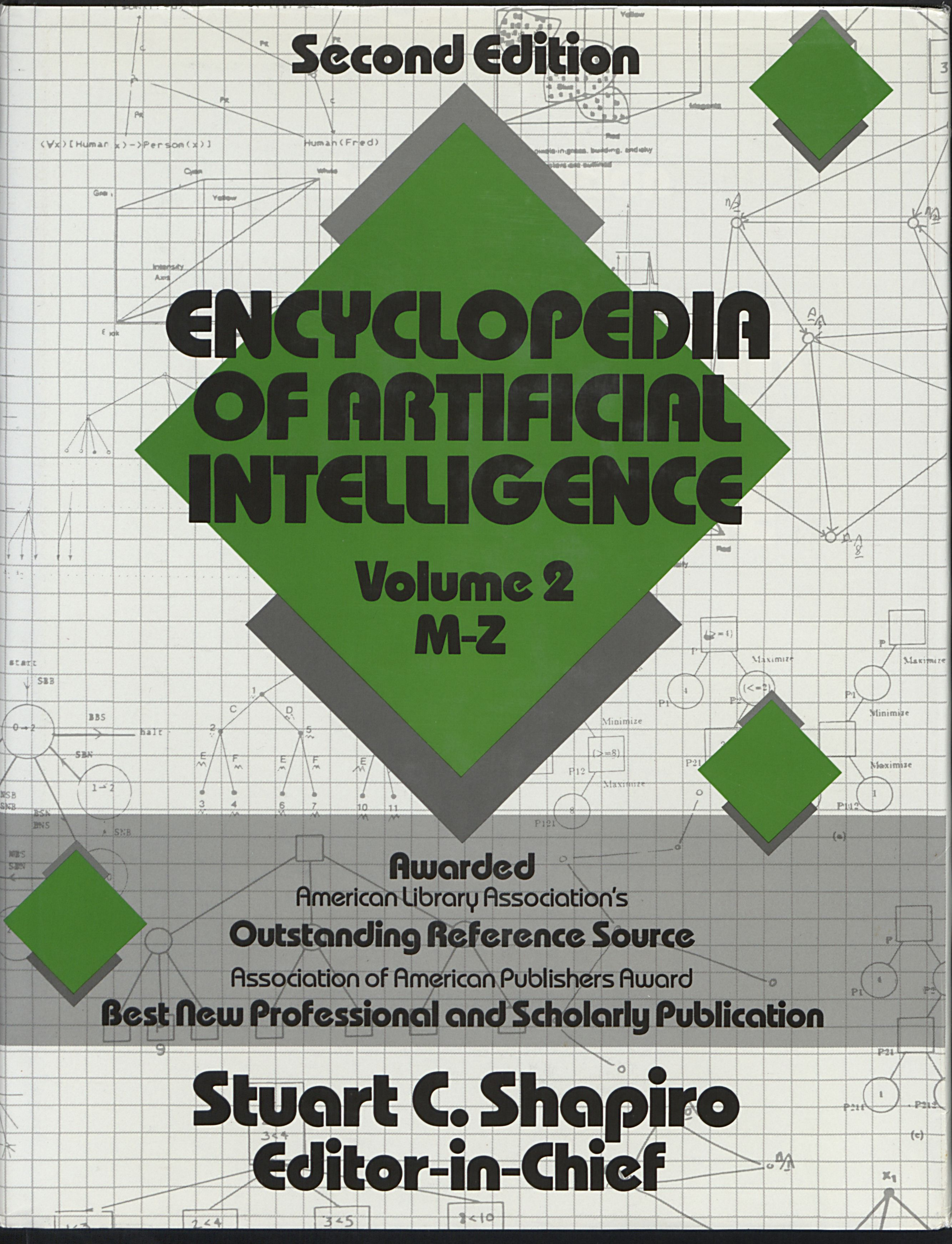
Second Edition

ENCYCLOPEDIA OF ARTIFICIAL INTELLIGENCE

Volume 2
M-Z

Awarded
American Library Association's
Outstanding Reference Source
Association of American Publishers Award
Best New Professional and Scholarly Publication

Stuart C. Shapiro
Editor-in-Chief



Representation & Reasoning

Menachem Y. Jona and Janet L. Kolodner,
REASONING, CASE-BASED, 1265-1279

From Shapiro, Stuart C., Editor-in-Chief,
Encyclopedia of Artificial Intelligence, 2nd
Ed., John Wiley & Sons, Inc., NY, 1992.

REASONING, CASE-BASED

Case-based reasoning is the technique of solving new problems by adapting solutions that were used to solve old problems. This reliance on previous experiences (or cases) is a hallmark of case-based reasoning. Each case can contain a great deal of information including a description of the situation that was encountered, ways in which the situation differed from similar situations, and how the system reacted to the situation.

There are many examples of people using case-based reasoning in their daily lives. A caterer who remembers a meal served at a previous banquet and adapts it to fit the demands of a new client is using case-based reasoning. So is a car mechanic who suspects the problem when a car is brought into the shop with symptoms similar to a previous one that has been fixed. A business executive uses case-based reasoning by remembering past experiences when doing long range planning for the future.

In particular, case-based reasoning can mean adapting old solutions to meet new demands, using old cases to explain new situations, using old cases to critique new solutions, or reasoning from precedents to interpret a new situation (much like lawyers do) or create an equitable solution to a new problem (much like labor mediators do).

Case-based reasoning is used extensively by people in both expert and commonsense situations. It provides a wide range of advantages.

- Case-based reasoning allows the reasoner to propose solutions to problems quickly, avoiding the time necessary to derive those answers from scratch.
- Case-based reasoning allows a reasoner to propose solutions in domains that are not completely understood.
- Case-based reasoning gives a reasoner a means of evaluating solutions when no algorithmic method is available for evaluation.
- Cases are particularly useful for use in interpreting open-ended and ill-defined concepts.
- Remembering previous experiences is particularly useful in warning of the potential for problems that have occurred in the past, alerting a reasoner to take actions to avoid repeating past mistakes.
- Cases help a reasoner to focus on important parts of a problem by pointing out what features of a problem are the important ones.

Below, the development of the case-based reasoning (CBR) paradigm is traced and the advantages of CBR as a problem-solving methodology are discussed in more detail. The general CBR algorithm and some of the fundamental issues that must be dealt with in any CBR system is described. Next, a survey of CBR systems that have been built to perform various tasks along with pointers for further reading is presented, and finally, a short discussion of the implications of CBR as a cognitive model and some pointers on how to go about building a CBR system are included.

EVOLUTION OF THE CBR PARADIGM

Case-based reasoning differs markedly from other types of reasoning and problem-solving techniques and is, in some instances, a direct reaction to the problems of these other techniques. Two important factors have contributed to the evolution of the CBR paradigm.

Theories of Memory

CBR evolved in part from research on human memory, particularly the theory of Dynamic Memory (qv) developed by Schank (1982). This theory introduced the memory organization packet (MOP) memory structures which can be considered the intellectual precursor to cases. MOPs are used to store not only general information about the world, but specific experiences as well. The theory of dynamic memory provided a framework for describing how these individual experiences can be stored in memory, how they may be combined and abstracted, and how they can be retrieved and used when the need arises. Specifying the content and process of memory in this way provided the groundwork for some of the ideas of CBR.

Problems with Rule-based Reasoning

A second driving force in the evolutionary history of CBR was dissatisfaction with rule-based reasoning (expert systems (qv)), the predominant problem-solving technique at the time. Three problems with rule-based systems which prompted a search for an alternative paradigm for problem-solving concern knowledge acquisition, memory, and robustness.

Knowledge Acquisition. Under the rule-based problem solving paradigm, collecting knowledge to encode into the systems was a very difficult endeavor. Knowledge engineers found it hard to uncover the hundreds of rules that the "expert" used to solve problems, mainly because the expert often had a hard time trying to articulate his or her problem-solving skill in the form of IF-THEN rules. This problem became known as the knowledge acquisition bottleneck (Hayes-Roth and co-workers, 1983). Furthermore, it became unclear whether experts were actually using rules at all; often experts will say that it is their experience that makes them experts. In addition, while rules seemed like nice compact representations to collect, it is often the case that rules have many exceptions, making the knowledge acquisition problem that much harder. To complicate matters even further, it was necessary to trace the interactions between rules to ensure that they could chain together properly, and that contradictions were eliminated.

No Memory. A second major criticism of rule-based reasoning systems is that most did not have any memory; that is, the system would not remember previous encounters with the same problem, and would have to solve them again from scratch. This, of course, is terribly inefficient, but even dismissing efficiency issues for a moment, an even more important point is that a system with no memory will not be able to remember past mistakes. Without this ability, the system is surely condemned to repeat those mistakes again and again. This type of stupidity is not tolerated in human problem solvers, and many people were dissatisfied with rule-based systems' inability to learn from their mistakes.

Robustness. A third serious criticism of rule-based systems is that they were brittle. Since all their knowledge was recorded in terms of rules, if a problem did not match any of the rules, the system could not solve it. In other words, rule-based systems had little or no ability to work beyond their rule base; they could not adapt to handle novel situations very well, if at all.

ADVANTAGES OF THE CBR APPROACH

CBR has many advantages as a theory and methodology of reasoning and problem-solving. Many of these advantages are in direct response to the factors, outlined above, that led to the development of the CBR paradigm.

Psychological Plausibility

Given that it grew out of research on human memory, it is not surprising that one of the things that makes CBR appealing as a model of reasoning and problem solving is that it is based on the way in which humans reason and solve problems. Reasoning from past cases, as opposed to a (large) set of rules, has been thought to be a more psychologically plausible model of how people reason (Holyoak and Koh, 1987).

There is much evidence that people use case-based reasoning in their daily reasoning. Ross (1989a, 1989b), for example, has shown that people learning a new skill often refer back to previous problems to refresh their memories on how to do the task. Other research has shown that both novice and experienced car mechanics use their own experiences and those of others to help them generate hypotheses about what is wrong with a car, recognize problems (eg, a testing instrument not working), and remember how to test for different diagnoses (Lancaster and Kolodner, 1988; Redmond, 1989). In an unpublished study, Kolodner found that physicians use previous cases extensively to generate hypotheses about what is wrong with a patient, to help them interpret test results, and to select therapies when several are available and none are understood very well. Goel and Pirolli (1989) observed architects and mechanical engineers recalling, merging, and adapting old design plans to create new ones. Klein and Calderwood (1988) have observed routine use of case-based reasoning among experts making decisions in dynamically-changing situations. Read and Cesa (1991) observed people using old cases to construct explanations of social situations.

Cases vs Rules

As discussed above, the ideas of CBR developed, in part, as a reaction to the problems of rule-based reasoning. Case-based reasoning offers advantages over rule-based systems in the following ways:

Knowledge Acquisition. Cases are more memorable than abstract rules. It is often easier for experts to remember and articulate specific examples of the problems they have encountered and their solutions to those problems (ie, their "war stories"), than it is for them to describe their problem-solving technique in terms of potentially large numbers of rules. In fact, several people building expert systems that know how to reason using cases have found it easier to build case-based expert systems than traditional ones (Barletta and Hennessy, 1989; Goodman, 1989).

Learning from Experience. Case-based reasoning systems, by definition, are built on a memory of prior cases. Each time the system solves a problem, that problem and its solution are stored in memory as a case. In this way CBR systems can easily learn from experience; they don't have to waste effort re-solving a problem that is just like one they have seen before, nor will they repeat the mistakes they may have made solving the problem the first time around. While systems that do problem-solving from first-principles spend large amounts of time solving their problems from scratch, case-based systems have been found to be several orders of magnitude faster (Koton, 1988a). This ability to learn from experience is discussed in greater detail in the next section.

Adaptivity. While rule-based systems are brittle, CBR systems display more robustness upon encountering new situations. This robustness derives from the techniques of case adaptation. When trying to solve a new problem, a CBR system can search its memory for previously seen problems with similar features and adapt the solutions to those problems so that they are useful in solving the new problem (Kass, 1989).

Natural Learning Mechanism

Learning from experience, as mentioned above, is one of the advantages that CBR has over rule-based systems. The CBR paradigm provides a natural mechanism for learning. A case-based reasoner learns in two basic ways. First, it can become a more efficient reasoner by remembering old solutions and and adapting them rather than having to derive answers from scratch each time. If a case was adapted in a novel way, if it was solved using some novel method, or if it was solved by combining the solutions to several cases, then when it is recalled during later reasoning, the steps required to solve it won't need to be repeated for the new problem. Second, a case-based reasoner becomes more competent over time, deriving better answers that it could with less experience. One of case-based reasoning's fortes is its ability to help a reasoner anticipate and thus avoid past mistakes.

Case-based learning offers many advantages as a learning paradigm, including:

- **Easier knowledge acquisition.** Because knowledge is stored primarily in the form of cases, the start-up threshold is smaller. That is, learning processes can begin with much less "data" than rule-based systems typically require. Debugging the knowledge base is also easier because there tend to be far fewer interactions between cases than between rules. Finally, many domains already have information encoded in case format, prime examples being the law, mathematics, and design domains.
- **Performance enhancements.** Because CBR systems store previous encounters with problems, they can reuse old solutions instead of having to derive new solutions from scratch. In addition, by remembering past mistakes, a CBR problem-solving system can avoid making the same mistakes again. Both of

these factors contribute to improved performance (efficiency) of CBR systems.

- **Straightforward learning.** In general, learning in a CBR system does not require a complex causal model of the domain or detailed domain knowledge. Of course, the addition of either or both of these items can enhance the performance and power of a case-based system.
- **Cases can serve as explanations.** One feature that is often desired in problem solving systems is the ability to offer an explanation for the solution obtained. In a CBR system, such explanations are simply the case (or cases) that were used, making them easy (or even trivial) to generate. In fact, because CBR solves problems like people do, an explanation based on a concrete past case may be more satisfying then explanations constructed out of chains of rules, the primary method of explanation in rule-based systems.
- **Scalability.** The common problem encountered when scaling up a system, that of massive search, is in some ways avoided in CBR by the use of indexing (discussed in the section on the Indexing Problem). With the development of parallel algorithms for retrieval, it appears that large CBR systems may be able to approach, or even achieve, real-time performance.

THE BASIC CBR ALGORITHM

The basic processing cycle of CBR is "input a problem, retrieve relevant past solutions, adapt them to the current problem, store the new case along with its solution." In this section we elaborate on this cycle, spelling out the various steps in a typical case-based reasoning algorithm. While different CBR systems may emphasize different parts of the cycle, all systems address the following steps in some way.

1. *Accept and Analyze.* Upon input of a new problem, the first step of processing is known as the analysis phase. In this step, the input is analyzed to extract features to use in retrieving cases with similar features. These features, which are given special status in CBR, are called indexes. Thus, the main task in this step is index extraction. Index extraction is a very complex problem, discussed in more detail in the following section.

2. *Retrieve Cases from Memory.* The indexes computed in Step 1 are used to retrieve cases from memory. The goal here is not to retrieve just any set of cases. Those cases that can be used in the reasoning to be done in the next steps and that have the potential to make useful predictions about the current problem are the kinds of cases that should be retrieved. Various techniques for retrieving cases exist, including iterative, parallel, and constraint-satisfaction models (see section on retrieval algorithms following).

3. *Select Most Relevant Case(s).* The (potentially large) set of relevant cases obtained in Step 2 often needs to be narrowed down to just a few "most relevant" cases. These cases will be the ones considered most worthy of intensive processing in forming the basis for the new solution. The problem in this step is assessing how relevant each case really is. Typical techniques for this include a variety of ranking schemes and similarity metrics, overlap of salient features, and importance of shared features being two examples. Some of the problems of assessing similarity are discussed in the section on similarity metrics below.

4. *Construct Solution.* This step uses the cases selected in Step 3 to create a solution or interpretation (depending on the task) for the input case. Along with the solution, many CBR systems will also construct the justifications or supporting arguments for the solution at this point. The retrieved cases are used in constructing the solution in at least two important ways. First, the actual solution is constructed by adapting the solutions to the previously seen cases so that they are relevant to the current case. Second, the retrieved cases can be used to warn of potential snags in solution construction, allowing the system to anticipate and thus avoid making the same kinds of mistakes encountered in previous problems.

5. *Evaluate Solution.* After a potential solution has been constructed, it is subjected to testing, evaluation, and criticism. The goal in this step is to assess the utility, strengths, and weaknesses of the proposed solution. Several methods for doing this exist, including testing the solution against counterexamples (real or hypothetical), using the solution as an index into memory to see if there are any examples of this solution that have been known to fail in similar circumstances, or simulating the results of the proposed solution. Examples of systems which use these methods are HYPO (Ashley, 1987, 1988; Rissland, 1986), which provides guidelines for creating and using hypotheticals, and CHEF (qv) (Hammond, 1986, 1989a), which used a simulation to test its solutions. Employing internal testing mechanisms like these is especially critical in domains where the cost of an incorrect or inefficient solution is high (eg, medical diagnosis).

6. *Execute Solution and Analyze Results.* In this step the solution is tried out in the real world and the system obtains feedback about what happened. This feedback is then subjected to careful analysis to see if the results were as expected. This process of obtaining and analyzing feedback is crucial if a CBR system is to learn from its mistakes and avoid repeating them. If something unexpected occurred the system attempts to explain the anomalous events. The problem of trying to decide which parts of the solution caused the problems, known throughout the machine-learning field as the credit/blame assignment problem, comes into play here. One technique that can be used to partially avoid this notoriously hard problem is to recall similar failures encountered in the past and to use the explanations of those failures in explaining the current failure (Rissland and Ashley, 1988). This process is simply another version of a case-based reasoning problem and can often be done by a recursive call to the CBR system.

7. *Update Memory.* After the results of testing the solution in the real world have been analyzed, the next step is to update memory by storing the new case. The new

case is composed of not only the solution arrived at, but also the justifications and supporting arguments constructed in Step 4. The most important aspect of this step is where to put this case in memory, or, in CBR terminology, how to index it. One common technique is to index a case by the problems or failures that were encountered (in Steps 5 and 6) so that these same mistakes can be avoided when a similar situation is encountered in the future (see section below for more detail on other indexing techniques). The success or failure of a CBR system depends heavily on this step. Good indexing strategies will cause the relevant cases to be recalled when they can best be used, resulting in good performance, while a poor indexing scheme will not cause the most relevant cases to be retrieved and system performance will be degraded.

This section was an attempt to give a brief overview of the general CBR algorithm. For more detail on CBR basics, particularly implementation details, an excellent starting point is Riesbeck and Schank's (1989), *Inside Case-Based Reasoning*. Good general introductions for the lay reader can be found in Slade (1991) and Kolodner (1990, 1991).

FUNDAMENTAL ISSUES IN CBR

Because the heart of any case-based reasoning system involves case retrieval and selection, the fundamental issues of CBR revolve around these issues. Below are presented some of the major areas of current research, the questions that each is trying to address, and some of the proposed solutions.

The Indexing Problem

Retrieval of relevant cases from memory being a cornerstone of the CBR algorithm, it follows that an extremely important issue is how to label cases so that they may be recalled when needed. The assignment of labels to cases is called indexing, the labels themselves are called indexes.

The Nature of Indexes. What are indexes comprised of? This question has long plagued CBR researchers. In general, an index can be any feature used in the representation of a case or computed from that representation. But which features should be used as indexes? One option is to use concrete features, ie, those features which are either included in the description of the case, or can be easily computed. These kinds of features, also known as low level, or surface features, have the advantage that they are simple to represent and do not require much (if any) effort to extract from the input. The simplicity and ease of computation of these kinds of features tend to make them favored for use in parallel retrieval algorithms (see section below). The problem with low level features, however, is that they may not be adequate to index cases so that only the most relevant cases are recalled. Combinations of simple features are often insufficient in describing the type of case being looked for with sufficient specificity, resulting in a large set of cases, only a few of which are relevant, being retrieved from memory. If this happens, the CBR system must then devote additional effort in weeding out the less relevant cases.

A second option is to use as indexes more complex, abstract features which are computed from the input. These high level features have the advantage of being able to more accurately represent the type of case being searched for, resulting in fewer, but more relevant cases being retrieved from memory. The two main problems with using high level features as indexes are deciding which high level features to compute and, of course, the cost of computing them. The former problem, that of deciding which high level features to compute (out of a potentially infinite set of possibilities) is such a notoriously hard problem in CBR that it has been given a special name. The indexing problem is the problem of determining what other nonobvious features, aside from those directly provided in the input, should be used as indexes in a particular domain.

Once a decision to use high level features has been made, however, there are still several issues that must be dealt with. First, it must be decided which high level features are more likely to help retrieve relevant cases in a given domain. A second, related issue that must be addressed is whether the end justifies the means. That is, will the improvement in system performance due to better case selection outweigh the extra cost of computing the high level features? CASEY (qv) (Koton, 1988a), JUDGE (Bain, 1986), CHEF (Hammond, 1986, 1989a), and ANON (Owens, 1989a, 1989b) each provide different initial approaches to this problem.

There has been a great deal of debate among CBR researchers about the relative merits of using low level vs high level features as indexes. Arguments for low level features include Thagard and Holyoak (1989b) and Waltz (1989), while endorsements for the use of high level features (like goal and plan interactions) include Collins' (1987) COACH, other recent work by Collins and Birnbaum (1989; 1990), as well as Schank (1982), Martin (1989a, 1989b), Owens (1989a), and Pazzani (1989) among others. More recently, however, the debate over the level of indexes used has waned as it has become clear that no one kind of index is appropriate for all systems. This has been paralleled by investigations of human analogical reasoning that have found both surface and deep features are used for retrieval (Gentner, 1989). Researchers are now arguing for a more functional approach in which index selection would be based on an analysis of the task the system is to perform. Those features, both surface and deep, which are best suited to the particular task facing the system would be selected as indexes (Hammond, 1989c).

One final trend in addressing the indexing problem has been to investigate whether there is any way of describing a general framework for the content of indexes. If such a method could be developed, it would be a first step towards automating the choice of indexes for a given domain. This would at least partially obviate the need for the programmer of a CBR system to make this difficult decision. A first attempt at describing a general content theory of indexes has resulted in the formulation of the Universal Indexing Frame (UIF), a representational system whose utility is currently being investigated (Schank and co-workers, 1990).

Guidelines for Addressing the Indexing Problem. The indexes of a case are those features that distinguish it from other cases, because they are predictive of something important in the case. Guidelines for addressing the indexing problem include the following:

- Feature combinations used as indexes should be predictive of something important for later reasoning.
- Indexes should be abstract enough to be generally applicable but concrete enough to be easily recognizable without a great deal of inference.
- Cases should be indexed in ways that support reasoning that a system has to do. For example:
  - To choose plans that achieve goals according to the details of a situation, index by goal, constraint, and feature combinations that led to solving a problem in a particular way.
  - To anticipate potential problems, to help explain problem solving errors, and to help recover from problem solving errors, index by the combinations of features responsible for failures.
  - To evaluate proposed solutions, index by combinations of features that were responsible for unexpected outcomes and by descriptions of those unusual outcomes.

