984.

[Vere 75] Vere S., "An Induction of Concepts in the Predicate Calculus," Proc. of the International Joint Conference on AI, Tbilisi, USSR, 1975.

[Voss & Bisanz 85] Voss J.F., and Bisanz G.L., "Knowledge and the Processing of Narrative and Expository Texts," in Britton B., and Black J., (eds) *Understanding Expository Text*, Lawrence Erlbaum, Hillsdale, NJ, 1985.

[Waltz 82] "The State of the Art in Natural Language Understanding," in Lehnert W.G., and Ringle M.H., (eds) *Strategies for Natural Language Processing*, Lawrence Erlbaum, Hillsdale, NJ, 1982.

[Waterman 70] Waterman, D.A., "Generalization Learning Techniques for Automating the Learning of Heuristics," *Artificial Intelligence*, 1: 121–170, 1970.

[Weizenbaum 76] Weizenbaum, J., *Computer Power and Human Reason*, W.H. Freeman Publishers, San Francisco, CA, 1976.

[Wilensky 83] Wilensky, Robert, *Planning and Understanding: A Computational Approach to Human Reasoning*, Addison-Wesley, Reading, MA, 1983.

[Wilensky 87] Wilensky, Robert, "Some Problems and Proposals for Knowledge Representation," Technical Report, University of California, Berkeley, CA, 1987.

[Wilensky 82] Wilensky, Robert, "Points: A Theory of the Structure of Stories in Memory," in Lehnert W.G., and Ringle, M.H., (eds) *Strategies for Natural Language Processing*, Lawrence Erlbaum, Hillsdale, NJ, 1982.

[Winograd 72] Winograd, Terry, *Understanding Natural Language*, Academic Press, New York, NY, 1972.

[Winston 71] Winston P.H., "Learning Structural Descriptions from Examples," Technical Report 231, MIT AI Laboratory, Cambridge, MA, 1970.

[Wittgenstein 53] Wittgenstein, Ludwig. *Philsophical Investigations.* Macmillan, New York, NY, 1953.

[Woolf 88] Woolf, Beverly, "Intelligent Tutoring Systems: A Survey," in Shrobe H.E., (ed) *Exploring Artificial Intelligence*, Morgan Kaufmann, San Mateo, CA, 1988.

[Zubin & Kopcke 86] Zubin, David and Kopcke, Klaus-Michael, "Gender and Folk Taxonomy: The Indexical Relation Between Grammatical and Lexical Categorization," in Craig C., (ed) *Categorization and Noun Classification*, Benjamins Northamerica, Philadelphia, PA, 1986.