Guidelines for choosing indexes are still ahead of the technology for automating index selection. Nevertheless, several methodologies for choosing indexes automatically do exist or have been suggested. (See Table 1 for information about the CBR systems mentioned throughout this article).

- Keep track of norms and index by features different from the norm (CYRUS (qv), MEDIATOR (qv), PERSUADER (qv)).
- Index by a fixed and well-known set of features known to be predictive (CYRUS (qv), HYPO).
- Index by differences between what is already in memory and the case being indexed (CYRUS, MEDIATOR, PERSUADER).
- Index on features that predict failure or unexpected success as follows: After receiving feedback and explaining a failure or unexpected success, use explanation-based learning (EBL) techniques to generalize the explanation. Index by the combination of features that go into the generalized explanation (CHEF (qv), JULIA (qv)).
- Index on features found to be useful in achieving some goal or doing some task as follows: Use EBL techniques to generalize the reasoning that went into making decisions while solving a problem. Index by the combination of features that make up that general reasoning chain (Lockheed AI Project, JULIA).

Memory Organization

Choosing good indexes is not the only factor that contributes to a CBR system's ability to retrieve relevant cases in an efficient manner. A critical factor, especially for large CBR systems containing hundreds or even thousands of cases, is how the system's memory is organized. Traditional approaches have used discrimination networks (cf Feigenbaum, 1963), typical examples being the earlier work of Kolodner (1983a, 1983b) and Lebowitz (1983). More recently, work has concentrated on memory organizations that support parallel retrieval methods, eg, Kolodner (1986, 1988). (See the following section for more discussion on retrieval methods.)

Intrinsically bound up with the question of how memory is organized is the question of how individual cases are represented. Should cases be stored in one place or should they be broken into pieces? The advantage of the former approach is that by storing the entire case in one place, it may be retrieved and used to solve a new problem in just "one shot." A disadvantage is that it is harder to create solutions that are based on pieces of several cases. To do this it would be necessary to go through memory finding and "collecting" the appropriate pieces from the various cases. Systems that use a unitary representation for cases include CASEY, CHEF, and HYPO (see Table 1 for more information on these and other CBR systems).

An alternative is to use a more piecemeal representational scheme for cases, that is, breaking cases into parts which are located in different areas of memory and connected by pointers. This technique makes it easier to create solutions based on partial solutions from several different cases because it is easier to identify and access the parts that are needed. Many feel that this type of solution construction results in more creative solutions to problems. The cost incurred by this approach, however, is that extra work needs to be done to put a single case "back together" before it can be used as a whole. JULIA and CELIA (Redmond, 1990) are examples of CBR systems that use this approach to case representation.

In addition to questions about how memory is organized and how cases are represented, another issue that needs to be considered is forgetting. Should case-based systems ever "throw out" cases? If so, when should this be done? This issue has received very little attention (but see, eg, Hunter, 1989), and much work still needs to be done.

Retrieval Algorithms

Because the CBR paradigm relies on a large memory of cases to give it problem-solving power, a major issue that needs to be considered is how to retrieve cases from memory quickly and efficiently. The larger the memory, the more important this question becomes.

The two main strategies that are used for retrieval are based on the two types of memory organizations discussed above. Memories organized in discrimination nets typically use a concept-refinement search that takes advantage of the generalization–specialization hierarchy built into the net. In this technique search starts at the top of

Table 1. Summary of CBR Systems*

Program	Reference	Domain	Task
ABE	Kass (1990)	Anomalous events	Adaptating explanations
ANON	Owens (1989a)	Proverbs	Indexing prototypical cases
CASEY	Koton (1988a, 1988b)	Heart failures	Explanation of anomalies
CELIA	Redmond (1990)	Automobile troubleshooting	Diagnosis
CHEF	Hammond (1986, 1989a); Riesbeck & Schank (1989)	Recipes	Goal-driven design, plan-repair
CLAVIER	Barletta & Hennessey (1989)	Autoclave layout	Layout design
COACH	Collins (1987); Riesbeck & Schank (1989)	Football strategy	Plan repair, counterplanning
CSI BATTLE PLANNER	Goodman (1989)	Military	Plan critique and repair
CYCLOPS	Navinchandra (1988)	Landscaping	Design
CYRUS	Kolodner (1984)	Political events	Memory organization
DMAP	Riesbeck & Martin (1985); Riesbeck & Schank (1989)	Natural language parsing	Classification, recognition
HYPO	Ashley (1987, 1988); Rissland (1986); Rissland & Ashley (1986, 1988)	Patent law	Evaluation by comparison
JUDGE	Bain (1986); Riesbeck & Schank (1989)	Criminal sentencing	Evaluation by comparison
JULIA	Hinrichs (1988, 1989)	Catering	Goal-driven design
KRITIK	Goal (1989); Goel & Chandrasekaran (1989)	Mechanical assemblies	Design
MEDIATOR	Simpson (1985); Kolodner & Simpson (1989)	Common-sense disputes	Goal-driven design
MEDIC	Turner (1989)	Medicine	Multiple diagnostic goals
PARADYME	Kolodner (1988)	Cooking	Parallel retrieval
PERSUADER	Sycara (1987)	Labor contracts	Goal-driven design
PLEXUS	Alterman (1986, 1988)	Subway riding	Execution time plan repair
PROTOS	Bareiss (1989)	Hearing disorders	Diagnosis, learning
EXPEDITOR	Robinson & Kolodner (1991)	Daily errands	Planning for multiple goals
SWALE	Kass and co-workers (1986); Kass & Leake (1988); Leake & Owens (1986)	Death and destruction	Explanation of anomalies
TRUCKER	Hammond (1989b)	Scheduling	Opportunistic planning

* After Slade (1991).

the net and progresses downward only when a match can be made at the current level. In this way, large portions of the memory can be eliminated from the search almost immediately. When the search terminates, the set of cases grouped below the current node can be returned. This set will have increasing similarity to the probe to the extent that the search progresses further down the net. Thus, the set of cases returned will all, in some sense, be "close" to the probe (Kolodner, 1983a, 1983b; Lebowitz, 1983).

The second major class of retrieval algorithms are parallel algorithms. These derive their power by examining all (or many) cases at once. Generally what is done is that each match is given a rating of its goodness (using some metric) and those cases with the highest ratings are the ones returned by the search. Kolodner's (1984) CYRUS was the first to investigate the use of parallelism in CBR. Her solution used a combination of shared feature networks and redundant discrimination networks that could be searched in parallel. Other types of parallel algorithms that have been developed for use in CBR systems include parallel search of a flat memory (as in MBR (Stanfill and Waltz, 1988)), and parallel search of a hierarchical memory (as in PARADYME (Kolodner, 1988)). Further discussion of parallel retrieval techniques can be found in Owens (1989b), Thagard and Holyoak (1989b), Domeshek (1989), and Waltz (1989).

Similarity Metrics

If a CBR system is to solve new problems by adapting solutions to old problems, immediately one must face the question of how to recognize one situation as being similar to another. In trying to choose the best cases to reason with, the system must first match the input to cases in memory to retrieve a set of candidate cases and then narrow down this set to include only the most relevant cases (Steps 2 and 3 in the CBR Algorithm section above). Since it is unlikely that a new case will always match a case in memory exactly, a system must be able to do partial matching in order to accomplish the first step of retrieving relevant cases. The system must also have the ability to compare the goodness-of-match of the retrieved cases in order to do the second step of narrowing down the set of relevant cases to a smaller set of "best" cases. Both of these processes entail the need to have similarity metrics, or ways of judging how alike two cases are (along various dimensions).

A naive method of assessing similarity between two cases would be to count the number of matching features that the two cases have. This technique is of limited usefulness though, since the relative importance of features often changes depending on the context. More sophisticated methods use the cases already in memory, along with various decision heuristics, in deciding which features are important for matching (Ashley and Rissland, 1988; Kolodner, 1988; Owens, 1989a; Rissland and Ashley, 1988; Stanfill, 1987).

The advantage of the naive approach, however, is that comparison of simple features is computationally less expensive than matching complex structures. The issue of efficiency becomes quite important in deciding on a similarity metric, since assessing similarity between cases plays a role in many of the steps in the CBR algorithm. Recent efforts at reducing the complexity of matching cases have used the UIF to implement flat-matching, that is, eliminating the use of variable binding in the matching algorithm by Domeshek (in press).

There remain a number of open issues in similarity assessment. Bareiss and King (1989) mention the following:

- How can similarity be computed when cases are represented in a uniform manner?
- How does general domain knowledge come into play in similarity assessment?
- How can the context of the problem solving situation affect the determination of similarity?
- Can similarity indeed be assessed independent of the items being compared? In other words, is similarity computed from first principles each time a judgment must be made, or is it recalled from past experiences? Work on judgments of similarity is relevant here (cf Holyoak and Koh (1987)).

See Bareiss and King (1989) for an overview of current work on similarity assessment, other relevant work being Ashley (1989), Kolodner (1989), Porter (1989), Thagard and Holyoak (1989a), and Whitaker and co-workers (1989).

Case Adaptation

Once the case or cases which are to be used in the construction of a solution have been selected, the next step is to adapt the solutions from the selected cases to the problem at hand. If the current problem is nearly the same as one that has already been solved, then the old solution can be used directly and no adaptation is needed. This, however, is an unusual occurrence and general strategies for adapting cases are needed to handle the more frequently occurring situation in which the solution cannot be used unaltered. The search for case adaptation strategies is basically an attempt to find ways to adapt a case to make it relevant to a new situation.

Techniques for adapting cases vary according to the type of task being performed by the CBR system and to the extent that they are dependent on the particular domain that the system is operating in. CBR systems that do planning or problem solving often have strict criteria that a potential solution must meet. This emphasis on evaluation places limits on the kinds of adaptations that are permitted. Systems that come up with designs or explanations often place more emphasis on creative solutions, encouraging a wider variety of adaptation techniques which, while they may yield many "bad" solutions, may often come up with interesting or creative ones.

Another trend in research on case adaptation strategies has been to try to discover very general, domain-independent rules for modifying cases. If such rules could be discovered, they could be "plugged in" to any CBR system, regardless of whether its task was designing recipes or diagnosing heart conditions. In addition, these general rules might also be able to form the basis for the learning of more specific, domain-dependent rules by the system itself. Several methods of adaptation have been identified to date.

Substitution methods are used to substitute an object, value, or set of objects or values in an old solution for one or a set that better fit the new situation.

Reinstantiation means instantiating the framework for the old solution with new arguments.

Parameter adjustment is a method of adjusting a solution parameter from the old case based on differences between the old and new case descriptions.

Local search is a search in semantic hierarchies for a substitute for some object in an old solution that must be replaced.

Query memory is a broader search for a substitute.

Specialized search directs search to portions of the knowledge base where a substitution is likely to be found.

Transformation methods transform a piece of an old solution to fit the new situation.

Commonsense transformation makes use of commonsense knowledge about what kinds of things can be transformed.

Model-guided repair uses a qualitative model to guide transformation.

Critic application is a methodology for implementing several of the types of adaptation listed above. It also provides a way of implementing ad-hoc adaptation heuristics, especially those that do insertions, deletions, and reorderings.

Derivational replay replays the method used in the old case for deriving some piece of the solution rather than taking the solution itself.

The area of case adaptation is currently a topic of active research. Some early examples of work that used adaptation are CHEF (Hammond, 1986, 1989a), and SWALE (Kass, 1986; Kass and co-workers, 1986). More recent work on adaptation includes Collins (1989), Goel and Chandrasekaran (1989), Hinrichs (1989), and Kass (1989). ABE (Kass, 1990) investigates adaptation techniques in the domain of explaining anomalous events.

REASONING USING CASES: APPLICATIONS OF CBR

There are two main styles of case-based reasoning: problem solving and interpretive. In the problem solving style of case-based reasoning, solutions to new problems are derived using old solutions as a guide. This style of CBR is characterized by heavy use of adaptation processes to generate solutions and interpretive processes to evaluate derived solutions.

In the interpretive style, new situations are evaluated in the context of old situations. A lawyer, for example, uses interpretive case-based reasoning when he or she uses a series of old cases to justify an argument in a new case. The interpretive style of case-based reasoning uses cases to provide justifications for solutions, allowing evaluation of solutions when no clear-cut methods are available and interpretation of situations when definitions of the situation's boundaries are open-ended or fuzzy.

This section presents a survey of the various tasks to which CBR systems have been applied, classified by the kind of CBR being done by each system. A summary of the CBR systems is presented in Table 1.

CBR and Problem Solving

Case-based reasoning is useful for a wide variety of problem solving tasks, including planning, diagnosis, and design. In each of these, cases are useful in both suggesting solutions and in warning of possible problems that might arise. There are additional advantages for each problem solving task.

CBR for Design. In design, problems are defined as a set of constraints, and the problem solver is required to provide a concrete artifact that solves the constraint problem. Usually the given constraints underspecify the problem (ie, there are many possible solutions). Sometimes, however, the constraints overconstrain the problem (ie, there is no solution if all constraints must be fulfilled). In addition, in design, a solution to one piece of a design problem is often tightly coupled to the solution of other pieces. While constraints can be used to maintain the connections between pieces, methodologies that require backtracking are too tedious for complex problems. Case-based reasoning addresses all of these issues.

- Cases suggest solutions to underconstrained problems. The solutions might not be exactly right, but since many different solutions might be appropriate, adaptation heuristics can generally create a satisfactory solution easily.
- When problems are over-constrained, cases suggest an alternative set of constraints that has worked in the past. While some adaptation might still have to be done, the full application of constraint relaxation can be avoided.
- When problem subparts are tightly coupled, cases can provide the glue that holds a solution together. Rather than solving the subparts by decomposing, recomposing, and fixing discrepancies, as is done in solving nearly-decomposable problems, a case suggests an entire solution, and the pieces that don't fit the new situation are adapted in place.

Several problem solvers have been built to do case-based design. JULIA (Kolodner, 1987; Hinrichs, 1988, 1989) plans meals; CYCLOPS (Navinchandra, 1988) uses case-based reasoning for landscape design; and KRITIK (Goel, 1989; Goel and Chandrasekaran, 1989) combines case-based with model-based reasoning for design of small mechanical assemblies. It uses case-based reasoning to propose solutions and uses the model to verify its proposed solutions, to point out where adaptation is needed, and to suggest adaptations. MEDIATOR (Kolodner and Simpson, 1989; Simpson, 1985), the earliest case-based problem solver, solved simple resource disputes, eg, two children wanting the same candy bar or two faculty members wanting to use the copy machine at the same time. PERSUADER (Sycara, 1987) solved labor management disputes.

At least one design problem solver is being put to use in the real world. CLAVIER (Barletta and Hennessy, 1989) is being used at Lockheed to lay out pieces made of composite materials in an oven to bake. The task is apparently a black art, ie, there is no known complete causal model of what works and why. Pieces of different sizes need to be in particular parts of the oven, but the size of some pieces and density of a layout might keep other pieces from heating correctly. The person in charge of layout kept a card file of the experiences, both those that worked and those that did not. Based on those experiences, CLAVIER can place pieces in appropriate parts of the oven and avoid putting pieces in the wrong places. It works as well as the expert whose experiences it uses, and is thus useful to Lockheed when the expert is unavailable. CLAVIER almost always uses several cases to do its design. One provides an overall layout, which is adapted appropriately. The others are used to fill in holes in the layout that adaptation rules by themselves cannot cover.

In almost all design problems, more than one case is necessary to solve the problem. Design problems tend to be large, and while one case can be used to solve some of it, it is usually not sufficient for solving the whole thing. In general, one case provides a framework for a solution and other cases are used to fill in missing details. In this way, decomposition and recomposition are avoided, as are large constraint satisfaction (qv) and relaxation problems.

CBR for Planning. Planning involves a number of complexities. Charniak and McDermott (1985) provide an excellent overview. Good plans must be sequenced appropriately so that late steps in a plan do not undo the intended results of earlier steps, preconditions of late steps in a plan are not violated by the results of earlier ones, and preconditions of later plan steps are fulfilled before the step is scheduled. As the number of plan steps increases, the computational complexity of projecting effects and comparing preconditions increases exponentially. In addition, a planner that must interact with the real world must deal with the real world's complexity, including the fact that it is in many ways unpredictable and that time is not limitless. Streams of goals might need to be achieved almost simultaneously. Time used for planning can take away from the time available for execution. Because conditions in the world can change between developing a plan and carrying it out, a plan might fail at execution time and require replanning, recovery, or repair. A planner with little time might miss opportunities during planning that can be better noticed and taken advantage of during execution. See Marks and co-workers (1988) for better explanations of these problems. Case-based reasoning can address many of these planning issues.

- Cases provide already worked-out plans in which sequencing, protection maintenance, and scheduling of preconditions have already been worked out. Rather than reasoning from scratch, the planner is required only to make repairs in old plans.
- If cases are indexed by the conjunctions of goals they achieve, they can be used to suggest ways of achieving several goals simultaneously or in conjunction with each other.
- Warnings provided by cases can help a planner anticipate and avoid problems, decreasing the likelihood of failure at execution time.
- Adaptation strategies used to adapt old plans to new situations can be used for execution-time recovery and repair.
- Suggestions made by cases shortcut the planning process, providing relatively more time for execution.
- Suggestions made by cases allow the reasoner to notice some opportunities (eg, to achieve goals simultaneously) more easily during planning.
- A case-based plan executor can notice opportunities during execution and use its adaptation strategies to update its plan accordingly.

Case-based reasoners are addressing many of these issues. PLEXUS (Alterman, 1986, 1988), a program that knows how to ride a subway, is able to do execution-time repairs by adapting and substituting semantically similar steps for those that have failed.

CHEF (Hammond, 1986, 1989a), one of the earliest case-based planners, addresses the problem of anticipating problems before execution time by learning from its problematic experiences. When problems happen at execution time, CHEF attempts to explain them and then to figure out how they could be repaired. It stores its hypothesized repair in memory and indexes the case by features that are likely to predict that the problem will recur. Before it begins plan derivation, it looks for failure situations and uses any it finds to anticipate the problems they point out. Later, it uses the repaired failure situations to suggest a plan that will avoid the problem it has anticipated.

Case-based planners that address some of the other problems mentioned above have also been built. TRUCKER (Hammond, 1989b) is an errand-running program that keeps track of its pending goals and is able to take advantage of opportunities that arise that allow it to achieve goals earlier than expected. MEDIC (qv) (Turner, 1989) is a diagnosis program. It is able to reuse previous plans for diagnosis but is flexible enough in its reuse to be able to follow up on unexpected turns of events. EXPEDITOR (Robinson and Kolodner, 1991) plans the events in the life of a single parent who must deal with kids and work. It caches its experiences achieving multiple goals by interleaving them. While it is slow in its initial planning, it gains competence over time as it is able to reuse its plans. The CSI BATTLE PLANNER (Goodman, 1989) shows how cases can be used to criticize and repair plans before they are executed.

CBR for Diagnosis. In diagnosis, a problem solver is given a set of symptoms and asked to explain them. A case-based diagnostician can use cases to suggest explanations for symptoms and to warn of explanations that have been found to be inappropriate in the past. Of course, one cannot expect a previous diagnosis to apply intact to the new case. Just as in planning and design, it is often necessary to adapt an old diagnosis to fit a new situation. CASEY (Koton, 1988a) was able to diagnose heart problems by adapting the diagnoses of previous heart patients to new patients. CASEY is a relatively simple program built on top of an existing model-based diagnostic program. When a new case is similar to one it has seen previously, it is several orders of magnitude more efficient at generating a diagnosis than is the model-based program (Koton, 1988a). CASEY's adaptations are based on a valid causal model (CASEY uses model-guided repair as its method of adaptation). Thus, its diagnoses are as accurate as those made from scratch based on the same causal model.

Cases are also useful in diagnosis in pointing the way out of previously-experienced reasoning quagmires. PROTOS (qv) (Bareiss, 1989) is designed to ensure that this happens in an efficient way. PROTOS diagnoses hearing disorders. In this domain, many of the diagnoses manifest themselves in similar ways and are difficult to differentiate. While novices are not aware of these subtle differences, experts are. PROTOS begins as a novice, and when it makes mistakes, a "teacher" explains its mistakes to it. As a result, PROTOS learns these subtle differences. As it does, it leaves difference pointers in its memory that allow it to move easily from the obvious diagnosis to the correct one.

Generating a diagnosis from scratch is a time-consuming task. In almost all diagnostic domains, however, there is sufficient regularity for a case-based approach to diagnosis generation to provide efficiency. Of course, no person or program can assume that a case-based suggestion is correct. The case-based suggestion must be validated. Often, however, validation is much easier than generation. In those kinds of domains, case-based reasoning can provide big wins.

Interpretive CBR

Interpretive case-based reasoning is a process of evaluating situations or solutions in the context of previous experience. It takes a situation or solution as input, and its output is a classification of the situation, an argument supporting the classification or solution, and/or justifications supporting the argument or solution. It is useful for situation classification, evaluation of a solution, argumentation, justification of a solution, interpretation, or plan, and projection of the effects of a decision or plan.

Interpretive case-based reasoning is most useful for evaluation when there are no computational methods available to evaluate a solution or position. Often, in these situations, there are so many unknowns that even if computational methods were available, the knowledge necessary to run them would usually be absent. A reasoner who uses cases to help evaluate and justify decisions or interpretations is making up for his lack of knowledge by assuming that the world is consistent.

Justification and Adversarial Reasoning. Adversarial reasoning means making persuasive arguments to convince others that we or our positions are right. A persuasive argument states a position and supports it, sometimes with hard facts and sometimes with valid inferences. But often the only way to justify a position is by citing relevant previous experiences or cases. Law thus provides a good domain for the study of adversarial reasoning and case-based justification for this reason, and much research in this area uses the legal domain (Ashley, 1987, 1988; Bain, 1986; Branting, 1989; Rissland, 1983).

HYPO (Ashley, 1987, 1988; Rissland, 1986) is the earliest and most sophisticated of the case-based legal reasoners. HYPO's method's for creating an argument and justifying a solution or position has several steps. First, the new situation is analyzed for relevant factors. Based on these factors, similar cases are retrieved. They are positioned with respect to the new situation. Some support it and some are against it. The most on-point cases of both sets are selected. The most on-point case supporting the new situation is used to create an argument for the proposed solution. Those in the nonsupport set are used to pose counter-arguments. Cases in the support set are then used to counter the counter-arguments. The result of this is a set of three-ply arguments in support of the solution, each of which is justified with cases. An important side effect of creating such arguments is that potential problem areas get highlighted.

In general, cases are useful in constructing arguments and justifying positions when there are no concrete principles or only a few of them, if principles are inconsistent, or if their meanings are not well-specified.

Classification and Interpretation. Interpretation in the context of case-based reasoning means deciding whether a concept fits some open-ended or fuzzy-bordered classification. The classification might be derived on the fly based on the task at hand or it might be well known but not well-defined in terms of necessary and sufficient conditions. Many of the classifications assumed to be defined are classifications of the open-ended variety. For example, it is assumed that a vehicle means a thing with wheels used for transportation, but when a sign says "No vehicles in the park," it is probably not referring to a wheelchair or a baby stroller, both of which fit our simple definition.

One way a case-based classifier works is to ask whether the new concept is enough like another one known to have the target classification. PROTOS (Bareiss, 1989), which diagnoses hearing disorders, works like this. Rather than classifying new cases using necessary and sufficient conditions, PROTOS does classification by trying to find the closest matching case in its case base to the new situation. It classifies the new situation by that case's classification. To do this, PROTOS keeps track of how prototypical each of its cases is and what differentiates cases within one classification from each other. It first chooses a most likely classification, then chooses a most likely matching case in that class. Based on differences between the case it is attempting to match and the new situation, it eventually zeros in on a case that matches its new one well.

When no case matches well enough, it is sometimes necessary to consider hypothetical situations. Much of the work on this type of interpretation also comes from the study of legal reasoning. HYPO (Ashley, 1987, 1988; Rissland, 1986) uses hypotheticals for a variety of tasks necessary for good interpretation: to redefine old situations in terms of new dimensions, to create new standard cases when a necessary one does not exist, to explore and test the limits of reasonableness of a concept, to refocus a case by excluding some issues, to tease out hidden assumptions, and to organize or cluster cases. HYPO creates hypotheticals by making "copies" of a current situation that are stronger or weaker than the real situation for one side or the other. This work is guided by a set of modification heuristics that propose useful directions for hypothetical case creation based on current reasoning needs. HYPO's strategies for argumentation guide selection of modification heuristics. For example, to counter a counterexample, one might propose variations on a new situation that make it more like the counterexample.

Projecting Effects. Projection, the process of predicting the effects of a decision or plan, is an important part of the evaluative component of any planning or decision making scheme. When everything about a situation is known, projection is merely a process of running known inferences forward from a solution to see where it leads. More often, however, in real-world problems, everything is not known and effects cannot be predicted with accuracy based on any simple set of inference rules.

Cases provide a way of projecting effects based on what has been true in the past. Cases with similar plans that were failures can point to potential plan problems. Cases with similar plans that were successes give credence to the current plan. In addition, when parts of a plan are targeted for evaluation, cases can help with that.

Automated use of cases for projection has not been a focus of case-based reasoning research, but aid to a person doing projection is being addressed. CSI's BATTLE PLANNER (Goodman, 1989) is a case-retrieval system whose interface is set up to allow a person to use cases to project effects. A student commander can propose a solution plan to the system. The BATTLE PLANNER retrieves the best-matching cases that use a similar plan and divides them into success and failure situations. The person can examine the cases, use them to fix a plan, and then attempt a similar evaluation of the repaired plan. Or, the person can use the system to do a sensitivity analysis. By manipulating the details of the situation and looking at the changes in numbers of wins and losses (in effect, asking a series of "what-if" questions), he or she can determine which factors of the current situation are the crucial ones to repair and which should be left unchanged.

Interpretive Case-Based Reasoning and Problem Solving. Much work on interpretation has centered on the law domain and has looked at justifying an argument for or against some interpretation of the law. Case-based interpretation is not merely for interpretive problems, however. It is very useful as part of the evaluative or critical component of problem solving and decision making whenever strong causal models are missing. Though there has been little work in this area, the processes involved in interpretive case-based reasoning have the potential to play several important roles for a problem solver. First, if the framework for a solution is known, or if constraints governing it are known, these methods could be used to choose cases that would provide such a solution. Second, argument creation and justification result in knowledge of what features are the important ones to focus on. Knowing where to focus is important in problem solving also. Third, a side effect of HYPO's methods is that it can point out which features, if they were present, would yield a better solution. It does this by keeping track of near-miss dimensions and creation of hypothetical cases. A problem solver could use such information to inform its adaptation processes. Finally, interpretive methods can be used to predict the usefulness, quality, or results of a solution.

IMPLICATIONS OF CBR AS A COGNITIVE MODEL

One goal of building CBR systems is to attempt to understand the processes involved in reasoning in a case-based way. Psychologists who study analogical reasoning are investigating similar processes. There are several important potential applications of an understanding of the way people solve problems in a natural way.

- Decision support systems. This understanding can be used to help build decision aiding systems for people that can help them retrieve cases better. Psychologists have found that people are comfortable using cases to make decisions but do not always remember the right ones. The computer could be used as a retrieval tool to augment people's memories. (See the following section for further discussion on this application of CBR.)
- Teaching as providing cases. An understanding of human case-based reasoning might allow us to create teaching strategies and build teaching tools that teach based on good examples. If people are comfortable using examples to solve problems and know how to do it well, then one of our responsibilities as teachers might be to teach them the right ones. Systems that teach in a case-based way have recently begun to be developed (Burke and Ohmaye, 1990; Schank, 1991).
- Teaching the process of CBR. If it is understood which parts of this natural process are difficult to do well, people can be taught how to do case-based reasoning better. One criticism of using cases to make decisions, for example, is that it puts unsound bias into the reasoning system, because people tend to assume an answer from a previous case is right without justifying it in the new case. This says that people should be taught how to justify case-based suggestions and that justification or evaluation is crucial to good decision making. If other problems people have in solving problems in a case-based way can be isolated, then people can be taught to do those things better.

BUILDING CBR SYSTEMS

There are many reasons one might want to build a CBR system. It might be needed to solve problems, to suggest concrete answers to problems, to be suggestive without providing answers (ie, to give abstract advice), or to just act as a database that can retrieve partially-matching cases. This suggests several different kinds of case-based reasoning systems that might be built. At the two extremes are fully-automated systems and retrieval-only systems. Fully automated systems are those that solve problems completely by themselves and have some means of interacting with the world to receive feedback on their decisions. Retrieval-only systems work interactively with a person to solve a problem. The role of such systems is just to augment a person's memory, providing cases for consideration that he or she might not have been aware of; the user is left responsible for doing the reasoning and making the hard decisions. The CSI BATTLE PLANNER (Goodman, 1989) provides this type of capability now for battle planning. Then there is the whole range of systems in between, some requiring more on the part of the person using the system, some less.

How might one go about building a CBR system? What is required for the simplest of systems is a library of cases that coarsely cover the set of problems that come up in a domain. Both success stories and failures must be included and the cases must be appropriately indexed. This library, along with a friendly and useful interface, can act as an "expert assistant" by augmenting the memory of a human user. Once a system consisting of the case library and user interface has been built, automated reasoning and problem-solving processes can then be added incrementally.

SUMMARY

The case-based reasoning paradigm grew out of research on human memory and a growing dissatisfaction with rule-based systems. CBR has several advantages as a theory and methodology of problem-solving, including psychological plausibility, easier knowledge acquisition, and robustness. In addition, CBR provides a natural mechanism for incorporating learning, offering such advantages as performance enhancements and scalability.

The basic method of CBR is to adapt past solutions to solve a current problem. This involves extracting indexes from the input and using these indexes to retrieve relevant cases from memory. After narrowing down the set of retrieved cases to a few most worthy of consideration, a CBR system adapts these cases to form a solution to the current problem. This solution is evaluated and, if acceptable, it is executed in the real world. The system then receives feedback about the success or failure of its solution and may modify the solution in response to an analysis of this feedback. Once the solution is acceptable, it is added to memory to be used to solve similar problems in the future.

This basic algorithm gives rise to many interesting issues which are currently being researched in the CBR community. Some of the issues discussed here are indexing vocabularies, memory organization, retrieval algorithms, similarity metrics, and case adaptation. While much progress has been made, a great deal more work remains to be done, making it likely that the CBR paradigm will continue to be an active area of research in the years to come.

CBR systems have been designed to perform a wide variety of both problem-solving and interpretive tasks. Design, planning, and diagnosis are three areas where problem-solving CBR has been applied. Systems doing interpretive CBR have been built to do justification and adversarial reasoning, classification, and interpretation tasks, as well as projecting the effects of plans. Given its success in such a diverse set of domains and tasks, it is not surprising that case-based reasoning continues to enjoy a great deal of popularity as a reasoning and problem solving paradigm.

BIBLIOGRAPHY

R. Alterman, "An Adaptive Planner," *Proceedings Fifth National Conference on Artificial Intelligence*, Philadelphia, Pa., AAAI, Menlo Park, Calif., 1986, pp. 65–69.

R. Alterman, "Adaptive Planning," *Cogn. Sci.* **12**, 393–422 (1988).

K. D. Ashley, "Distinguishing—A Reasoner's Wedge," *Proceedings of the 1987 Conference of the Cognitive Science Society*, Lawrence Erlbaum, Hillsdale, N.J., 1987, pp. 737–747.

K. D. Ashley, *Modeling Legal Argument: Reasoning with Cases and Hypotheticals*, Ph.D. dissertation, COINS Technical Report No. 88–01, Department of Computer and Information Science, University of Massachusetts, Amherst, Mass., 1988.

K. Ashley, "Assessing Similarities Among Cases: A Position Paper," *Proceedings: Case-Based Reasoning Workshop (DARPA), II*, Morgan-Kaufmann, San Mateo, Calif., 1989, pp. 72–75.

K. Ashley and E. Rissland, "Waiting on Weighting: A Symbolic Least Commitment Approach," in *Proceedings of the Seventh National Conference on Artificial Intelligence*, St. Paul, Minn., AAAI, Menlo Park, Calif., 1988, pp. 239–244.

W. Bain, *Case-Based Reasoning: A Computer Model of Subjective Assessment*, Ph.D. dissertation, Yale University, New Haven, Conn., 1986.

E. R. Bareiss, *Exemplar-Based Knowledge Acquisition: A Unified Approach to Concept Representation, Classification, and Learning*, Academic Press, Inc., Boston, Mass., 1989.

R. Bareiss and J. King, "Similarity Assessment in Case-Based Reasoning," *Proceedings: Case-Based Reasoning Workshop (DARPA), II*, Morgan-Kaufmann, San Mateo, Calif., 1989, pp. 67–71.

R. Barletta and D. Hennessy, "Case Adaptation in Autoclave Layout Design," *Proceedings: Case-Based Reasoning Workshop (DARPA), II*, Morgan-Kaufmann, San Mateo, Calif., 1989, pp. 203–207.

L. Birnbaum and G. Collins, "Remindings and Engineering Design Themes: A Case Study in Indexing Vocabulary," *Proceedings: Case-Based Reasoning Workshop (DARPA), II*, Morgan-Kaufmann, San Mateo, Calif., 1989, pp. 47–51.

L. K. Branting, "Integrating Generalizations with Exemplar-Based Reasoning," *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society*, Lawrence Erlbaum, Hillsdale, N.J., 1989, pp. 139–146.

R. Burke and E. Ohmaye, "Case-Based Environments for Learning," in B. Woolf and E. Soloway, eds., *Knowledge-based Environments for Learning and Teaching*, Symposium conducted at AAAI Spring Symposium Series, Stanford University, Stanford, Calif., 1990.

E. Charniak and D. McDermott, *Introduction to Artificial Intelligence*, Addison-Wesley, Reading, Mass., 1985.

G. Collins, *Plan Creation: Using Strategies as Blueprints*, Ph.D. dissertation, Yale University, New Haven, Conn., 1987.

G. Collins, "Plan Adaptation: A Transformational Approach," *Proceedings: Case-Based Reasoning Workshop (DARPA), II*, Morgan-Kaufmann, San Mateo, Calif., 1989, pp. 47–51.

G. Collins and L. Birnbaum, "Problem-Solver State Descriptions as Abstract Indices for Case Retrieval," in *Working Notes of the 1990 AAAI Spring Symposium on Case-Based Reasoning*, Stanford, Calif., March 1990.

E. Domeshek, "Parallelism for Index Generation and Reminding," *Proceedings: Case-Based Reasoning Workshop (DARPA), II*, Morgan-Kaufmann, San Mateo, Calif., 1989, pp. 244–247.

E. Domeshek, *Case-Based Advising in the Social Domain: Representation, Indexing and Retrieval*, Ph.D. dissertation, Yale University, New Haven, Conn., (in preparation).

E. Feigenbaum, "The Simulation of Verbal Learning Behavior," in E. Feigenbaum and J. Feldman, eds., *Computers and Thought*, McGraw-Hill, New York, 1963, pp. 297–309.

D. Gentner, "Finding the Needle: Accessing and Reasoning from Prior Cases," *Proceedings: Case-Based Reasoning Workshop (DARPA), II*, Morgan-Kaufmann, San Mateo, Calif., 1989, pp. 137–143.

A. Goel, *Integration of Case-Based Reasoning and Model-Based Reasoning for Adaptive Design Problem Solving*, Ph.D. dissertation, Ohio State University, Columbus, Ohio, 1989.

A. Goel and B. Chandrasekaran, "Use of Device Models in Adaptation of Design Cases," *Proceedings: Case-Based Reasoning Workshop (DARPA), II*, Morgan-Kaufmann, San Mateo, Calif., 1989, pp. 100–109.

V. Goel and P. Pirolli, "Motivating the Notion of Generic Design within Information-Processing Theory: The Design Problem Space," *AI Mag.* **10**(1), 18–36 (1989).

M. Goodman, "CBR in Battle Planning," *Proceedings: Case-Based Reasoning Workshop (DARPA), II*, Morgan-Kaufmann, San Mateo, Calif., 1989, pp. 264–269.

K. Hammond, "CHEF: A Model of Case-Based Planning," *Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, Pa., AAAI, Menlo Park, Calif., 1986, pp. 267–271.

K. J. Hammond, *Case-Based Planning: Viewing Planning as a Memory Task*, Academic Press, Inc., Boston, Mass., 1989a.

K. Hammond, "Opportunistic Memory," *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Detroit, Mich., Morgan-Kaufmann, San Mateo, Calif., 1989b, pp. 504–510.

K. Hammond, "On Functionally Motivated Vocabularies: An Apologia," *Proceedings: Case-Based Reasoning Workshop (DARPA), II*, Morgan-Kaufmann, San Mateo, Calif., 1989c, pp. 52–56.

F. Hayes-Roth, D. A. Waterman, and D. B. Lenat, eds., *Building Expert Systems*, Addison-Wesley, Reading, Mass., 1983.

T. R. Hinrichs, "Towards an Architecture for Open World Problem Solving," in J. Kolodner, ed., *Proceedings: Case-Based Reasoning Workshop (DARPA)*, Morgan-Kaufmann, San Mateo, Calif., 1988, pp. 182–189.

T. Hinrichs, "Strategies for Adaptation and Recovery in a Design Problem Solver," *Proceedings: Case-Based Reasoning Workshop (DARPA), II*, Morgan-Kaufmann, San Mateo, Calif., 1989, pp. 115–118.

K. Holyoak and K. Koh, "Surface and Structural Similarity in Analogical Transfer," *Memory & Cognition* **15**, 332–340 (1987).

L. Hunter, "Finding Paradigm Cases or When is a Case Worth Remembering?" *Proceedings: Case-Based Reasoning Workshop (DARPA), II*, Morgan-Kaufmann, San Mateo, Calif., 1989, pp. 57–61.

A. Kass, "Modifying Explanations to Understand Stories," *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, Lawrence Erlbaum, Hillsdale, N.J., 1986, pp. 691–696.

A. Kass, "Adaptation-Based Explanation: Extending Script/Frame Theory to Handle Novel Input," in *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Detroit, Mich., Morgan-Kaufmann, San Mateo, Calif., 1989, pp. 141–147.

A. Kass, *Developing Creative Hypothesis by Adapting Explanations*, Ph.D. dissertation, Yale University, New Haven, Conn., 1990. Reprinted as Technical Report 6, Institute for the Learning Sciences, Northwestern University, Evanston, Ill.

A. M. Kass and D. B. Leake, "Case-Based Reasoning Applied to Constructing Explanations," in J. Kolodner, ed., *Proceedings: Case-Based Reasoning Workshop (DARPA)*, Morgan-Kaufmann, San Mateo, Calif., 1988, pp. 190–208.

A. M. Kass, D. B. Leake, and C. C. Owens, "Swale: A Program That Explains," in R. C. Schank, ed., *Explanation Patterns: Understanding Mechanically and Creatively*, Lawrence Erlbaum, Hillsdale, N.J., 1986, pp. 232–254.

G. A. Klein and R. Calderwood, "How Do People Use Analogues to Make Decisions?" in J. Kolodner, ed., *Proceedings: Case-Based Reasoning Workshop (DARPA)*, Morgan-Kaufmann, San Mateo, Calif., 1988, pp. 209–218.

J. L. Kolodner, "Reconstructive Memory: A Computer Model," *Cogn. Sci.* **7**, 281–328 (1983a).

J. Kolodner, "Towards an Understanding of the Role of Experience in the Evolution from Novice to Expert," *Int. J. Man Machine Studies* **19**, 497–518 (1983b).

J. L. Kolodner, *Retrieval and Organization Strategies in Conceptual Memory: A Computer Model*, Lawrence Erlbaum, Hillsdale, N.J., 1984.

J. Kolodner, "Towards a Memory Architecture That Supports Reminding," *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, Lawrence Erlbaum, Hillsdale, N.J., 1986, pp. 467–477.

J. L. Kolodner, "Capitalizing on Failure Through Case-Base Inference," *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*, Lawrence Erlbaum, Hillsdale, N.J., pp. 715–726, 1987.

J. L. Kolodner, "Retrieving Events from a Case Memory: A Parallel Implementation," in J. Kolodner, ed., *Proceedings: Case-Based Reasoning Workshop (DARPA)*, Morgan-Kaufmann, San Mateo, Calif., 1988, pp. 233–240.

J. Kolodner, "Judging Which is the Best Case for a Case-based Reasoner," *Proceedings: Case-Based Reasoning Workshop (DARPA), II*, Morgan-Kaufmann, San Mateo, Calif., 1989, pp. 77–81.

J. Kolodner, *An Introduction to Case-based Reasoning*, Technical Report No. GIT-ICS-90/19, College of Computing, Georgia Institute of Technology, Atlanta, Ga., 1990.

J. Kolodner, "Improving Human Decision Making Through Case-based Decision Aiding," *AI Mag.* **12**(2), 52–68 (1991).

J. L. Kolodner and R. L. Simpson, "The MEDIATOR: Analysis of an Early Case-based Problem Solver," *Cog. Sci.* **13**, 507–549 (1989).

P. Koton, *Using Experience in Learning and Problem Solving*, Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, Mass., 1988a.

P. Koton, "Reasoning about Evidence in Causal Explanations," *Proceedings of the Seventh National Conference on Artificial Intelligence*, St. Paul, Minn., AAAI, Menlo Park, Calif., 1988b, pp. 256–261.

J. S. Lancaster and J. L. Kolodner, "Varieties of Learning from Problem Solving Experience," in *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*, Lawrence Erlbaum, Hillsdale, N.J., 1988, pp. 447–453.

D. Leake and C. Owens, "Organizing Memory for Explanations," in *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, Lawrence Erlbaum, Hillsdale, N.J., 1986, pp. 710–715.

M. Lebowitz, "Generalization from Natural Language Text," *Cog. Sci.* **7**, 1–40 (1983).

M. Marks, K. J. Hammond, and T. Converse, "Planning in an Open World: A Pluralistic Approach," in J. Kolodner, ed., *Proceedings: Case-Based Reasoning Workshop (DARPA)*, Morgan-Kaufmann, San Mateo, Calif., 1988, pp. 271–285.

C. Martin, "Indexing Using Complex Features," *Proceedings: Case-Based Reasoning Workshop (DARPA), II*, Morgan-Kaufmann, San Mateo, Calif., 1989a, pp. 26–30.

C. Martin, "Complex Indices: A Metaphorical Example," *Proceedings: Case-Based Reasoning Workshop (DARPA), II*, Morgan-Kaufmann, San Mateo, Calif., 1989b, pp. 295–299.

D. Navinchandra, "Case-Based Reasoning in CYCLOPS, a Design Problem Solver," in J. Kolodner, ed., *Proceedings: Case-Based Reasoning Workshop (DARPA)*, Morgan-Kaufmann, San Mateo, Calif., 1988, pp. 286–301.

C. Owens, "Domain-Independent Prototype Cases for Planning in J. Kolodner, ed., *Proceedings: Case-Based Reasoning Workshop (DARPA)*, Morgan-Kaufmann, San Mateo, Calif., 1988, pp. 302–311.

C. Owens, "Plan Transformations as Abstract Indices," *Proceedings: Case-Based Reasoning Workshop (DARPA), II*, Morgan-Kaufmann, San Mateo, Calif., 1989a, pp. 62–65.

C. Owens, "Integrating Feature Extraction and Memory Search," *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society*, Lawrence Erlbaum, Hillsdale, N.J., 1989b, pp. 163–170.

M. Pazzani, "Indexing Strategies for Goal Specific Retrieval of Cases," *Proceedings: Case-Based Reasoning Workshop (DARPA), II*, Morgan-Kaufmann, San Mateo, Calif., 1989, pp. 52–56.

B. Porter, "Similarity Assessment: Computation vs Representation," *Proceedings: Case-Based Reasoning Workshop (DARPA), II*, Morgan-Kaufmann, San Mateo, Calif., 1989, pp. 82–84.

S. Read and I. Cesa, "This Reminds Me of the Time When . . . : Expectation Failures in Reminding and Explanation," *J. Experimental Social Psych.* **27,** 1–25, (1991).

M. Redmond, "Combining Explanation Types for Learning by Understanding Instructional Examples," *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society,* Lawrence Erlbaum, Hillsdale, N.J., 1989, pp. 147–154.

M. Redmond, "Distributed Cases for Case-Based Reasoning; Facilitating Uses of Multiple Cases," *Proceedings of the Eighth National Conference on Artificial Intelligence,* AAAI, Menlo Park, Calif., 1990, pp. 304–309.

C. Riesbeck and C. Martin, *Direct Memory Access Parsing,* Technical Report YALEU/CSD/RR 354, Yale University, New Haven, Conn., 1985.

C. K. Reisbeck and R. S. Schank, *Inside Case-Based Reasoning,* Lawrence Erlbaum, Hillsdale, N.J., 1989.

E. L. Rissland, "Examples in Legal Reasoning: Legal Hypotheticals," *Proceedings of the Eighth International Joint Conference on Artificial Intelligence,* Karlsruhe, FRG, Morgan-Kaufmann, San Mateo, Calif., 1983, pp. 90–93.

E. L. Rissland, "Learning How to Argue: Using Hypotheticals," in J. L. Kolodner and C. K. Riesbeck, eds., *Experience, Memory, and Reasoning,* Lawrence Erlbaum, Hillsdale, N.J., 1986, pp. 115–126.

E. L. Rissland and K. D. Ashley, "Hypotheticals as a Heuristic Device," *Proceedings of the Fifth National Conference on Artificial Intelligence,* Philadelphia, Pa., AAAI, Menlo Park, Calif., 1986, pp. 289–297.

E. Rissland and K. Ashley, "Credit Assignment and the Problem of Competing Factors in Case-Based Reasoning," in J. Kolodner, ed., *Proceedings: Case-Based Reasoning Workshop (DARPA),* Morgan-Kaufmann, San Mateo, Calif., 1988, pp. 327–344.

S. M. Robinson and J. L. Kolodner, "Indexing Cases for Planning and Acting in Dynamic Environments: Exploiting Hierarchical Goal Structures," *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society,* Lawrence Erlbaum, Hillsdale, N.J., 1991.

B. H. Ross, "Remindings in Learning and Instruction," in S. Vosniadou and A. Ortony, eds., *Similarity and Analogical Reasoning.* Cambridge University Press, New York, 1989a, pp. 438–469.

B. H. Ross, "Some Psychological Results on Case-based Reasoning," *Proceedings: Case-Based Reasoning Workshop (DARPA), II,* Morgan-Kaufmann, San Mateo, Calif., 1989b, pp. 144–147.

R. Schank, *Dynamic Memory: A Theory of Learning in Computers and People,* Cambridge University Press, New York, 1982.

R. C. Schank, *Case-Based Teaching: Four Experiences in Educational Software Design,* Technical Report No. 7, Institute for the Learning Sciences, Northwestern University, Evanston, Ill., 1991.

R. Schank and co-workers, "Towards a General Content Theory of Indices," in *Working Notes of the AAAI Spring Symposium on Case-Based Reasoning,* American Association for Artificial Intelligence, Stanford, Calif., 1990, pp. 36–40.

R. L. Simpson, *A Computer Model of Case-Based Reasoning in Problem Solving: An Investigation in the Domain of Dispute Mediation,* Ph.D. dissertation, Technical Report No. GIT-ICS-85/18, School of Information and Computer Science, Georgia Institute of Technology, Atlanta, Ga., 1985.

S. Slade, "Case-Based Reasoning: A Research Paradigm," *AI Mag.* **12**(1), 42–55 (1991).

C. Stanfill, "Memory-Based Reasoning Applied to English Pronunciation," in *Proceedings of the Sixth National Confere^nce on Artificial Intelligence,* AAAI, Menlo Park, Calif., 1987, pp. 577–581.

C. Stanfill, and D. L. Waltz, "The Memory-Based Reasoning Paradigm," in J. Kolodner, ed., *Proceedings: Case-Based Reasoning Workshop (DARPA),* Morgan-Kaufmann, San Mateo, Calif., 1988, pp. 414–424.

E. P. Sycara, *Resolving Adversarial Conflicts: An Approach to Integrating Case-based and Analytic Methods,* Ph.D. dissertation, Technical Report No. GIT-ICS-87/26, School of Information and Computer Science, Georgia Institute of Technology, Atlanta, Ga., 1987.

P. Thagard and K. Holyoak, "How to Compute Semantic Similarity," *Proceedings: Case-Based Reasoning Workshop (DARPA), II,* Morgan-Kaufmann, San Mateo, Calif., 1989a, pp. 85–86.

P. Thagard and K. Holyoak, "Why Indexing is the Wrong Way to Think About Analog Retrieval," *Proceedings: Case-Based Reasoning Workshop (DARPA), II,* Morgan-Kaufmann, San Mateo, Calif., 1989b, pp. 36–40.

R. M. Turner, *A Schema-based Model of Adaptive Problem Solving,* Ph.D. dissertation, Technical Report No. GIT-ICS-89/42. School of Information and Computer Science, Georgia Institute of Technology, Atlanta, Ga., 1989.

D. Waltz, "Is Indexing Used for Retrieval?" *Proceedings: Case-Based Reasoning Workshop (DARPA), II,* Morgan-Kaufmann, San Mateo, Calif., 1989, pp. 41–44.

L. Whitaker, S. Wiggins, and G. Klein, "Using Qualitative or Multi-Attribute Similarity to Retrieve Useful Cases from a Case Base," *Proceedings: Case-Based Reasoning Workshop (DARPA), II,* Morgan-Kaufmann, San Mateo, Calif., 1989, pp. 345–347.

MENACHEM Y. JONA
Northwestern University
JANET L. KOLODNER
Georgia Institute of Technology

This work was supported in part by the Defense Advanced Research Projects Agency, monitored by the Air Force Office of Scientific Research under contract F49620-88-C-005; by the NSF under grant IST-8608362; and by ARI under contracts MDA-903-86-C-173 and MDA-903-90-K-0112. The Institute for the Learning Sciences was established in 1989 with the support of Andersen Consulting, part of The Arthur Andersen Worldwide Organization. The authors thank Marie Paro for helpful comments on a draft of this article.

particular domains. Examples include diagnosing medical problems or circuit faults, simulating physical systems, understanding stories, and planning robotic tasks. Most AI research involving causal reasoning falls into this "applied" category. In contrast, *theories of causation* focus on understanding the nature of causal relations and causal reasoning per se. Traditionally, such analyses have been the concern of philosophers of science, statisticians, and cognitive scientists. Recently, AI researchers have developed computational theories of causation, typically cast as formal logics. These formalisms define domain-independent logical representations for causal relationships in the world and automated deduction algorithms that capture causal inferences.

Causal models can be partitioned into two subclasses based on how they depict causation: explicitly or implicitly. Explicit causal models draw behavioral inferences in terms of structures that are specifically and uniquely interpreted as causal relations. Typically, causal links (eg, "causes," "possibly causes," or "causes with probability x"), are used to connect states or events in a chain or network. Causal reasoning in these systems hinges on tracing the links signifying causal relations from known states to other states that represent either future behavior (prediction) or elaborations of already observed situations (explanation).

In contrast, causal knowledge in implicit models is either: (1) not represented explicitly; or (2) modeled using explicit structures that do not have uniquely causal interpretations. Approach 1 is typified by symbolic simulators of mechanical or electronic devices: the algorithms that generate model system behaviors implicitly reflect causal structure in the world, but the simulators make no essential reference to causation at all. Most logic-based planning systems exemplify approach 2: axioms, called causal or projection rules, specify the behavioral consequences of basic model actions such as picking up and moving blocks about a table. Planning systems apply these predictive axioms to determine whether a given action will help to achieve desired goals. However, reasoning based on such "causal" axioms is not explicitly distinguished from inferences derived from noncausal axioms (eg, that prohibit simultaneous actions or two objects cooccupying a single location.)

This categorization captures important distinctions in the field although, like most organizing frameworks, it is somewhat broad. Causal modelers frequently make important observations about the nature of causation, while causal theorists sometimes apply their accounts to construct causal models in particular domains. The following sections review AI and related literature on causal models and theories. A brief critical assessment of the field and future research directions concludes the entry.

EXPLICIT CAUSAL MODELS

Explicit causal models often depict causal relationships as links between nodes in a network that represent states, state changes, events, or actions. These models have been explored most extensively in two quite different contexts: diagnosis of medical problems or circuit faults and story understanding. In both domains, the goals of causal inference are to (1) map available observations onto network nodes and (2) activate additional nodes as necessary to construct a subnetwork of states and events linked in a causally coherent pattern. This subnetwork constitutes a causal explanation or prediction for the observed situation.

Rieger and Grinberg (1977) introduced the first extensive causal-link representation. Their goal was to express causal knowledge of the operation of a complex mechanism sufficient to simulate its behavior. Their model encompasses types of nodes that represent actions, tendencies, states, and state changes, together with link types that depict primitive causal relations among these nodes. Connective link types include causality, enablement, and concurrency. Each node and causal link in a mechanism description is implemented as an independent computational agent. Activation percolates through the network according to the semantics of each type of link, in a temporal sequence intended to correspond to the sequence of states, actions, or state changes in the behavior of the mechanism. The simulator can, however, introduce spurious temporal ordering relations among events on independent causal paths; no provisions were made for detecting or correcting these anomalies. Rieger and Grinberg demonstrated this apparatus by building structural models and behavioral simulations for several nontrivial mechanisms, including the forced hot-air furnace and the flushing toilet.

Research on story understanding (see STORY ANALYSIS) employs causal reasoning to reconstruct complete and coherent scenarios from narrative fragments such as "Joe burned his hand because he forgot the stove was on." Schank parses stories into complex symbolic networks of acts and their participants (Schank and Abelson, 1977). An elaborate grammar specifies legal constructions of objects, actors, and primitive "act" types such as PROPEL, GRASP, and PTRANS (physical transfer of objects). Stereotypic activities such as dining at a restaurant are captured in predefined network templates called scripts (qv). Scripts establish an expected sequencing of acts, including nominal and exceptional "branching" situations, and their relationships to actors' plans and goals. Shank's program extends parsed story networks through a set of inference patterns, enabling causal questions (eg, how and why), to be answered about elements unmentioned in the input narrative. In particular, causality inference patterns generate explicit causal links from conjoined sequences of acts such as "John hit Bob and he fell." Similarly, belief and intention patterns ground reasoning about the mental states and causal motivations of actors in the story. Subsequent research has explored various alternative modeling frameworks and inference techniques (Dyer, 1983); however, reasoning about the causal content of stories remains a central issue.

As in story understanding, the purpose of causal reasoning in medical diagnosis (see MEDICINE, AI IN) is to determine a causally coherent set of events that has taken place and that matches the available observations. The explicit causal models embodied in diagnostic systems typically employ simpler models consisting of a single type of node representing partial descriptions of patients' states and a single type of causal link.

The CASNET program (Weiss, 1978) diagnoses various forms of the eye disease glaucoma. Causal relationships in CASNET are represented as links weighted with confidence factors, scaled from 1 (rarely causes) to 5 (almost always causes). A disease process is modeled as a causal chain of anomalous "pathophysiological" states, whose partial ordering depicts the progression of the disease over time. An example causal chain is that angle closure (if prolonged and untreated) Causes elevated intraocular pressure, which Causes optical-disk cupping, which Causes glaucomatous visual field loss. CASNET incorporates two other levels of description, one for disease categories such as open-angle glaucoma, and one for clinical observations such as patient symptoms and test results. Support for diagnostic hypotheses is propagated by associational links from clinical evidence to pathophysiological states, across causal links among pathophysiological states, and by classification links from anomalous states to disease categories. Diagnostic inferencing in CASNET treats the weighted causal links simply as conditional probabilities. However, the links reflect actual connections in the glaucoma domain and therefore can be activated in orderly chains with causally meaningful relations to disease hypotheses. Thus, the causal interpretation of links is relevant for constructing and validating the medical knowledge base, but not for the internal operation of the program per se.

The ABEL program (Patil, 1981) uses weighted causal links between anomalous states to reason about acid–base and electrolyte disturbances in patients. ABEL's links are more complex than CASNET's, consisting of multivariate relations between multiple cause and effect states that can also reflect diagnostic context and default assumptions. ABEL's causal links, while homogeneous, can also be decomposed hierarchically. For example, "{Coleostomy, Diarrhea, Fistula} Cause Lower-GI-Fluid-Loss, which Consists-of {Water-loss, Sodium-Loss, . . .} and which Causes {Dehydration, Acute-Renal-Failure, Hypotension, . . .}." Finally, ABEL interprets link weights as measures of magnitudes rather than as probabilities. This allows ABEL to reason about interactions between multiple diseases by deciding whether a known cause is sufficient to account for the observed magnitude of the disorder. If not, a second disturbance may be interacting with the first to cause the observed problem. ABEL relies on numeric calculations to generate disease-interaction hypotheses and to test whether a given cause is sufficient to account for an observed effect, which restricts the applicability of this technique to domains where adequate numerical theories exist. CADUCEUS (qv) (Pople, 1982) is the successor to INTERNIST-1, a general diagnostic program for internal medicine. CADUCEUS develops a pattern of interactions between causal and taxonomic links among disease hypotheses and pathophysiological states, focusing on the search for appropriate intermediate states. The causal network portion of the model features a single kind of link, meaning "may be caused by," to connect nodes that represent patient states. During diagnosis, the causal network is used to focus search on a small set of possible causes for an observed finding or inferred intermediate states. Simultaneously, the independent taxonomic network is navigated to isolate a parallel set of differential diagnosis problems. CADUCEUS then combines the two analyses to determine the diagnostic test that best refines the set of candidate diagnostic hypotheses at lowest cost.

Causal-link models generally depend on networks made up of behavioral fragments, state/event/action nodes, connected by causal relation arcs. Such models have been criticized on the grounds that they cannot represent the structure of mechanisms, nor derive behavioral predictions from structure. Without this capability, interactions between processes are difficult to reason about since they interact by having simultaneous influences on structural parameters. As noted above, ABEL (Patil, 1981) addresses this issue by assigning numeric strengths to causal links and performing substantial calculations with those numbers. In contrast, Pipitone (1984) *combines* causal and structural knowledge to troubleshoot electronic circuits: causal knowledge is incorporated as rules that depict the propagation of abnormal behavior through structural models. Most rules take the form "Given Test-Precondition-X, Parameter-1 Abnormality-1 at Terminal-1 always/sometimes causes Parameter-2 Abnormality-2 at Terminal-2." Abnormalities can be quantitative or qualitative, such as "DC-voltage-HIGH." The structural model specifies connections between terminals of relevant device modules. The rules and structural model are applied to find combinations of component faults consistent with test results reported by a technician. Given test costs and a priori probabilities of component failures, Pipitone's system computes the probability of a given fault combination and the most cost-effective test to perform next to further isolate device faults. Causal rules can also represent explicit fault models, such as "Faulty module-X sometimes causes Abnormality-Y," considerably enhancing diagnostic efficiency.

IMPLICIT CAUSAL MODELS

Symbolic simulation is a type of causal reasoning in which the behaviors of individual components of complex physical or biological systems are described; the global behavior of the system is then deduced from the interactions of the components. Fundamentally different approaches for describing component behavior have been developed for discrete and continuous process models of systems.

Symbolic simulation of a continuous system derives a qualitative description of that system's possible behaviors from a qualitative description of its structure. The structural description, which represents an abstraction of the exact differential equations that model the system, is based on a set of continuous state variables and constraints that must be satisfied by the values of those variables at each instant in time. Consider, for example, a system consisting of two containers filled with liquid connection by a uniform pipe with a valve. State variables would include flow, rate of flow, the relative heights and fluid levels of the containers, and valve position. Values can be landmark points in a continuous space (eg, 0, Min-X, ∞), intervals between landmark points, (eg, partially full), or points from a discrete space (eg, open, closed). System constraints, following simple fluid mechanics, would be that the total volume of fluid remain constant, that the rate of flow into one container is equal to the rate of flow out of the other, and so on. Biophysical systems can be characterized similarly, by combining qualitative physics and chemistry.

A single behavior of a mechanism is represented as a sequence of qualitative values of the state variables, together with directions of value changes. For example, both flow and rate of flow can increase, decrease, or remain constant. Qualitative simulation propagates changes of state through the structural description (eg, opening the pipe valve), by cycling through the following steps: (1) propagating a partial set of variable values across constraints to construct a complete system description at given instants; (2) examining variables with changing values in a given state to determine whether they are approaching limiting landmark values; and (3) determining the next qualitatively distinct state by analyzing the possible transitions of individual variables to determine which transitions can occur next. Different researchers use transition-ordering decision procedures (Williams, 1984a, 1984b; Forbus, 1984; Kuipers, 1984) or constraint-based transition filtering rules (Kuipers, 1985; de Kleer, 1984; de Kleer and Brown, 1984; de Kleer, 1979) to accomplish this step. Structural descriptions may not provide sufficient information to specify the next qualitative state uniquely, so qualitative simulators create a branching tree or directed graph of state descriptions that represent possible behaviors (and alternative solutions to the abstracted differential equations). Variations in qualitative structural models and simulation algorithms are described more fully in Bobrow (1985) (see PHYSICS, QUALITATIVE).

Reasoning in qualitative simulation is based on structural models devoid of explicit causal structures. Causal knowledge is nonetheless both implicitly present and important. For example, process-based structural descriptions (Forbus, 1984) depend on the concept of spatially and temporally localized histories (Hayes, 1979) to infer restrictions on the set of possible causal interactions. More significantly, simulation algorithms for propagating disturbances through a network of constraint equations presuppose notions of "causal flow." Intuitively, causal flow depicts the movement of "information" through a system, such as fluid, current, or applied force. Flow serves to identify which state variables are dependent and independent with respect to one another; these relationships, in turn, determine the order for solving the simultaneous constraint equations. Iwasaki and Simon (1986) offer an insightful critique of the "mythical causality" flow model underlying de Kleer and Brown's (1984) qualitative simulator as compared with Simon's (1977) earlier theory of causal ordering for solving exact differential equations. By virtue of this latent causal content and the ordering of model events generated by a simulation, qualitative behavioral descriptions can be used to answer causal questions about the systems they depict. Thus, while qualitative simulation fails to model or explicate causation per se, it clearly provides an important vehicle for understanding and predicting the behavior of complex physical and biological systems.

Symbolic simulation of continuous systems maps continuous value spaces into discrete spaces of qualitative descriptions. In digital electronic systems, on the other hand, state variables already have discrete values. The set of permissible state changes is constrained not by continuity but by the Boolean semantics of digital circuit components (eg, AND and OR logic gates). Digital simulators thus rely on catalogs of device descriptions rather than a fixed set of qualitative state-transition rules; both structure and behavior of circuit modules are modeled as equations relating input, output, and state variables. Symbolic descriptions of variable values are still needed to handle abstractions of values and disjunctions of possible values; such symbolic data must be propagated forward through the simulation model when sufficient information is not available to specify exact values uniquely.

Discrete symbolic simulation has been extensively investigated for automatically: verifying designs for digital circuits; generating tests for manufactured circuits (Shirley, 1986) and diagnosing malfunctioning circuits. For example, VERIFY (Barrow, 1984) simulates the behavior of a complex digital circuit and compares this prediction to the device specification. Hierarchical decomposition of system descriptions into modules allows different levels to operate in different value domains: voltage levels, logic values, integers. Diagnosis is required when the behavior of an actual circuit fails to satisfy a correctly specified design. A hypothesized fault in the circuit can be tested by simulating the circuit forward from the suspect component(s) with the given set of test inputs to see whether the prediction matches the observed anomalous outputs. The space of possible fault hypotheses in nontrivial digital circuits is generally very large, so that intelligent search (qv) is crucial. Davis's (1983) troubleshooting system controls search through a set of assumptions about the nature of the fault, which are progressively suspended as search is completed in the most likely portions of fault space. Research continues very actively on alternative, more efficient search algorithms for fault diagnosis. Most simulators for circuit diagnosis only reason about faults involving topological adjacency: logical connectivity between gates or modules spelled out in the design schematic. Davis (1984) investigated faults related to physical adjacency (eg, bridging short circuits caused by wire or solder fragments connecting adjoining traces, faults induced by electromagnetic field or thermal proximity). Discrete simulators that reason about the spatial arrangement of the actual fabricated circuit elements can uncover fault candidates through potential causal pathways of interaction that are not revealed in logical schematics.

Discrete simulation is a very useful application of behavioral modeling in the specialized domain of digital circits. As with continuous qualitative reasoning, causal information such as "Fault-X explains Behavior-Y" can be derived from simulations that are not explicitly causal in character. However, the extension or applicability of discrete simulation research results to the general study of causal reasoning appears to be limited.

The third major class of implicit causal models consists of logic-based systems for planning (qv) and other forms of commonsense reasoning (qv) (Davis, 1990). Causal structure in these models is captured implicitly in so-called "causal" or projection axioms (Dean, 1987). Such axioms specify the behavioral consequences of applying a domain-specific operator or being in a particular state in a dynamic physical system. An example blocks-world axiom would be "IF (On ?x ?z ?t1) & (Kind-of ?x block) & (Kind-of ?z block) & (Clear ?x ?t1) & Move (?x table ?t1) & (= ?t2 (+ 1 ?t1)) THEN (On ?x table ?t2) & (Clear ?x ?t2) & (Clear ?z ?t2)." Intelligent "look-ahead" planners use such axioms to predict the utility of possible alternative actions in achieving goals. As in other implicit models, neither the causal nature nor content of projection axioms play any explicit, distinguished role in problem-solving reasoning.

THEORIES OF CAUSATION

Explicit causal models take as primitive the notion that one event can cause another, and derive implications from this relationship without defining the meaning of causation; causal links simply stand in for deeper, "first principles" theories or "compiled" bodies of statistical (eg, diagnostic) knowledge. Implicit causal models supply deep behavioral theories (eg, qualitative physics, Boolean logic of digital circuits), but their reasoning lacks any essential reference to causation, either scientific or commonsensical. In contrast, theories of causation focus directly on the meaning of causation and the criteria for legitimately asserting causal relationships given observed regularities in the world. This section discusses research on causal theories in AI, statistics, cognitive science, and the philosophy of science.

Much of what we take to be "causal" knowledge of the commonsensical physical world is not easily captured in simple causal links or other causal models. For example, we know that water poured from a canteen might splash off a rock, wetting both the rock and one's legs, and then soak into the ground. Hayes (1979) notes that a critical prerequisite for automating reasoning about such phenomena is a representational model that is sufficiently expressive to state the simple facts and relationships that constitute commonsense knowledge. He also argues that causation, as such, is not a self-contained category that can support independent axiomatization; rather, it is a type of knowledge that must be represented separately for different physical domains: causal knowledge about liquids is knowledge about liquids, not about causality. Subsequent research by Hayes and others to codify causal knowledge in this piecemeal manner is collected in (Hobbs and Moore, 1985).

CYC, an intelligent encyclopedia project (Lenat and coworkers, 1990), is an ambitious attempt to construct a massive knowledge base that captures and reasons about the body of commonsense facts and concepts about the world that we know and assume others to know. Like most AI researchers, Lenat subscribes to a linguistic rather than an ontological view of causation: causal relations hold between propositions describing events rather than between events per se. His theory of causation can be summarized via the following set of axioms: (1) causation entails (and is stronger than) material implication [ie, if P Causes Q then (P ⊃ Q)]; (2) causing events occur before the start of caused events; (3) propositions describing events are true either because they were asserted or by virtue of a deduction involving a causal rule (eg, $P(e_1)$ Causes $Q(e_2)$ and $P(e_1)$ is true; (4) Causes is a general class relation that has numerous specializations; and (5) in statements of the form "$P(e_1)$ Causes $Q(e_2)$," Causes is either a primitive relation or decomposable such that "$P(e_1)$ Causes_3 $P_3(e_3)$ . . . $P_n(e_n)$ Causes_n $Q(e_2)$." Axioms 4 and 5 are of particular interest. CYC supports a diversity of specialized causal relations, such as electricalConductingCauses, a specialization of physicallyCauses that holds among ElectricalEvents. Lenat also holds that complex causal relations are decomposable into temporally extended sequences chains of different, more specific types of causal links. [In contrast, ABEL (Patil, 1981) allows only for hierarchies of a single type of causal link.] Together, these features allow CYC to (1) systematically address problems concerning alternative causal explanations and (2) provide different levels of explanation in terms of causal relations of varying levels of complexity or granularity. Lenat holds that causal relations are nonprimitive in the sense that causal statements constitute a distinguished subset of implications [ie, P Causes Q is defined as (P ⊃ Q) & Causal '(P ⊃ Q)]; however, he has not yet proposed a precise definition for the metalogical predicate "Causal."

The RX program (Blum, 1982) is an "automated statistician" that examines medical evidence stored in a time-oriented database and determines whether causal relationships can be inferred. RX follows accepted procedures in statistical data analysis and uses an operational definition of causality: A is said to cause B if, over repeated observations (1) A is generally followed by B, (2) the intensity of A is correlated with the intensity of B, and (3) no known third variable C is responsible for the correlation. RX's causal links incorporate multiple attributes, including numeric ones such as intensity and frequency. The central concern of RX lies with the third test of causality, for nonspuriousness, and is responsible for the bulk of its domain-specific knowledge of causal relations.

Pearl (1986) defines a type of belief network based on a probabilistic causal theory (see BAYESIAN INFERENCE METHODS). Causal networks are directed, acyclic graphs of nodes representing propositions about multivalued observations or mutually exclusive hypotheses (eg, Test-result is low medium, high, Patient-diagnosis is disease A, B, or C). Causal relations are represented by directed links connecting pairs of nodes X and Y; each link is quantified by a matrix (M(Y|X). A matrix element M(Y|X)i,j is the conditional probability of Yi given Xj, whose value is a constant between 0 and 1. Matrix values are static and fixed. The influence of new evidence on belief nodes in the causal network is propagated through links using an updating algorithm based on Bayes's theorem for manipulating conditional probabilities. Pearl demonstrates evidence fusion and belief propagation for binary causal trees, in which exactly one variable can be the cause of another. For example, the patient having disease Xj causes "Test-Result-k" to be Zj. Extension to multiply connected networks is based on the use of auxiliary dummy variables, called "hidden causes," to flatten out the network.

Pearl argues that his causal belief networks reflect the cognitive structure of human causal model formation. For example, we posit unifying, centralized causal variables, such as standard time, to account for observed correlations in the world, such as the agreement of personal watches and public clocks. The common "causal" mechanism also enables us to treat individual observations as being independent of one another. Pearl's Bayesian model is cognitively attractive because of its modularity and computational efficiency, but unintuitive in its purely quantitative measure of (the strength of) causal links. Suppes (1970) also defines causation via conditional probabilities but specifies conditions to qualify causal relations as genuine, spurious, direct, indirect, supplementary, sufficient, or negative.

Most of the recent theories of causation by AI researchers extend formal logics developed for temporal reasoning. This trend reflects the intimate relationship between theories of time, which are necessary to describe change, and theories of causation, which constrain possible changes to conform to some restricted set of regular patterns. The primary goal of causal logics in AI is to address the classic frame problem (see Frames), which Shoham (1988b) reformulates into two issues: the *qualification* problem—how to specify projection rules that characterize the "physics" of change in a domain without endless assumptions that qualify relevant background conditions; and the *extended prediction* problem—how to make predictions over significant intervals of time without having to worry about possible intervening events (eg, a gun becoming unloaded before it is fired). To solve these problems, formal logics attempt to establish (1) the unique syntactic forms that causal-temporal assertions take in a body of knowledge, (2) a precise semantics for interpreting such statements, and (3) algorithms for propagating causal knowledge forward in time. These logics are called nonmonotonic (see Reasoning, nonmonotonic), because statements true at one time can subsequently become false as the world changes.

Hayes's (1979) concept of histories is an important precursor to causal logics in addressing the prediction problem. Objects and events are taken to have four-dimensional spatiotemporal extents, called histories. We generally have definite intuitions about such boundaries in both space and time. Two objects can interact only if their histories intersect. For example, moving a block on a tabletop cannot affect a block that is now in the other room because the histories of the two blocks do not intersect and because the history of a move action does not have a sufficient radius to affect both blocks. On the other hand, the move action could potentially affect other blocks on the same table, so a more careful check for collisions is warranted. Thus, histories constrain the types of causal interactions that need to be considered in a given situation.

Causal-temporal logics in AI establish a logical representation of events, continuous change, and primitive causal relations. Various axioms ground inferences about temporal and causal assertions. Allen (1984) takes for his primitives properties, events, processes, and temporal intervals. ECauses is a primitive causal relation on events and intervals that is transitive, antisymmetric, and antireflexive. Two of Allen's deductive axioms are (1) if ECause(e,t,e′,t′) and e occurs, then e′ occurs; and (2) if ECause(e,t,e′,t′) then interval t′ precedes, meets, overlaps, or lies within interval t. ACause, Allen's other primitive relation, holds between human agents and their actions. McDermott (1982) posits primitive facts, events, and instants; intervals are defined as sets of instants. His axioms are similar to Allen's. McDermott models time as a branching set of chronicles, with a single past and multiple possible futures. A chronicle is a global history of the entire universe, in contrast to Hayes's bounded histories of individual objects. McDermott posits two primitive causal predicates, ECause, where one event causes another, possibly with an intervening interval of delay (eg, setting a timer in the morning to switch on lights at night), and PCause, where an event causes a persisting fact. Persistent facts address the problem of extended prediction; once known, facts can be assumed to remain true over some specified, extended interval in the absence of information to the contrary. McDermott also analyzes the causal notion of a current action preventing a future event from taking place, in terms of alternative future chronicles.

Shoham's (1988a, 1990) causal-temporal logic is the most comprehensive AI theory of causation to date, complete with a rigorous semantic interpretation. Unlike other causal theorists, Shoham adopts an epistemic modal logic (see Logic, modal). Base axioms take the form □(t1,t2,P), where the modal operator "□," read "It is necessary that," applied to the temporal proposition is interpreted as "an agent *believes that* P is true during interval [t1,t2]." Shoham defines "A Causes B" as "If whenever one believes A(t_i), and one does not believe that some set of background conditions C(tj) are false, then one believes B(t_k)" for i,j < k. C represents a set of plausible default assumptions, which, if known to be violated would force the "effect" belief to be retracted. For example, □(t5, t5, turn-key) Causes □(t6, t6, car-starts) given that □(t5, t5, battery-OK & plugs-clean, . . .}). A *theory* is a collection of causal statements (as defined above) and base axioms. *Chronological ignorance* is a criterion for selecting theories in which as little as possible that runs contrary to the defaults is known for as long as possible. For example, prefer the theory in which □(t6, t6, car-starts) over one in which □(t5, t5, battery-dead). Shoham proves that there is a unique and effectively computable causal theory that is maximally chronologically ignorant, thus addressing the qualification problem. Similarly, Shoham deals with extended prediction by selecting a (unique, computable) subset of causal theories, called inertial theories. Potential histories, a variant of McDermott's persistent facts, form the basis of the inertial preference criterion. Intuitively, a potential history picks out a "natural" course of events that persists unless there is explicit information to the contrary about intervening causal influences. Shoham also offers formal definitions of causal notions such as prevention and enablement.

An important feature of Shoham's account is that causal statements have a unique logical form. Moreover, causal and lawlike statements are distinct, both syntactically and in meaning: causal statements are contingently true (ie, "defeasible"), whereas laws are universally and necessarily true. An important disadvantage is that causal reasoning in Shoham's logic is exclusively predictive; no uniqueness theorems exist in his logic for projecting theories backward in time. This precludes abduction, which explains why things are the way they are by reasoning from effects to their likely causes. [Note: Pearl (1988) deals with this problem of "retrodictive" explanation by adding a separate class of probabilistic links to his belief networks. These links reflect accrued evidential support (eg, if smoke, then fire), where causal links, directed oppositely, reflect accrued causal support (eg, if fire, then smoke).] Finally, Shoham argues that his logic represents an attractive cognitive model with its low memory overhead, computational efficiency, and modularity; background conditions are isolated from causal principles and maintained as default or statistically most likely assumptions relative to a given environment.

Lifshitz (1987) proposed a theory of action based on a nontemporal causal logic to address the problems of qualification and prediction. He takes causation as a primitive relational predicate on actions, states and fluents, which are functions on situations (eg, The current U.S. President). Explicit causal axioms of the form Causes(action old-state new-state) depict the effects of successfully performed actions, such as Cause(toggle-switch on off) and Cause(shoot-gun loaded false). The preconditions for actions to be successful are declared in separate axioms, such as Precond(loaded-gun, shoot-gun). Lifschitz demonstrates that changes occur in his models only if they conform to the causal axioms, without explicit reference to time.

AI research on causal reasoning benefits greatly by considering the constraints on human causal knowledge discovered by cognitive scientists. Cognitive theories attempt to accommodate the kinds of variation observed in the human population (individual, developmental, and historical) within general models of causal relations and problem-solving performance. Such models provide valuable resources for AI-based causal knowledge representations and automated reasoning methods in both commonsensical and technical domains. As in many areas of developmental psychology, Piaget (1966) provides the earliest systematic studies of causal knowledge, looking at the types of early theories children create to explain phenomena such as the wind blowing, clouds moving, boats floating, shadows, bicycles, and airplanes. He identifies three distinct classes of causal explanations—psychological–magical, animistic, and rational–mechanical; only the third encompasses what we normally consider as "causation." Piaget argues that evolution through these classes is driven by three processes of cognitive development: the progression from a subjective to an objective view of causality as external to the self, from the view of causality as an almost immediate cooccurrence of events to an ordered sequence of intermediate steps, and from causal relations seen as irreversible changes to reversible mechanical connections.

Empirical cognitive studies collecting and analyzing verbal protocols have been used to construct explicit causal computational models of the behavior of heat engines (Williams and co-workers, 1983) and to build a qualitative simulation of the physiological mechanisms underlying a kidney disease (Kuipers and Kaissirer, 1984). Other researchers have investigated the influence of causal knowledge on other kinds of reasoning. For example, Tversky and Kahneman (1980) demonstrated that schemas of causal relations strongly dominate the process of estimating relative and absolute probabilities in situations where exact knowledge is unavailable. Cognitive research has been particularly active in education, studying the differences between naive causal models of mechanical systems used by novice students and lay adults as contrasted with those of expert physicists. Related studies have explored analogies between the historical evolution of physical theories (eg, from pre-Galilean to Newton dynamics) and the progression of causal models adopted by science students. Gentner and Stevens (1983) have collected important research papers in this area.

A third important class of cognitively oriented studies of causation consists of recent computational models for acquiring or learning causal knowledge. Anderson (1989) incorporates "inate knowledge" of causal inferences into his cognitive model PUPS to demonstrate rationalistic learning, which he defines as the extraction of problem-solving operators from experience. His store of causal induction principles include heuristics involving identity, previous action, and minimal contrast. For example, on typing (lis) into a computer that responds (lis: unknown function object), the recurring token "lis" can be inferred to have a causal role in the occurrence of the second event. The second heuristic stipulates that if an event has no discernible cause, ascribe as its cause an immediately preceding action. Finally, if pairs of antecedent and consequent events are identical except in one point, infer that the difference between the first pair of events causes the difference in the second pair.

The machine learning program OCCAM (Pazzani, 1987) acquires simple causal models for predicting the outcomes of everyday phenomena such as dropping cups or opening heavy doors. Models are comprised of roles (actors, objects), actions (move, decide), and descriptive attributes (material, age). Causal elements in these models consist of domain-specific "dispositions" (eg, fragility, strength), of objects or agents that effect actions. For example, by virtue of their strength, adults applying force to heavy objects such as doors induce state changes, namely, door motions. OCCAM forms a "current best causal hypothesis" by differential analysis of current observations and events recalled from memory. Preference is given to hypotheses that involve minimal sets of previously useful distinctions. For example, the program prefers age over hair color in selecting between "When a person (who is an adult/who has brown hair) pulls on a door, it opens" because of its previous utility in successfully predicting outcomes of household activities. Generalization rules ground example-based learning of new causal regularities or dispositions. An example rule schema is that differences in actors performing similar actions cause differences in results. OCCAM's rules are structured to reflect the causal factors of (1) covariation (effects always accompany causes), (2) temporal order (causes precede effects), and (3) mechanism (physical mediators link cause and effect).

Causation has been analyzed most extensively by philosophers of science. A comprehensive review of this literature is well beyond the scope of this entry; Mackie (1974) provides an excellent general introduction, while Sosa (1975) collects important contemporary papers (eg, Lewis, 1973). Most accounts derive from the work of Hume, the eighteenth-century empiricist philosopher. Modern Humean "regularity" accounts hold events to be causally connected just in case an ordered sequence of those events instantiates an observed regular succession of events belonging to appropriate event categories or classes. For example, let C and E stand for the propositions that events c and e exist or occur. Let L stand for some nonempty set of true lawlike propositions (eg, all metallic objects expand when heated), F denote some set of true propositions of particular fact, and ⊃ stand for the relation of material implication. Then c causes e if and only if (1) C and E are true; and (2) L and F jointly imply C ⊃ E, although L and F jointly do not imply E, and F alone does not imply C ⊃ E. Here, causation does not connect events per se; rather, it is defined in terms of logical relations among *propositions* that describe the relevant events (C,E), environmental context or background conditions (F), and lawlike regularities (L). In accordance with intuition, causal relations are asymmetric and irreversible: expanding metallic objects do not cause them to be heated; nor is it the case that not heating metallic objects cause them not to expand (ie, the contrapositive).

Regularity accounts engender several difficulties. First, events characterized in some ways may give rise to causal relations as per definition, while events that we intuitively take to be identical, specified by different descriptions, may turn out to be causally *un*related on the very same analysis. For example, consider the events c (Y's firing of a bullet at time t), e (X's death from a bullet wound in the heart suffered shortly after t, and e′ (X's death from a bullet wound in the heart suffered shortly after t, where that bullet was fired by Y at time t). c and e are causally related by the definition presented above, but not c and e′, since F alone implies C ⊃ E′ when C ⊃ E′ is a tautology.

Second, there are misfits between the notions of lawful and causal dependence. Regularities having no apparent relevance to our notions of direct causal dependence can be cast into (universal or statistical) lawlike form, ostensibly suitable for supporting causal relations as per definition. Consider, for example, the semantic regularity that a woman inevitably becomes a widow on the death of her husband. Another (noncausal) psychological regularity is that conscientious programmers who discover bugs in their code invariably try to correct their errors. A third illustration (McDermott, 1982) is the physical regularity that the arrival of one's shadow always precedes one's own arrival.

Finally, uniform definitions of causal relations invariably tend to be both stronger and weaker than our intuitions in unusual circumstances. Here, stronger means that causal relations obtain by definition but counter to intuition; weaker corresponds to the reverse situation. These tensions arise because our commonsense causal intuitions are simply not uniformly consistent across age, education, or history. Our individual intuitions are particularly divergent when faced with novel or complex situations, such as physical systems containing feedback loops (Iwasaki and Simon, 1986).

Any theory of causation, whether proposed by philosophers, cognitive scientists, or AI researchers, must provide answers to these three problems, in the form of (1) a comprehensive ontology of events (or states) that specifies criteria for event identity and individuation; (2) a general account of "significant" lawhood, which must be free of presuppositions concerning causal relations to avoid circularity; and (3) a precise specification of whose commonsense or scientific causal intuitions are being analyzed. These issues are as profound and difficult as the concept of causation itself, which means that a fully adequate theory of causation is not imminent. Nevertheless, research on these fundamental questions is indispensible for continued advances in causal reasoning, as the concluding section will suggest.

DISCUSSION

AI research on causal models has achieved important successes in representing and reasoning about behavior in complex systems and commonsensical causal relationships. However, these models display some serious shortcomings, both individually and collectively. First, causal models employ diverse, typically incompatible patterns of causal inference and representations for causal relations (eg, network links, rule connectives, projection axioms). The clear utility of these models strongly suggests that they all capture important aspects of causal knowledge. Deep theories of causation are needed to provide (1) an analytic framework for unifying heterogeneous causal models across scientific domains and commonsense knowledge and (2) rigorous foundations for causal reasoning. Lenat's taxonomy of causal relations and Shoham's causal logic represent important steps in this direction.

Second, in general, current causal models cannot be constructed mechanically. Causal structures in explicit causal models are primitive and unanalyzable; thus, designers and domain experts must build, verify, and extend causal knowledge bases manually. Excepting component-based simulators, similar limitations hold for AI systems that represent causal structures or inference implicitly. Automated tools for developing causal models will depend on theories of causation that specify explicit definitional grounds for causal relations: knowledge acquisition and maintenance tools can then be constructed that exploit these truth conditions to mechanically confirm or deny the assertion of causal relations in particular domains and problem contexts.

Third, current causal models cannot discover new kinds of causal structures, nor recognize and react to causally interesting aspects of observed situations. Such adaptive capacities presuppose (1) generalized principles of causal induction and (2) a capacity for meta-level reasoning, such as reflection about ongoing inferences. The causal theories embodied in PUPS and OCCAM represent a strong start on research into the first topic. Important early investigations in the second area include Davis's research on relaxing causal assumptions in digital circuit diagnosis and Weld's (1986) algorithm for causal "aggregation," which recognizes recurring (discrete) cycles in qualitative simulations and abstracts them into a continuous repetitive process. In sum, significant research challenges remain in understanding the nature of causation itself. The theories of causation that result from this research will provide solid foundations for a new generation of causal models that truly automate causal reasoning in AI.

BIBLIOGRAPHY

J. F. Allen, "Towards a General Theory of Action and Time," *Artif. Intell.* **23,** 123–154 (1984).

J. R. Anderson, "Theory of the Origins of Human Knowledge," *Artif. Intell.* **40,** 347–410 (1989).

H. G. Barrow, "VERIFY: A Program for Proving Correctness of Digital Hardware Designs," *Artif. Intell.* **24,** 437–491 (1984).

R. L. Blum, "Discovery and Representation of Causal Relationships from a Large Time-Oriented Clinical Database: The RX Project," *Lecture Notes in Medical Informatics,* Vol. 19, Springer, New York, 1982.

D. G. Bobrow, ed., *Qualitative Reasoning about Physical Systems,* MIT Press, Cambridge, Mass., 1985, reprinted in *Artif. Intell.* **24,** 1984 (which collects important papers on implicit causal models; qualitative and discrete simulation).

E. Davis, *Representations of Commonsense Knowledge,* Morgan-Kaufmann, Palo Alto, Calif., 1990.

R. Davis, "Diagnosis via Causal Reasoning: Paths of Interaction and the Locality Principle," *Proceedings of the Third National Conference on Artificial Intelligence,* Washington, D.C., AAAI, Menlo Park, Calif., 1983.

R. Davis, "Diagnostic Reasoning Based on Structure and Behavior," *Artif. Intell.* **24,** 347–410 (1984).

T. L. Dean and M. Boddy, "Incremental Causal Reasoning," *Proceedings of the 6th National Conference on Artificial Intelligence,* Seattle, Wash., AAAI, Menlo Park, Calif., 1987.

M. G. Dyer, *In-Depth Understanding: A Computer Model of Intelligent Processing for Narrative Comprehension,* MIT Press, Cambridge, Mass., 1983.

K. D. Forbus, "Qualitative Process Theory," *Artif. Intell.* **24,** 85–168 (1984).

D. Genter and A. Stevens, eds., *Mental Models,* Erlbaum, Hillsdale, N.J., 1983 (an excellent overview collection of papers on cognitive science research on causal reasoning).

P. J. Hayes, "The Naive Physics Manifesto," in D. Michie, ed., *Expert Systems in the Micro-Electronic Age,* Edinburgh University Press, Edinburgh, 1979.

J. R. Hobbs and R. C. Moore, eds., *Formal Theories of the Commonsense World,* Ablex Publishing, Norwood, N.J., 1985.

Y. Iwasaki and H. A. Simon, "Causality in Device Behavior," *Artif. Intell.* **29,** 3–32 (1986).

J. de Kleer, "The Origin and Resolution of Ambiguities in Causal Arguments," *Proceedings of the Sixth IJCAI,* Tokyo, Morgan-Kaufmann, San Mateo, Calif., 1979.

J. de Kleer and J. S. Brown, "A Qualitative Physics Based on Confluences," *Artif. Intell.* **24,** 7–83 (1984).

B. J. Kuipers, "Commonsense Reasoning about Causality: Deriving Behavior from Structure," *Artif. Intell.* **24,** 169–204 (1984).

B. J. Kuipers, "The Limits of Qualitative Simulation," *Proceedings of the Ninth IJCAI,* Los Angeles, Calif., Morgan-Kaufmann, San Mateo, Calif., 1985.

B. J. Kuipers and J. P. Kassirer, "Causal Reasoning in Medicine: Analysis of a Protocol," *Cog. Sci.* **8,** 363–385 (1984).

D. B. Lenat, R. V. Guha, K. Pittman, D. Pratt, and M. Shepherd, "CYC: Toward Programs with Common Sense," *Commun. ACM* **33,** 30–49 (1990).

D. Lewis, "Causation," *J. Philos.* **70,** 556–567 (1973); reprinted in E. Sosa, ed., *Causation and Conditions,* Oxford University Press, London, 1975.

V. Lifshitz, "Formal Theories of Action (Preliminary Report)," *Proceedings of the Tenth IJCAI,* Milan, Italy, Morgan-Kaufmann, San Mateo, Calif., 1987.

J. L. Mackie, *The Cement of the Universe: A Study of Causation,* Oxford University Press, London, 1974 [reviews both historical (Hume, Kant) and more contemporary philosophical theories of causation].

D. McDermott, "A Temporal Logic for Reasoning about Processes and Plans," *Cog. Sci.* **6,** (1982).

R. S. Patil, P. Szolovits, and W. B. Schwartz, "Causal Understanding of Patient Illness in Medical Diagnosis," *Proceedings of the Seventh IJCAI,* Vancouver, B.C., Morgan-Kaufmann, San Mateo, Calif., 893–899, 1981.

M. Pazzani, M. Dyer, and M. Flowers, "Using Prior Learning to Facilitate the Learning of New Causal Theories," *Proceedings of the Tenth IJCAI,* Milan, Italy, Morgan-Kaufmann, San Mateo, Calif., 1987.

J. Pearl, "Fusion, Propagation, and Structuring in Belief Networks," *Artif. Intell.* **29,** 241–288 (1986).

J. Pearl, "Embracing Causality in Default Reasoning," *Artif. Intell.* **35,** 259–271 (1988).

J. Piaget, *The Child's Conception of Physical Causality,* Routledge and Kegan Paul, London, 1930; reprinted by Littlefield, Adams & Co., Totowa, N.J., 1966.

F. Pipitone, "An Expert System for Electronics Troubleshooting Based on Function and Connectivity," *Proceedings of the First IEEE Conference on AI Applications,* Boulder, Colo., IEEE Computer Society Press, Washington, D.C., 1984.

H. E. Pople, Jr., "Heuristic Methods for Imposing Structure on Ill Structured Problems: The Structuring of Medical Diagnostics," in P. Szolovits, ed., *Artificial Intelligence in Medicine,* AAAS/Westview, Boulder, Colo., 1982.

C. Rieger and M. Grinberg, "The Declarative Representation and Procedural Simulation of Causality in Physical Mechanisms," *Proceedings of the Fifth IJCAI,* Cambridge, Mass., Morgan-Kaufmann, San Mateo, Calif., 1977.

R. C. Schank and R. P. Abelson, *Scripts, Plans, Goals, and Understanding,* Erlbaum, Hillsdale, N.J., 1977.

M. H. Shirley, "Generating Tests by Exploiting Designed Behavior," *Proceedings of the Fifth National Conference on Artificial Ingelligence*, Philadelphia, AAAI, Menlo Park, Calif., 1986.

Y. Shoham, "Nonmonotonic Reasoning and Causation," *Cog. Sci.* **14**, 213–252 (1990).

Y. Shoham, "Chronological Ignorance: Experiments in Nonmonotonic Temporal Reasoning," *Artif. Intell.* **36**, 279–331 (1988a).

Y. Shoham, *Reasoning about Change: Time and Causation from the Standpoint of Artificial Intelligence*, MIT Press, Cambridge, Mass., 1988b.

H. A. Simon, *Models of Discovery*, D. Reidel, Boston, 1977.

E. Sosa, ed., *Causation and Conditions*, Oxford University Press, London, 1975 (a collection of articles on causation, including counterfactual theories).

P. Suppes, *A Probabilistic Theory of Causation*, North Holland, Amsterdam, 1970.

A. Tversky and D. Kahneman, "Causal Schema in Judgments under Uncertainty," in M. Fishbein, ed., *Progress in Social Psychology, 1*. Erlbaum, Hillsdale, N.J., 1980.

S. M. Weiss, C. A. Kulikowski, S. Amarel, and A. Safir, "A Model-Based Method for Computer-Aided Medical Decision-Making," *Artif. Intell.* **11**, 145–172 (1978).

D. S. Weld, "The Use of Aggregation in Causal Simulation," *Artif. Intell.* **30**, 1–34 (1986).

B. Williams, "The Use of Continuity in a Qualitative Physics," *Proceedings of the Fourth IJCAI*, Austin, Tex., Morgan-Kaufmann, San Mateo, Calif., 1984a.

B. Williams, "Qualitative Analysis of MOS Circuits," *Artif. Intell.* **24**, 281–346 (1984b).

M. D. Williams, J. D. Hollan, and A. L. Stevens, "Human Reasoning about a Simple Physical System," in D. Genter and A. Stevens, eds., *Mental Models*, Erlbaum, Hillsdale, N.J., 1983.

RICHARD M. ADLER
Symbiotics, Inc.

REASONING, COMMONSENSE

For an artificial system to act sensibly in the real world, it must know about that world and it must be able to use its knowledge effectively. The common knowledge about the world that is possessed by every schoolchild and the methods for making obvious inferences from this knowledge are called common sense in both humans and computers. Almost every type of intelligent task (natural language processing, planning, learning, high level vision, expert-level reasoning) requires some degree of commonsense reasoning to carry out. The encoding of commonsense knowledge has been recognized as one of the central issues of AI since the inception of the field (McCarthy, 1959).

Endowing a program with common sense, however, is a very difficult task. Common sense involves many subtle modes of reasoning and a vast body of knowledge with complex interactions. Consider the following quotation from *The Tale of Benjamin Bunny*, by Beatrix Potter:

> Peter did not eat anything; he said he should like to go home. Presently he dropped half the onions.

Except that Peter is a rabbit, there is nothing subtle or strange here, and the passage is easily understood by five-year-old children. Yet these three clauses involve, implicitly or explicitly, concepts of quantity, space, time, physics, goals, plans, and speech acts. An intelligent system cannot, therefore, understand this passage unless it possesses a theory of each of these domains and the ability to connect this theory in a useful way to the story.

Many of the central issues in the automation of commonsense reasoning appear in all types of AI reasoning, particularly the development of domain-independent knowledge structures and inference techniques, and the analysis and implementation of plausible reasoning. Because these issues are common throughout AI, they will not be studied in this article. (See KNOWLEDGE REPRESENTATION; REASONING, DEFAULT; REASONING, NONMONOTONIC; REASONING, PLAUSIBLE). Here, the focus will be on issues that arise in the study of specific commonsense domains.

GENERAL ISSUES AND METHODOLOGY

The analysis of reasoning in a commonsense domain has three major parts:

1. *Representation*. The development of knowledge structures that can express facts in the domain.
2. *Domain Theory*. The characterization of the fundamental properties of the domain and the rules that govern it.
3. *Inference Techniques*. The construction of algorithms or heuristics that can be used to automate useful types of reasoning.

A popular methodology for carrying out these kinds of analysis runs along the following lines (McCarthy, 1968; McCarthy and Hayes, 1969; Hayes, 1977, 1978; McDermott, 1978; Davis, 1990; Minsky, 1975; McDermott, 1987). The researcher begins by defining a microworld, a small, coherent domain of study. Aspects of the real world that lie outside the microworld will either be ignored in the work, or will be represented in some very coarse, *ad hoc* manner. Next, a coherent collection of commonsensically obvious inferences in the microworld are assembled. The researcher determines what problem-specific information and general domain knowledge (qv) is needed, explicitly, or implicitly, to justify these inferences. A language is developed in which these facts can be expressed and these inferences can be validated; typically, this language is written in some known logic (qv). Having categorized the types of information and rules that are needed, the researcher can work on developing data structures and procedures that allow the efficient solution to some significant classes of problems.

A knowledge representation for a commonsense domain must satisfy three requirements. First, the representation must be able to describe the relevant aspects of the domain involved. Second, it must be possible to use the representation to express the kinds of partial knowledge typically available; the design of the language must take account of likely kinds of ignorance. Third, it must be possible to implement useful inferences as computations using the representation. These requirements are called ontological adequacy, expressivity or epistemic adequacy, and effectiveness or heuristic adequacy (McCarthy and Hayes, 1969). Each of these requirements is defined relative to a certain set of problems; a representation may be adequate for one kind of problem but not for another.

The greatest difference between representations for commonsense reasoning and representations used in other areas of computer science lies in the expressivity requirement. Most computer science representations assume either complete information, or information that is partial only along some limited dimensions. By contrast, commonsense reasoning requires dealing with a wide range of possible types of partial information, and degrading gracefully as the quality and quantity of information declines. Reasoning from partial information is important for three reasons: (1) it may be expensive, time-consuming, or impossible to get complete information; (2) computing with exact information may be too complex; (3) reasoning with partial information allows the inference of general rules that apply across a wide class of cases. For instance, suppose a person is driving a car at the top of a cliff, and the driver wishes to determine whether it would be better to take the winding road to the bottom or to drive down the cliff. Given exact specifications of the car and the exact topography of the cliff, it may be possible to predict exactly what would happen if the car was driven off the cliff. But the driver may not have this information or any way to get it; even if the driver had the information, the computation would be horrendous; the conclusion would apply only to a specific car and a specific cliff. The calculation would have to be redone for each new car and each new cliff.

TIME

Temporal reasoning is probably the most central issue in commonsense reasoning. Almost every application involves reasoning about time and change; few microworlds of interest are purely static.

The first task of a temporal representation is to express changes over time; for example, to represent such facts as "At one time, the light was off; later, it was on." Such a representation is often based on the concepts of situations and fluents. A situation is an instantaneous snapshot of the world at an instant. A fluent is a description that changes its value from one situation to another, such as "the light being on" or "the president of the United States." A fluent like "the light being on" that has possible values "true" and "false" is called a Boolean fluent or state.

A first-order language (see LOGIC, PREDICATE) for describing situations and fluents can be defined using the following nonlogical symbols:

- True_in(S,A), predicate: state A is true in situation S.
- Value_in(S,F), function: the value of fluent F in situation S.
- Precedes($S1$,$S2$), predicate: situation $S1$ precedes $S2$.

For example, the sentence "At one time the light was off; later it was on," can be expressed in the formula

$$\exists_{S1,S2}\ \text{precedes}(S1,S2) \wedge \neg\text{true\_in}(S1,\text{on}(\text{light1})) \wedge \text{true\_in}(S2,\text{on}(\text{light1}))$$

Most events do not occur instantaneously; they occur over finite stretches of time. To incorporate events into representations, the concept of a time interval, a set of successive situations, is introduced. The following symbols are added to the language:

- $S \in I$, predicate: situation S is in interval I.
- [$S1$,$S2$], function: the closed interval from $S1$ to $S2$.
- Occurs(I,E), predicate: event E occurs during interval I.

Using this language, a variety of dynamic microworlds can be described. For instance, the blocks world rule "If X and Z are clear, then the result of putting X onto Z will be that X is on Z" can be expressed as follows:

$$\forall_{S1,S2,X,Y}\ [\text{true\_in}(S1,\text{clear}(X)) \wedge \text{true\_in}(S1,\text{clear}(Y)) \wedge \text{occurs}([S1,S2],\text{puton}(X,Y))] \Rightarrow \text{true\_in}(S2,\text{on}(X,Y))$$

In developing such a theory of a microworld, where the occurrence of an event changes the state of the world, the following problem is encountered: the theory must specify, not only the fluents that change as a result of an event but also the fluents that remain the same. For example, in the blocks world, it is necessary to infer that, when the robot puts X on Y, the only "on" states affected are those involving X. The problem of expressing or deriving such rules efficiently is known as the "frame" problem (McCarthy and Hayes, 1969); it has been the focus of much recent research (Hanks and McDermott, 1987; Shoham, 1988; Brown, 1987; Pylyshyn, 1987).

The language described above follows McDermott (1982). Many other types of temporal languages have been devised, including languages that use only intervals but no individual situations (Allen 1983, 1984), languages that distinguish sections of space–time (Hayes, 1978), and modal temporal languages (Prior, 1967; van Benthem, 1983).

SPACE

Commonsense spatial reasoning (see REASONING, SPATIAL) serves three major functions:

- *High Level Vision.* The interpretation of visual information in terms of world knowledge and the integration of information gained through vision into a general knowledge base.
- *Cognitive Map Maintenance.* The formation, maintenance, and use of a knowledge base describing the spatial layout of the environment. In particular, the use of a cognitive map for navigation, planning a route to a destination.

- *Physical Reasoning*. Spatial characteristics of physical systems are generally critical in understanding its behavior. The behavior of many physical systems consists largely of spatial motions. Spatial reasoning is, therefore, a vital component of physical reasoning.

Finding a language for spatial knowledge that is both expressive and computationally tractable is difficult. Ideally, a spatial language would allow the description of any physically meaningful spatial layout and spatial behavior, including specifications of shapes, positions, and motions; it would allow the expression of all types of information that are relevant to commonsense reasoning; it would allow a wide range of partial specifications, corresponding to the types of information that may be obtained from perception, natural language text, or physical inference; and it would do all this in a way that supports efficient algorithms for commonsense reasoning. No such language has yet been found.

The following are some of the more extensively studied spatial representations (Requicha, 1980; Ballard and Brown, 1982; Hoffmann, 1989):

1. *Occupancy Array*. The space is divided up into a rectangular grid, and each cell of the grid is associated with one element of an array. Each element of the array holds the name of the object(s) that intersect the corresponding rectangle in space. One disadvantage of this representation is that it is costly in terms of memory. This can be mitigated by the use of quad-trees or oct-trees, which merge adjacent array elements with identical labels.
2. *Constructive Solid Geometry*. A shape is characterized as the union and difference of a small class of primitive shapes.
3. *Boundary Representations*. A shape is characterized in terms of its boundary. For example, the representation might approximate a two-dimensional shape as a polygon, which is defined by listing its edges, its vertices, and the coordinates of the vertices.
4. *Topological Representations*. A spatial layout is characterized by describing topological relations between objects. For instance, the TOUR program (Kuipers, 1978) describes a road map by stating the order in which places appear on a path and the cyclic order in which paths meet at a place. Randell and Cohn (1989) characterize the spatial relations between objects in terms of such relations as abutment and overlapping.

PHYSICAL REASONING

Unlike the sciences, which aim at a simple description of the underlying structure of physical reality, the commonsense theory of physics must try to describe and systematize physical phenomena as they appear and as they may most effectively be thought about for everyday purposes. The problems addressed in physical reasoning include predicting the future history of a physical system, planning physical actions to carry out a task, and designing tools to serve a given purpose.

Commonsense physical reasoning characteristically avoids the use of exact numerical values. Rather, it relies on qualitative characterization of the physical parameters involved, such as "If a kettle of water is placed on a flame, it will heat up; the higher the flame, the faster the water will heat." This rule does not specify the exact rate at which the temperature changes; it specifies that the change is positive, and that the rate is an increasing function of the height of the flame. Accordingly, the mathematical structure of such constraints has been extensively studied (see QUALITATIVE PHYSICS) (de Kleer and Brown, 1985; Kuipers, 1986).

Complex physical systems, particularly artificial devices, can often be effectively analyzed by viewing them as a collections of connected components. In simple cases, the connections between the components remain constant over time; what varies are the values of various one-dimensional parameters of the system. Components are connected at ports; each parameter is associated with one port. The laws that govern these systems are component characteristics, which constrain the values of the parameters at the ports of the component, and connection characteristics, which constrain the values of parameters at ports that meet in a connection. For example, in electronics, the component characteristics are rules such as "The difference between the voltages at the two ends of a resistor is equal to the current through it times its resistance." The connection characteristics are the rules, "At a connection, the voltages of all the ports is equal, and sum of all the current flows into the ports is zero." The particular device is specified by describing the components it contains, and the connections between their ports (de Kleer and Brown, 1985).

An alternative way to decompose physical systems (Forbus, 1985) focuses on the processes that occur. Consider, for example, a closed can of water above a flame. A process-based description of the behavior of this system would say that there is first a heating process, in which the temperature of the water rises to its boiling point; then a boiling process, in which the water turns from a liquid to a gas; then another heating process, in which the temperature and pressure of the gas rise steadily, and finally a bursting event, when the pressure of the gas exceeds the strength of the can. The central elements of such a representation are process types, such as heating and boiling, and parameters, such as temperature and pressure. The laws that govern the system describe how a process influences a parameter, such as "A boiling process tends to reduce the quantity of liquid and increase the quantity of gas"; they describe influences of one parameter on another, such as "The pressure of a gas tends to rise with its temperature"; and they describe the circumstances under which a process can take place, such as "Boiling occurs just if there were a heat flow into a body of water that is at its boiling point."

Reasoning about systems of solid objects involves techniques of a different kind. The central problem here is the geometric reasoning required, which can be very complicated, particularly with three-dimensional objects of complex shapes. However, the physical laws describing such systems are fairly straightforward. Many mechanisms, particularly synthetic devices where the parts are tightly constrained can be analyzed using only the physical laws that solid objects are rigid and cannot overlap. Such an analysis is known as a kinematic analysis (Faltings 1987; Joskowicz, 1987). For loosely constrained systems of solid objects, such as a bouncing ball, it is generally necessary to use dynamic analysis, invoking the concepts of Newtonian mechanics. Liquids are still more complicated to represent and reason about, because they are not divided into discrete objects; they continually combine and separate. Hayes (1985) discusses a logical analysis of a commonsense theory of liquids.

One final issue in physical reasoning is causality. Ordinary discourse about physical events is often framed in terms of one event causing another; however, none of the theories mentioned above make any use of causality as a concept. Extracting a causal account from such theories has proven to be difficult, particularly as there is no consensus on exactly what purpose a causal account should serve (de Kleer and Brown, 1985; Iwasaki and Simon, 1986; Shoham, 1988; Pearl, 1988).

KNOWLEDGE AND BELIEF

To reason about other agents, or even about oneself at other times, it is necessary to have a theory describing their mental life. AI studies of commonsense theories of cognitions have primarily focused on agents' knowledge and beliefs, discussed in this section, and their plans and goals, discussed in the next.

Representations of knowledge and belief must necessarily be quite different in structure from the representations that were considered above for temporal, spatial, and physical information. The relation, "A knows ϕ" takes as its argument a proposition, which may contain Boolean operators, quantifiers, or imbedded statements about knowledge. (Consider, for example, "John knows that Mary knows that all Libras were born in either September or October.") Operators in first-order languages, by contrast, can take as arguments only terms that denote entities. Moreover, first-order operators are referentially transparent; if two terms τ and ω denote the same entity then ω can be substituted for τ in any sentence without changing the truth of the sentence. By contrast, psychological relations, such as "A knows ϕ," are referentially opaque; substitution of equal terms may change the truth of a sentence. For example, given that Sacramento is the capital of California, it follows that "John is in Sacramento" is true if "John is in the capital of California" is true. However, it is possible for "John knows that he is in Sacramento" to be true but "John knows that he is in the capital of California" to be false, if John believes that Los Angeles is the capital of California.

Three general types of representations have been developed for these kinds of relations:

1. *Know* and *believe* can be represented as operators in a modal logic (see LOGIC, MODAL), a logic that extends first-order logic by allowing additional operators on sentences. In such a theory, the sentence mentioned above could be represented

$$\text{know(john, know(mary, } \forall_x \text{ libra}(X) \Rightarrow \text{born\_in}(X,\text{sept}) \\ \vee \text{ born\_}(X,\text{oct})))$$

2. *Know* and *believe* can be represented as first-order predicates that take as arguments a string of characters that spell out the sentence known or believed. The above sentence would be represented

$$\text{know(john, } \{\text{know(mary, } \{\forall_x \text{ libra}(X) \Rightarrow \\ \text{born\_in}(X,\text{sept}) \vee \text{born\_in}(X,\text{oct})\})\})$$

where $\{$ and $\}$ are string delimiters. In this example, the difference between the two theories appears trivial; in fact, there are deep differences in their logical characteristics.

3. Facts about knowledge and belief can be expressed in terms of accessibility relations among possible worlds. A possible world is one conceivable way that the world could be; a fact may be true in one possible world and false in another. A world $W1$ is accessible from world $W0$ relative to the knowledge of agent A if nothing in $W1$ contradicts something that A knows in $W0$. Then the fact that A knows ϕ in world W can be expressed by saying that ϕ is true in every world accessible from W. Thus the statement "John knows in world $W0$ that he is in Sacramento" can be represented

$$\forall_{W1} \text{ know\_acc(john, } W0, W1) \Rightarrow \\ \text{true\_in}(W1,\text{in(john,scaramento)})$$

where the predicate "know_acc(A,$W0$,$W1$)" means that world $W1$ is accessible from world $W0$ relative to the knowledge of A, and "true_in" means the same as in the section on temporal reasoning. The other sample sentence can be represented

$$\forall_{W1,W2} [\text{know\_acc(john,}W0,W1) \\ \wedge \text{ know\_acc(mary,}W1,W2)] \Rightarrow \\ \forall_x \text{ true\_in}(W2,\text{libra}(X)) \Rightarrow [\text{true\_in}(W2,\text{born\_in}(X,\text{sept})) \\ \vee \text{ true\_in}(W2,\text{born\_in}(X,\text{oct}))]$$

Extensive discussions of these representations and their relative merits have been published (Moore, 1985a; Halpern and Moses, 1985; Morgenstern, 1988).

The next problem is to characterize what agents know and believe in a way that supports reasonable commonsense inferences. Most theories to date have been modeled on implicit knowledge and belief. An agent implicitly knows ϕ if, in principle, he has enough information to determine ϕ; that is, if ϕ is a logical consequence of facts that he knows. However, in many situations, such as teaching, implicit knowledge is not a reasonable theory; a teacher who assumes that the students can immediately perceive all the consequences of what is said will be disappointed. It has been difficult to find more psychologically plausible theories of knowledge and belief that accommodate the fact that reasoners are limited in the speed and power of their inferential abilities (Konolige, 1985; Levesque, 1984).

Other issues that have been studied in AI theories of knowledge include the gaining of knowledge through perception (Davis, 1988) and the auto-epistemic inference, which allows an agent to infer that ϕ is false from the fact that he does not know ϕ (Moore, 1985b).

PLANS AND GOALS

The second major focus of AI commonsense psychological theories has been in representing and reasoning about plans and goals. Plans and goals have been studied primarily in connection with two high level tasks: plan construction, the problem of finding actions that an agent can perform to accomplish the goal; and motivation analysis, the problem of explaining an agent's actions in terms of plans and goals. The chief problems in analyzing plans and goals are the following:

- Constructing a language to describe plans and goals and defining what it means to carry out a plan or to accomplish a goal described in the language.
- Characterizing the feasibility of a plan, the validity of a plan for achieving a given goal, and the cost of a plan.
- Characterizing the typical high level goals of human actors.
- Giving criteria for evaluating alternative explanations of actions in terms of plans and goals.
- The problem of searching for the best plan in plan construction or the best explanation in motivation analysis.

Most of the planning literature in AI has assumed a particularly simple model of plans and goals. A goal is taken to be a desired state of the world, such as "Block C is on block B" or "John is home." A plan is taken to be a sequence of primitive actions, actions that can be directly carried out by a low level robotic controller. For example, in the blocks world, the action, "Put X on Y" could be taken to be primitive. Plans would then be sequences of "put on" instructions such as "First put A on the table; then put C on B." Furthermore, it is assumed that the planner is omniscient and knows everything that can possibly be relevant. Under these assumptions, the definitions of feasibility and correctness are straightforward: a plan is feasible if the preconditions of each successive action hold at the time that the action is scheduled to be performed; a plan accomplishes a goal if the goal holds after all the actions have been performed. The main problem is then one of search: finding a correct plan, given a starting situation and a goal. More sophisticated theories generalize this basic notion of plans and goals in a number of different ways:

1. A plan may only partially specify the actions to be taken, leaving details to be completed at execution time. Consider, for example, a plan to mail a letter consisting of five steps: (*1*) Insert the letter in an envelope, (*2*) address the envelope, (*3*) attach a stamp to the envelope, (*4*) seal the envelope, and (*5*) put the envelope in a mail box. When forming the plan, it is probably not necessary to identify exactly which envelope, stamp, and mail box should be used; when the plan is executed, the most convenient objects of these types may be chosen. Moreover, the steps need not be totally ordered at planning time. As long as step 4 follows 1, and 5 is the last operation, other ordering relations among the steps may be chosen at execution time (Sacerdoti, 1975; Chapman, 1987).

2. If it will be necessary for an agent to achieve goals of a similar form repeatedly, it may be worthwhile constructing a generic plan, that will accomplish these goals in all circumstances, rather than planning each case individually. For example, in the blocks world, it is possible to construct the generic plan "Clear block X; clear block Y; put X on Y" for achieving the goal "X on Y." (Sussman, 1975; Manna and Waldinger, 1987.)

3. If planners that are not omniscient, but have only partial knowledge of the environment are considered, then the analysis of plans becomes more complicated in several respects. First, in this context, plans and goals become referentially opaque operators, like knowledge and belief. John may plan or wish to go to the capital of California and yet not plan or wish to go to Sacramento, if he does not know that they are the same place. It is, therefore, necessary to use one of the techniques described in the previous section (modal logic, syntactic operators, or possible worlds) to represent the plans and goals of agents who are not omniscient.

Planners with partial knowledge must also deal with circumstances in which the planner must gain information to achieve the goal. For example, if John wants to call Mary, but does not know her phone number, he may construct the plan, "First look up Mary's number in the phone book; then dial that number." The analysis of this kind of plan is known as the knowledge preconditions problem (Moore, 1985a; Morgenstern, 1988).

4. For either generic plans or planning with partial knowledge, it may be useful to augment the planning language so that a plan can specify actions that depend on the state of the world. For this purpose, it may be useful to introduce operators similar to those of programming languages, such as conditionals, loops, variable binding, interrupts, and so on.

In order to carry out motivation analysis (the explanation of an agent's actions in terms of goals and plans) it is necessary to have a theory that describes characteristic goals. Otherwise, it would be possible to explain any action as done for the fun of it. Schank and Abelson (1977) suggest five general categories of top-level goals:

Satisfaction Goals. Basic physical needs, such as hunger, thirst, and fatigue, that arise periodically.

Preservation Goals. The desire to preserve certain key personal states, such as preservation of life, health, and possessions.

Achievement Goals. Large-scale ambitions accomplished over a long term, such as raising a family or success in a career.

Entertainment Goals. The short-term enjoyment of some activity, such as seeing a movie.

Delta Goals. The acquisition of certain goods, particularly wealth and knowledge.

OTHER ISSUES

Other commonsense domains that have been studied in the AI literature include emotions (Dyer, 1983; Sanders, 1989), interactions among agents (Wilensky, 1983; Bond and Gasser, 1988), communication (Perrault and Allen, 1980), and thematic relations between people (Schank and Abelson, 1977).

BIBLIOGRAPHY

J. Allen, "Maintaining Knowledge about Temporal Intervals," *Comm. ACM* **28,** 832–843 (1983).

J. Allen, "Towards a General Theory of Action and Time," *Artif. Intell.* **23,** 123–154 (1984).

D. Ballard and C. Brown, *Computer Vision,* Prentice Hall, Inc., Englewood Cliffs, N.J., 1982.

A. Bond and L. Gasser, *Readings in Distributed Artificial Intelligence,* Morgan-Kaufmann, San Mateo, Calif., 1988.

F. Brown, ed., *The Frame Problem in Artificial Intelligence: Proceedings of the 1987 Workshop,* Morgan-Kaufmann, San Mateo, Calif., 1987.

D. Chapman, "Planning for Conjunctive Goals," *Artif. Intell.* **32,** 333–378 (1987).

E. Davis, "Inferring Ignorance from the Locality of Visual Perception," in *Proceedings of the Seventh National Conference on Artificial Intelligence,* St. Paul, Minn., AAAI, Menlo Park, Calif., 1988, pp. 786–790.

J. de Kleer and J. S. Brown, "A Qualitative Physics Based on Confluences," in D. Bobrow, ed., *Qualitative Reasoning about Physical Systems,* MIT Press, Cambridge, Mass., 1985, pp. 7–84.

M. Dyer, *In-Depth Understanding—A Computer Model of Integrated Processing for Narrative Comprehension,* MIT Press, Cambridge, Mass., 1983.

B. Faltings, "Qualitative Kinematics in Mechanisms," *Proceedings of the Tenth IJCAI,* Milan, Italy, Morgan-Kaufmann, San Mateo, Calif., 1987, pp. 1331–1336.

K. Forbus, "Qualitative Process Theory," in D. Bobrow, ed., *Qualitative Reasoning about Physical Systems,* MIT Press, Cambridge, Mass., 1985, pp. 85–168.

J. Halpern and Y. Moses, "A Guide to the Modal Logics of Knowledge and Belief," in *Proceedings of the Ninth IJCAI,* Los Angeles, Morgan-Kaufmann, San Mateo, Calif., 1985, pp. 480–490.

S. Hanks and D. McDermott, "Nonmonotonic Logic and Temporal Projection," *Artif. Intell.* **33,** 379–412 (1987).

P. Hayes, "In Defense of Logic," in *Proceedings of the Fifth IJCAI,* Cambridge, Mass., Morgan-Kaufmann, San Mateo, Calif., 1977, pp. 559–565.

P. Hayes, "The Naive Physics Manifesto," in D. Michie, ed., *Expert Systems in the Micro-Electronic Age,* Edinburgh University Press, Edinburgh, UK, 1978.

P. Hayes, "Naive Physics 1: Ontology for Liquids," in J. Hobbs and R. Moore, eds., *Formal Theories of the Commonsense World,* Ablex, Norwood, N.J., 1985, pp. 71–108.

C. Hoffmann, *Geometric and Solid Modeling: An Introduction,* Morgan-Kaufmann, San Mateo, Calif., 1989.

Y. Iwasaki and H. Simon, "Causality in Device Behavior," *Artif. Intell.* **29,** 3–32 (1986).

L. Joskowicz, "Shape and function in Mechanical Devices," in *Proceedings of the Sixth National Conference on Artificial Intelligence,* Seattle, Wash., AAAI, Menlo Park, Calif., 1987, pp. 611–618.

K. Konolige, "Belief and Incompleteness," in J. Hobbs and R. Moore, eds., *Formal Theories of the Commonsense World,* Ablex, Norwood, N.J., 1985, pp. 71–108.

B. Kuipers, "Modeling Spatial Knowledge," *Cogn. Sci.* **2**(2), 129–154 (1978).

B. Kuipers, "Qualitative Simulation," *Artif. Intell.* **29,** 289–338 (1986).

H. Levesque, "A Logic of Explicit and Implicit Belief," in *Proceedings of the Fourth National Conference on Artificial Intelligence,* Austin, Tex., AAAI, Menlo Park, Calif., 1984.

J. McCarthy, "Programs with Common Sense," in *Proceedings of the Symposium on Mechanisation of Thought Processes,* Vol. 1, London, 1959.

J. McCarthy, "Programs with Common Sense," in M. Minsky, ed., *Semantic Information Processing,* MIT Press, Cambridge, Mass., 1968, pp. 403–418.

J. McCarthy and P. Hayes, "Some Philosophical Problems from the Standpoint of Artificial Intelligence," in B. Meltzer and D. Michie, eds., *Machine Intelligence,* Vol. 4, Edinburgh University Press, Edinburgh, UK, 1969, pp. 463–502.

D. McDermott, "Tarskian Semantics, or No Notation Without Denotation!" *Cogn. Sci.* **2**(3), 277–282 (1978).

D. McDermott, "A Temporal Logic for Reasoning about Processes and Plans," *Cogn. Sci.* **6,** 101–155 (1982).

D. McDermott, "A Critique of Pure Reason," *Computat. Intell.* **3,** 151–160 (1987).

Z. Manna and R. Waldinger, "A Theory of Plans," in M. Georgeff and A. Lansky, eds., *Reasoning about Actions and Plans,* Morgan-Kaufmann, San Mateo, Calif., 1987.

M. Minsky, "A Framework for Representing Knowledge," in P. Winston, ed., *The Psychology of Computer Vision,* McGraw-Hill Book Co., Inc., New York, 1975.

R. Moore, "A Formal Theory of Knowledge and Action," in J. Hobbs and R. Moore, eds., *Formal Theories of the Commonsense World,* Ablex, Norwood, N.J., 1985a, pp. 319–358.

R. Moore, "Semantical Considerations on Nonmonotonic Logic," *Artif. Intell.* **25,** 75–94 (1985b).

L. Morgenstern, *Foundations of a Logic of Knowledge, Action, and Communication,* Ph.D. dissertation, New York University, 1988.

J. Pearl, *Probabilistic Reasoning in Intelligent Systems; Networks of Plausible Inference,* Morgan-Kaufmann, San Mateo, Calif., 1988.

C. Perrault and J. Allen, "A Plan-Based Analysis of Indirect Speech Acts," *Am. J. Computat. Ling.* **6,** 167–182 (1980).

A. N. Prior, *Past, Present, and Future,* Clarendon Press, Oxford, UK, 1967.

Z. Pylyshyn, *The Frame Problem and Other Problems of Holism in Artificial Intelligence,* Ablex, Norwood, N.J., 1987.

D. A. Randell and A. G. Cohn, "Modeling Topological and Metrical Properties in Physical Process," in *Proceedings of the Firt International Conference on Principles of Knowledge Representations and Reasoning,* Toronto, 1989.

A. A. G. Requicha, "Representations for Rigid Solids: Theory, Methods, and Systems," *ACM Comput. Surv.* 12(4), 437–464 (1980).

E. Sacerdoti, *A Structure for Plans and Behaviors,* Elsevier Science Publishing Co., Inc., New York, 1975.

K. Sanders, "A Logic for Emotion," in *Proceedings of the Conference for Cognitive Science,* Ann Arbor, Mich., 1989, pp. 357–363.

R. Schank and R. Abelson, *Scripts, Plans, Goals, and Understanding,* Lawrence Erlbaum, Hillsdale, N.J., 1977.

Y. Shoham, *Reasoning about Change,* MIT Press, Cambridge, Mass., 1988.

G. Sussman, *A Computer Model of Skill Acquisition,* Science Publishing Co., Inc., New York, 1975.

J. van Bethem, *The Logic of Time,* Reidel, Dordrecht, 1983.

R. Wilensky, *Planning and Understanding,* Addison-Wesley Publishing Co., Inc., Reading, Mass., 1983.

General References

R. Brachman and H. Levesque, eds., *Readings in Knowledge Representation,* Morgan-Kaufman, San Mateo, Calif., 1985. Reprints of many classic articles and extensive bibliography.

E. Charniak and D. McDermott, *Introduction to Artificial Intelligence,* Addison-Wesley Publishing Co., Inc., Reading, Mass., 1985. Chapters 1, 6, and 7 give an excellent introduction to knowledge representation and commonsense reasoning.

E. Davis, *Representations of Commonsense Knowledge,* Morgan-Kaufmann, San Mateo, Calif., 1990. Comprehensive textbook.

M. Genesereth and N. Nilsson, *Logical Foundations of Artificial Intelligence,* Morgan-Kaufmann, San Mateo, Calif., 1988. Extensive discussion of logic, nonmonotonic inference, and knowledge representation.

J. Hobbs and R. Moore, *Formal Theories of the Commonsense World,* Ablex, Norwood, N.J., 1985. Collection of research papers.

D. Lenat and R. Guha, *Building Large Knowledge-Based Systems: Representation and Inference in the CYC Project,* Addison-Wesley Publishing Co., Inc., Reading, Mass., 1990. Description of the CYC program, a large knowledge base for commonsense knowledge.

D. Weld and J. de Kleer, *Qualitative Reasoning about Physical Systems,* Morgan-Kaufmann, San Mateo, Calif., 1989. Extensive collection of research papers on physical reasoning.

ERNEST DAVIS
Courant Institute

REASONING, DEFAULT

A main goal of AI research is the construction of programs capable of displaying commonsense behavior. The work on default reasoning purports to contribute to this goal in two ways: by developing frameworks for understanding the nature and form of inference patterns that rely on assumptions, and by developing representation languages and inference procedures for capturing such patterns in AI programs.

What is a default inference? It is an inference that relies on hidden assumptions. For example, the expectations that the car is now where it was last parked and that the car is going to start when the ignition key is turned are default inferences. Both presume that certain facts, like the car being stolen or the battery being dead, are not true. Because these assumptions usually hold, the expectations they support will hold as well, permitting appropriate action plans to be adopted. On the other hand, if the assumptions are found to be wrong, the expectations will be revised and new plans of actions will be considered.

In order to capture this type of behavior in systems that encode knowledge declaratively, two issues need to be addressed. First, a language is needed in which to express both categorical and default knowledge. For example, it is desirable to be able to express things such as "normally, if the ignition key is turned, the car will start," "the car will not start if the battery is dead," and so on. Second, it is necessary to have a semantics for the language, ie, a specification of the legitimate expectations that default gives rise to. The language and the semantics must also provide meaningful primitives with an interpretation that must correspond with the intuitions of the knowledge base builder. As will be noted later, the failure of a number of nonmonotonic logics to accomplish this goal has been a main driving force of much of the recent work in the area.

Among the knowledge representational languages proposed to accommodate some form of default inference, two groups can be distinguished. On the one hand, are special-purpose systems that evolved from the experimental work in AI. These include inheritance networks, truth-maintenance (qv) systems, and logics programs with negation as failure. On the other hand are formalisms derived from classical logic which aim to capture default inference as classical logic captures deductive inference. The special-purpose systems will be reviewed first, and then the more general formalisms, which draw their main intuitions from the former.

SYSTEMS THAT REASON BY DEFAULT

A number of systems in AI involve forms of default inference. These systems point to the type of inferences that an adequate formal account of default inference must accommodate and illustrate that even without a complete understanding of defaults a lot is already known about how they are used in commonsense reasoning. In this section some of these systems will be considered, focusing on databases, inheritance hierarchies (qv), general logic programs, and truth-maintenance systems.

Databases

Databases are systems designed for the efficient storage and retrieval of information about objects and their relations. A departmental database, for example, may contain a relation *teach* with two tuples ⟨*gray, c*⟩ and ⟨*kay, lisp*⟩, indicating that Professor Gray teaches C and Professor Kay, LISP. Relations and tuples are understood as encoding ground atoms in classical first-order logic; in this case, the atoms *teach*(*gray, c*) and *teach*(*kay, lisp*). Thus if queried about who teaches C or PASCAL, the answer *gray* can be understood from the fact that the atomic encoding of the database sanctions the sentence $teach(gray, c) \vee teach(gray, pascal)$ as a theorem.

The logic of databases, however, involves more than atoms. For example, conclusions such as "*kay* does not teach *c*" and "only *gray* teaches *c*," do not follow from the atoms contained in the database. To account for such conclusions, the atomic encoding of the database must be augmented with certain assumptions about both the names of the objects and the world that the database is supposed to represent. These are the unique names assumption, by which individuals with distinct names are assumed distinct, the domain closure assumption, by which all individuals are assumed named, and the closed world assumption, by which it is assumed that there are no more instances of a relation than those deducible from the database (Reiter, 1984). Provided with these assumptions, the conclusions supported by the database will now be theorems of its logical encoding. This logical encoding, however, is not incremental; more information in the database (eg, a new class on C taught by Kay) will render some of the former assumptions false. This is not surprising though; the behavior of the database changes nonmonotonically (ie, more information sometimes implies fewer conclusions), whereas the behavior of its logical encoding can only change monotonically.

Inheritance Hierarchies

Inheritance hierarchies (qv) are directed acyclic graphs used to represent subsumption relations among classes of objects (Touretzky, 1986). Nodes in the hierarchy stand for individual objects or classes, and links stand for class membership (if they connect an individual to a class) or class subsumption (if they connect a class to another class). The concept of inheritance hierarchies originated in the work on semantic networks and in recent years has found applications in both programming and knowledge representation languages.

Figure 1 depicts a simple inheritance network. The network involves two types of link: positive links (→), which assert that one class is a (not necessarily strict) subclass of another (eg, birds are flying things), and negative links (↛), which assert that one class is a (not necessarily strict) subclass of the complement of another (eg, penguins are not flying things). Classes are assumed to inherit the properties of their superclasses unless otherwise specified. In the net depicted in Figure 1, for example, canaries are assumed to inherit the property "fly" from birds, just as penguins are assumed to inherit the property "animal." On the other hand, penguins do not inherit the property "fly" from birds, because the link from penguins to the complement of "fly" is more specific than the link from birds to "fly" and overrides the default inheritance path "penguin → bird → fly." This preference for more specific information in cases of conflict is at the core of inheritance algorithms and points to an important aspect of default inference that an adequate account of defaults must be able to capture.

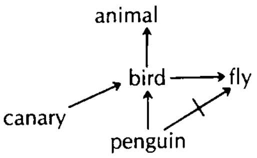


Figure 1. A simple inheritance hierarchy.

Negation as Failure

Logic programs are collections of implicitly universally quantified rules of the form $A \leftarrow L_1, L_2, \ldots, L_n$, where A is an atom called the head of the rule and each L_i, $i = 1, \ldots, n$, $n \geq 0$ is positive or negative literal in the rule's body. When the rules only involve positive literals, logic programs can be given both a procedural and a declarative reading: a rule $A \leftarrow L_1, L_2, \ldots, L_n$ can be understood as stating either that A is true when the literals L_i, $i = 1, \ldots, n$ are true, or that the goal A can be derived by deriving each of the subgoals L_i, $i = 1, \ldots, n$. When some of the literals L_i are negative, however, things are not so simple, and the declarative reading of logic programs is usually dropped. Such programs are commonly understood in procedural terms, with negative literals $\neg A_i$ assumed to be derivable when the derivation for the atom A_i finitely fails (Clark, 1978). Such form of negation as failure has turned out to be particularly useful in programming and follows a tradition that goes back to PLANNER-like languages (Hewitt, 1972). The effect of negation as failure is to assume that negative literals hold by default. Logical accounts of such a behavior have been recently developed, and will be discussed in the next section.

Truth Maintenance Systems

Truth maintenance systems (qv) (TMSs) keep track of dependencies among propositions and often perform some type of inference (Doyle, 1979; de Kleer, 1986). See also the bibliography in Martins (1991). In Doyle's TMS, a user expresses justifications among propositions in a restricted propositional language and the TMS generates a labeling where each proposition is believed (IN) or not (OUT), according to whether or not the proposition has a valid justification. Each justification is made up of two lists of propositions, an IN list and an OUT list, and is valid when each proposition in the IN list and no proposition in the OUT list has a valid justification. To avoid circularities, admissible labelings are also required to be well founded, or what amounts the same, to be minimal in the set of propositions that are believed.

As an example, consider the following justifications (syntax is IN list | OUT list → prop):

$$J_1 : D | H \rightarrow W$$

$$J_2 : C | \rightarrow H$$

$$J_3 : \rightarrow D$$

stating that (*1*) "if today is a working day and it is not believed that John is at home, then John is at work"; (*2*) "if John's car is parked in front of his home, then John is at home"; and (*3*) "today is a working day." The TMS algorithm will then label both D and W as IN, and H as OUT. If a new justification → C ("John's car is parked in front of his home") is added, however, H will become IN, defeating the default justification for W and thus forcing W to go OUT.

Although long understood in procedural terms, some formal accounts of the TMS belief revision process have been recently advanced (Elkan, 1988; Reinfrank and coworkers, 1989). More interestingly, such accounts reveal that Doyle's TMS is not very different from a propositional logic program and that admissible TMS labeling are nothing else but stable models (Gelfond and Lifschitz, 1988) of the logic program that results from mapping each TMS justification $p_1, \ldots, p_n | q_1, \ldots, q_m \rightarrow p$ into a logic programming rule $p \leftarrow p_1, \ldots, p_n, \neg q_1, \ldots, \neg q_m$.

NONMONOTONIC LOGICS

The systems reviewed in the previous section all embed a default component. Still, such a component is the result of well-crafted algorithms tailored to specific languages and tasks. Nonmonotonic logics, on the other hand, were developed with the aim of providing general-purpose languages for representing and reasoning with defaults (see LOGIC, NONMONOTONIC). For that they need to address mathematical as well as epistemological issues. The mathematical issues arise because default reasoning, unlike deductive reasoning, is nonmonotonic; namely, conclusions sometimes need to be retracted in the light of new information. This implies for instance that proofs, if there are such things in the context of nonmonotonic logics, will be quite different from proofs in classic logic: they will not only depend on the information which is present in the knowledge base, but on information which is absent as well. The epistemological issues, on the other hand, arise because defaults, much like the standard logical connectives, possess a meaning to the builder of the knowledge base that an adequate logical account must be able to reflect. Logical accounts of defaults that predict conclusions that bear no relation to those intended by the user could have some mathematical interest, but will certainly have little value for knowledge representation.

In this section some of the standard nonmonotonic formalisms will be reviewed, and how they relate to one another and to the systems reviewed in the previous section will be discussed.

Default Logic

Reiter's (1980) default logic extends classic first-order logic with tentative rules of inference of the form:

$$\frac{\alpha(x):\beta(x)}{\gamma(x)}$$

where $\alpha(x)$, $\beta(x)$, and $\gamma(x)$ are formulas with free variables among those of $x = \{x_1, x_2, \ldots\}$, called the precondition, the test condition and the consequent of the default, respectively. For a tuple a of ground terms, such a default permits $\gamma(a)$ to be derived from $\alpha(a)$, provided that $\neg\beta(a)$ is not derivable. For instance, a default

$$\frac{bird(x):flies(x)}{flies(x)}$$

yields the conclusion $flies(Tim)$ from $bird(Tim)$. However, if the negation of $flies(Tim)$ is observed, the default gets blocked and the former conclusion is no longer supported.

The appeal to nonderivability in the body of defaults together with their use to extend the set of derivable sentences lead, in certain cases, to conflicts among defaults. For instance, given a second default:

$$\frac{injured(x):\neg flies(x)}{\neg flies(x)}$$

and that Tim is injured, two defaults become applicable. However, if the first one is applied, the second one becomes blocked, and vise versa. Reiter deals with those situations by introducing the notion of extensions of a default theory $T = \langle W, D\rangle$, for a set of wffs W and a set of defaults D.

Formally, if it is assumed that $\Gamma(S)$ expands a set of wffs S according to T when $\Gamma(S)$ stands for the minimal deductively closed set of wffs that includes W and every consequent γ of defaults $\alpha:\beta/\gamma$ in D for which $\alpha \in \Gamma(S)$ and $\beta \notin S$, then an extension of T is a set E of wffs that expands into itself, ie, $E = \Gamma(E)$. A default theory $T = \langle W, D\rangle$ may give rise to one, none, or many extensions, and each one reflects a possible "completion" of the classic theory W, according to the defaults in D. In the example above, two different extensions arise, one in which Tim flies and one in which he does not. Default logic has been used for specifying the behavior of inheritance hierarchies with exceptions (Etherington and Reiter, 1983) and logic programs with negation (Bidoit and Fridevaux, 1987). The complexity of reasoning in default logic has been recently studied (Kautz and Selman, 1989).

Circumscription

Circumscription (qv) is a formal device for asserting that the objects that can be shown to satisfy a certain predicate P in a given first-order theory are the only objects that do (McCarthy, 1980, 1986). For instance, from a database only containing the fact $Q(a)$, the circumscription of Q yields the formula $\forall x.\ Q(x) \Rightarrow x = a$. Thus if b is an object different from a, the circumscription of Q sanctions $\neg Q(b)$ as a conclusion. If $Q(b)$ is learned, however, $\neg Q(b)$ goes away and the new conclusions turn out to be those derivable from the formula $\forall x.\ Q(x) \Leftrightarrow x = a \vee x = b$. Circumscription thus behaves as a powerful, adaptable, closed-world assumption, capable of dealing with theories that are richer than those expressible in databases.

Formally, if $A(P)$ stands for a first-order sentence containing the predicate P, and $A(\Phi)$ denotes the sentence that results from replacing all the occurrences of P by a predicate Φ with the same arity as P, the circumscription Circ$[A(P); P]$ of P in $A(P)$ is given by the second-order schema:

$$A(P) \wedge A(\Phi) \wedge \forall x.\ [\Phi(x) \Rightarrow P(x)] \Rightarrow \forall x.\ (P(x) \Rightarrow \Phi(x))$$

that asserts that among the predicates Φ satisfying the constraint $A(\Phi)$, P is the strongest. Thus if $A(Q)$ is the sentence $Q(a)$, for example, the substitution of $\Phi(x)$ by the predicate $\Phi^*(x) : \lambda x.\ (x = a)$ in the schema renders the formula

$$Q(a) \wedge a = a \wedge \forall x.\ [x = a \Rightarrow Q(x)] \Rightarrow \forall x.\ Q(x) \ \Rightarrow x = a$$

which simplifies to $\forall x.\ Q(x) \Leftrightarrow x = a$. The predicate $\Phi^*(x)$ is indeed the strongest predicate that satisfies $A(\Phi)$, and thus the effect of circumscribing Q in $A(Q)$ is to set Q to Φ^*.

Circumscription accommodates nonmonotonic forms of reasoning but does not uniquely specify how defaults should be encoded. For that purpose McCarthy (1986) introduced a convention by which defaults such as "birds fly" are encoded in the circumscriptive framework as formulas

$$\forall x.\ bird(x) \wedge \neg ab_i(x) \Rightarrow flies(x)$$

read as "every nonabnormal bird flies." Once defaults are so expressed, the expected behavior is obtained by circumscribing the ab_i predicates or, as McCarthy prefers to say, by "minimizing abnormality." However, before that can be done effectively, a more powerful form of circumscription is needed in which certain predicates can be minimized at the expense of others.

The circumscription Circ[$A(P, Z)$; P, $\mathbf{Z}$] of the predicate P in the sentence $A(P, \mathbf{Z})$, where $\mathbf{Z}$ stands for a tuple of predicates allowed to vary in the minimization of P, is defined by the following second-order formula (McCarthy, 1986):

$$A(P, \mathbf{Z}) \wedge \forall \Phi, \Psi\ A(\Phi, \Psi) \wedge \forall x.\ [\Phi(x) \Rightarrow P(x)] \Rightarrow \forall x.\ [P(x) \Rightarrow \Phi(x)]$$

This formula permits us to minimize the extension of P at the expense of the extension of the predicates in $\mathbf{Z}$. Indeed, the formula Circ[$A(P, \mathbf{Z})$; P, $\mathbf{Z}$] can be shown to sanction as theorems the sentences that hold in all models of the sentence $A(P, \mathbf{Z})$, which are minimal in P with respect to $\mathbf{Z}$ (Lifschitz, 1985; Etherington, 1988). A model M of $A(P, \mathbf{Z})$ is minimal in P with respect to $\mathbf{Z}$ if there are no other models M' of $A(P, \mathbf{Z})$, which assign a smaller extension to P and which preserve from M the same domain and the same interpretation of symbols other than P and those in $\mathbf{Z}$.

The generalization of circumscription for dealing with multiple predicates, known as parallel circumscription, is straightforward. More interesting is the case of prioritized circumscription in which the user is allowed to specify a priority ordering among the circumscribed predicates (McCarthy, 1986; Lifschitz, 1985). For instance, the circumscription Circ[A; $P_1 > P_2 > \ldots > P_n$; Z] of predicates $P_1, P_2, \ldots, P_n$ in decreasing order of priority, translates into the conjunction of $n - 1$ circumscriptions of the form Circ[A; P_i; $\mathbf{Z} \cup \{P_{i+1}, \ldots, P_n\}$] together with Circ[$A$; P_n; $\mathbf{Z}$]. Predicates of higher priority are thus circumscribed at the expense of predicates of lower priority. Although it is not clear in general how priorities among predicates are to be selected, general guidelines for the domains of logic programs and inheritance hierarchies have been proposed (Lifschitz, 1988; Krishnaprasad and co-workers, 1989). A proof theory for prioritized circumscription has also been developed (Baker and Ginsberg, 1989; Geffner, 1990).

Autoepistemic Logic

Autoepistemic logic is a nonmonotonic extension of classic logic proposed by Moore (1985) as a reconstruction of McDermott and Doyle's (1980) nonmonotonic logic. Since then, autoepistemic logic has received growing attention and has been studied by a number of researchers. Autoepistemic logic deals with autoepistemic theories: propositional theories augmented by a belief operator L, where sentences of the form $L\alpha$ are read as "α is believed." The stable expansions of an autoepistemic theory T are defined as the sets of formulas $S(T)$, which satisfy the equation

$$S(T) = \mathrm{Th}(T + \{Lp : p \in S(T)\} + \{\neg Lp : p \notin S(T)\})$$

where Th(X) stands for the set of tautological consequence of X. Stable expansions are supposed to reflect possible states of belief of an ideal rational agent, closed both under positive and negative introspection. A default such as "if it is a bird, it flies" can be encoded in autoepistemic logic as a sentence $bird \wedge \neg Lab_i \Rightarrow flies$. Then, given $bird$, the only autoepistemic expansion will contain the autoepistemic sentence $\neg Lab_i$ and, therefore, the target sentence $flies$.

An autoepistemic theory may have one, none, or many stable expansions. For instance, a theory such as $T = \{\neg Lp \Rightarrow p\}$ has no stable expansions, whereas a theory $T' = \{\neg Lp \Rightarrow q, \neg Lq \Rightarrow p\}$ has two. Autoepistemic logic has been successfully applied to characterize the semantics of general logic programs (Gelfond and Lifschitz, 1988) and truth maintenance systems (Elkan, 1988; Reinfrank and co-workers, 1989). Both characterizations are natural and simple, requiring in essence the replacement of logic negation ($\neg p$) by autoepistemic negation ($\neg Lp$). They also suggest how to compute with certain classes of autoepistemic theories, and due to the close relation between autoepistemic and default logic (Konolige, 1988; Marek and Truszczynski, 1989), how to compute with certain class of default theories as well.

RECENT DEVELOPMENTS

Each of the formalisms reviewed in the last section, circumscription, default logic, and autoepistemic logic, extends classic logic with some formal device that permits nonmonotonic forms of reasoning. These formalisms generalize and provide a logical basis to the mechanisms discussed earlier and take us closer to the goal of a general-purpose language for representing and reasoning with defaults. Defaults, however, exhibit features other than nonmonotonicity, which are not always captured by these formalisms. In this section some of these features will be examined and some recent proposals for dealing with them will be discussed.

Model Preference

A source of difficulties for capturing default inference in the standard formalisms was noticed by Hanks and McDermott (1987) who noted that the encoding of theories involving causal relations in the standard formalisms often failed to legitimize conclusions that were otherwise obvious to the user. Their example, known as the "Yale shooting" problem, deals with a person who is alive in a given situation but who is shot with a gun that was loaded an instant earlier. The intuition is that the gun stays loaded and that the person dies as a result of the shooting. However, the encoding in the formalisms analyzed by Hanks and McDermott also accommodates a different outcome in which the fluent "loaded" changes, and the fluent "alive" stays the same. Many proposals have been advanced since then to account for the distinction between intuitive and counterintuitive behavior in the presence of actions and persistences. Some involve variations of the formalisms and encodings used by Hanks and McDermott [eg, Morris (1989) uses nonnormal defaults and stable closures]. Others, such as chronological minimization (Shoham, 1988) have led to alternative ways of specifying nonmonotonic inference relations.

Chronological minimization is a semantic criterion for interpreting theories for reasoning about change in which a preference is established for models that give rise to minimal sets of changes that occur as late as possible. This preference relation is used to define a nonmonotonic entailment relation that unlike classic entailment only considers the overall preferred models of the target theory. In the Yale shooting problem, for example, the class of models in which the person dies turns out to be preferred to the class of models in which the gun gets unloaded, as the change sanctioned by the first class of models occurs later. As a result, the right behavior in the Yale shooting problem is captured.

As discussed above, the idea of a preference relation on models is not foreign to the semantics of circumscription which establishes a preference for models that minimize certain predicates. The difference, however, it that chronological minimization bypasses the circumscriptive axiom altogether, appealing directly to a model-preference criterion. This move has resulted in a more powerful way of specifying nonmonotonic behavior and no significant loss: the circumscriptive axion is not a particular source of insight and it has not been found of general use for computing circumscription (Przymusinski, 1989; Ginsberg, 1989). Model preference, on the other hand, has been found to be a flexible device for specifying nonmonotonic inference in a variety of domains (Morgenstern and Stein, 1988; Selman and Kautz, 1989).

Conditional Logics

Another feature of default reasoning that is not explicitly accounted by standard nonmonotonic formalisms is specificity: when two defaults are in conflict there is usually a preference to accept the conclusion supported by the most specific one. Inheritance hierarchies (Fig. 1) provide plenty of examples. To account for such a behavior in a framework such as circumscription, priorities must be introduced (Krishnaprasad and co-workers, 1989). What is the origin of these priorities or why they are needed, however, are not questions that circumscription addresses. Something similar happens with default and autoepistemic logic that do not provide devices as clean as priorities for capturing specificity preferences.

Some answers to these questions have recently emerged from the field of conditional logic as the similarities between defaults and conditionals have become increasingly apparent (Nute, 1984) (see LOGIC, CONDITIONAL). For example, both conditionals and defaults appear to violate principles such as chaining (from $p \rightarrow q$ and $q \rightarrow r$ derive $p \rightarrow q$), contraposition (from $p \rightarrow q$ derive $\neg q \rightarrow \neg p$), and strengthening the antecedent (from $p \rightarrow r$ derive $p \wedge q \rightarrow r$), all of which are corner stones of classic logic. Similarly, they both seem to obey principles like augmentation (from $p \rightarrow q$ and $p \rightarrow r$ derive $p \wedge q \rightarrow r$), reduction (from $p \rightarrow q$ and $p \wedge q \rightarrow r$ derive $p \rightarrow r$), and cases (from $p \rightarrow r$ and $q \rightarrow r$ derive $p \vee q \rightarrow r$).

In AI, two types of conditional interpretation of defaults have recently been studied. In one, defaults $p \rightarrow q$ are regarded as stating that the conditional probability $P(q|p)$ is arbitrarily high, short of being one (Geffner and Pearl, 1990); in the other $p \rightarrow q$ is regarded as a constraint on model-preference orders, stating that q must be true in all preferred models of p (Delgrande, 1987; Kraus and co-workers, 1988). The benefit of the conditional interpretation of defaults is that they account for certain preferences among conflicting defaults without having to explicate exceptions or priorities. The shortcoming, on the other hand, is that patterns of inference involving independence assumptions are no longer captured. Some recent proposals attempt to combine the benefits of standard nonmonotonic logics and conditional interpretations (Lehmann, 1989; Pearl, 1989). An interesting result is that the conditional aspects of defaults can be captured in the framework of prioritized circumscription by a careful selection of the default priorities (Geffner, 1990).

Architectures for Default Reasoning

Most of the work in default reasoning has been focused on the mathematical and semantic aspects of default inference: how to analyze default inference in logical terms and how to capture the intended meaning of defaults. For default reasoning to be useful in building AI programs, however, there is an additional need for reasoning with defaults: namely computing the valid default consequences as opposed to just specifying what makes them valid. The latter is what the formalisms for default reasoning do; the former is the task of default reasoning architectures. Because of the results that relate inheritance hierarchies, truth maintenance systems, and logic programs with negation to specialized circumscriptive, default, and autoepistemic theories, the former systems can be regarded as specialized default reasoning architectures. More recent work has attempted to use or extend these architectures to cope with more expressive languages. Gelfond and Lifshitz (1989), for example, investigate the compilation of circumscriptive theories into logic programs and Junker and Konolige (1990), the computation of extensions of default and autoepistemic theories by means of a TMS. Similarly, argument-based systems have extended the language of inheritance hierarchies to accommodate both conjunctions and negation and, in certain cases, the full power of first-order classical logic. Default inference is viewed in argument-based systems as the result of the interaction between supporting and rebutting arguments, in analogy to the interaction between positive and negative paths in inheritance-based systems. Most argument-based systems have also addressed the problem of the specificity of arguments (Nute, 1986; Poole, 1985; Loui, 1987). Recent results also show a great resemblance between argument-based systems and the proof theory of prioritized circumscription (Baker and Ginsberg, 1989; Geffner, 1990).

CONCLUSIONS

The first decade of work in default reasoning has witnessed an enormous growth in the understanding of the mathematics of nonmonotonic inference and a significant, although less dramatic, growth in the understanding of the epistemological issues involved. A handful of formal devices are now available for extending classical logic with nonmonotonic features, and there is a much better grasp of their power and limitations. Mechanisms such as truth maintenance, negation as failure, and inheritance are understood in logical terms, and new and more powerful architectures seem forthcoming. The gap between intuitions and formalizations has also gotten narrower, as a better understanding of the standard and new formalisms and a better appreciation of the causal and conditional aspects of defaults have developed.

An area where progress has been scant is applications. Except for a few attempts to deal with theories involving actions and persistences, no default formalism has been tried in realistic domains. This has been, in part, due to the emphasis on getting the theory right first as well as the lack of adequate algorithms. As progress along both dimensions continues, however, it is expected that more interesting applications will follow. The work by Grosof (1988) and Loui (1990), using defaults for reasoning about partially specified probabilities and utilities, are an indication of such a trend.

BIBLIOGRAPHY

A. Baker and M. Ginsberg, "A Theorem Prover For Prioritized Circumscription," in *Proceedings of the Eleventh IJCAI,* Detroit, Mich., Morgan-Kaufmann, San Mateo, Calif., 1989, pp. 463–467.

N. Bidoit and C. Froidevaux, "Minimalism Subsumes Default Logic and Circumscription in Stratified Logic Programming," in *Proceedings of the Symposium on Principles of Database Systems,* 1987.

K. Clark, "Negation as Failure," in H. Gallaire and J. Minker, eds., *Logic and Data Bases,* Plenum Press, New York, 1978, pp. 293–322.

J. de Kleer, "An Assumption-Based Truth Maintenance System," *Artif. Intell.* **28,** 280–297 (1986).

J. Delgrande, "An Approach to Default Reasoning Based on a First-Order Conditional Logic," in *Proceedings in the Sixth National Conference on Artificial Intelligence,* Seattle, Wash., AAAI, Menlo Park, Calif., 1987, pp. 340–345.

J. Doyle, "A Truth Maintenance System," *Artif. Intell.* **12,** 231–272 (1979).

C. Elkan, *A Rational Reconstruction of Nonmonotonic TMSs,* Technical Report, Cornell University, Ithaca, N.Y., 1988.

D. Etherington and R. Reiter, "On Inheritance Hierarchies with Exceptions," in *Proceedings of the Intelligence,* Washington, D.C., AAAI, Menlo Park, Calif., 1983, pp. 104–108.

D. Etherington, *Reasoning with Incomplete Information,* Pitman, London, 1988.

H. Geffner, "Conditional Entailment: Closing the Gap between Defaults and Conditionals," in *Proceedings of the Third International Workshop on Non-Monotonic Reasoning,* South Lake Tahoe, Calif., 1990, pp. 58–72.

H. Geffner and J. Pearl, "A Framework for Reasoning with Defaults," in H. Kyburg, R. Loui, and G. Carlson, eds., *Knowledge Representation and Defeasible Inference,* Kluwer, The Netherlands, 1990.

M. Gelfond and V. Lifschitz, "The Stable Model Semantics for Logic Programming," in *Proceedings 1988 Symposium on Logic Programming,* MIT Press, Cambridge, Mass., 1988, pp. 1070–1080.

M. Gelfond and V. Lifschitz, "Compiling Circumscriptive Theories into Logic Programs," in M. Reinfrank and co-workers, eds., *Proceedings of the Second International Workshop on Non-Monotonic Reasoning,* Berlin, Germany, 1989, pp. 74–99.

M. Ginsberg, "A Circumscriptive Theorem Prover," *Artif. Intell.* **39,** 209–230 (1989).

B. Grosof, "Non-Monotonicity in Probabilistic Reasoning," in J. Lemmer and L. Kanal, eds., *Uncertainty in Artificial Intelligence,* Vol. 2, Elsevier Science Publishing Co., Inc., New York, 1988, pp. 237–249.

S. Hanks and D. McDermott, "Non-Monotonic Logics and Temporal Projection," *Artif. Intell.* **33,** 379–412 (1987).

C. Hewitt, *Description and Theoretical Analysis of Planner: A Language for Proving Theorems and Manipulating Models in a Robot,* Technical Report TR-258, MIT, AI Laboratory, Cambridge, Mass., 1972.

U. Junker and K. Konolige, "Computing the Extensions of Autoepistemic and Default Logic with a Truth Maintenance System," in *Proceedings of the Ninth National Conference on Artificial Intelligence,* Boston, AAAI, Menlo Park, Calif., 1990.

H. Kautz and B. Selman, "Hard Problems for Simple Default Logics," in *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning,* Toronto, Ont., 1989, pp. 189–197.

K. Konolige, "On the Relation between Default Logic and Autoepistemic Logic," *Artif. Intell.* **35,** 343–382 (1988).

S. Kraus, D. Lehmann, and M. Magidor, *Preferential Models and Cumulative Logics,* Technical Report, Hebrew University, Jerusalem, Israel, Aug. 1988.

T. Krishnaprasad, M. Kiefer, and D. Warren, "On the Circumscriptive Semantics of Inheritance Networks," in Z. Ras and L. Saitta, eds., *Methodologies for Intelligent Systems,* Vol. 4, North-Holland, Amsterdam, The Netherlands, 1989.

D. Lehmann, "What Does a Conditional Knowledge Base Entail?" in *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning,* Toronto, 1989, pp. 212–222.

V. Lifschitz, "Computing Circumscription," in *Proceedings of the*

Ninth IJCAI, Los Angeles, Calif., Morgan-Kaufmann, San Mateo, Calif., 1985.

V. Lifschitz, "On the Declarative Semantics of Logic Programs," in J. Minker, ed., *Foundations of Deductive Databases and Logic Programming,* Morgan-Kaufmann, San Mateo, Calif., 1988, pp. 177–192.

R. Loui, "Defeat among Arguments: A System of Defeasible Inference," *Comput. Intell.* **3**(3), 100–106 (1987).

R. Loui, "Defeasible Specification of Utilities," in H. Kyburg, R. Loui, and G. Carlson, eds., *Knowledge Representation and Defeasible Inference,* Kluwer, The Netherlands, 1990.

W. Marek and M. Truszczynski, "Relating Autoepistemic and Default Logics," *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning,* Toronto, 1989, pp. 276–288.

J. Martins, "The Truth, the Whole Truth, and Nothing But the Truth," *AI Mag.* **11**(5) (1991).

J. McCarthy, "Circumscription—A Form of Non-Monotonic Reasoning," *Artif. Intell.* **13,** 27–39 (1980).

J. McCarthy, "Applications of Circumscription to Formalizing Commonsense Knowledge," *Artif. Intell.* **28,** 89–116 (1986).

D. McDermott and J. Doyle, "Non-Monotonic Logic I," *Artif. Intell.* **13,** 41–72 (1980).

R. Moore, "Semantical Considerations on Non-Monotonic Logics," *Artif. Intell.* **25,** 75–94 (1985).

L. Morgenstern and L. Stein, "Why Things Go Wrong: A Formal Theory of Causal Reasoning," in *Proceedings of the Seventh National Conference on Artificial Intelligence,* St. Paul, Minn., AAAI, Menlo Park, Calif., 1988.

P. Morris, "Autepistemic Stable Closures and Contradiction Resolution," in M. Reinfrank and co-workers, eds., *Proceedings of the Second International Workshop on Nonmonotonic Reasoning,* Berlin, 1989, pp. 60–73.

D. Nute, "Conditional Logic," in D. Gabbay and F. Guenthner, eds., *Handbook of Philosophical Logic,* D. Reidel, Dordrecht, 1984, pp. 387–439.

D. Nuto, *LDR: A Logic for Defeasible Reasoning,* Technical Report ACMC Research Report **01-0013**, University of Georgia, Athens, 1986.

J. Pearl, *System Z: A Natural Ordering of Defaults with Tractable Applications to Non-Monotonic Reasoning,* Technical Report, UCLA, Los Angeles, 1989.

D. Poole, "On the Comparison of Theories: Preferring the Most Specific Explanation," in *Proceedings of the Ninth IJCAI,* Morgan-Kaufmann, San Mateo, Calif., 1985, pp. 144–147.

T. Przymusinski, "An Algorithm for Circumscription," *Artif. Intell.* **38,** 49–73 (1989).

M. Reinfrank, O. Dressler, and G. Brewka, "On the Relation between Truth Maintenance and Autoepistemic Logic," in *Proceedings of the Eleventh IJCAI,* Morgan-Kaufmann, San Mateo, Calif., 1989, pp. 1206–1212.

R. Reiter, "A Logic for Default Reasoning," *Artif. Intell.* **12,** 81–132 (1980).

R. Reiter, "Towards a Logical Reconstruction of Relational Database Theory," in M. Brodie, J. Mylopoulos, and J. W. Schmidt, eds., *On Conceptual Modelling,* Springer-Verlag, New York, 1984, pp. 163–189.

B. Selman and H. Kautz, "The Complexity of Model Preference Default Theories," in M. Reinfrank and co-workers, eds., *Proceedings of the Second International Workshop on Nonmonotonic Reasoning,* Berlin, 1989, pp. 115–130.

Y. Shoham, *Reasoning about Change: Time and Causation from the Standpoint of Artificial Intelligence,* MIT Press, Cambridge, Mass., 1988.

D. Touretzky, *The Mathematics of Inheritance Systems,* Pitman, London, 1986.

Hector Geffner
IBM T. J. Watson Research Center

REASONING, MEMORY-BASED

Memory-based reasoning (MBR) is a technique in which artificial intelligence is realized by direct reference to memory (Stanfill and Waltz, 1986, 1988). In MBR, there is no fundamental distinction among learning, reasoning, and remembering. Most forms of AI that benefit from experience use experiences to create an intermediate abstraction, such as a set of rules or a set of weights in a network; MBR retains the actual events. Most memory models use knowledge available at the time of an experience to decide how to store and index it; MBR depends instead on intensive computation at the time an experience is required (Waltz, 1989).

The simplest, best studied form of MBR is as a solution to the classification problem. This problem may be represented as follows: a system is given two sets of objects, a training set and a performance set. Each object is represented by a vector of features (attributes), and each object is assigned to a class. The system has access to all features of all objects, and to the classes of objects in the training set. The task of the system is to assign classes to the objects in the performance set, minimizing the number of incorrect class assignments. The memory-based approach to this problem is to iteratively (1) choose a target object from the performance set, (2) find its nearest neighbor in the training set (the training set object that is most similar to the target), and (3) assign the class of the nearest neighbor to the target. Alternatively, it is possible to (2) find the k nearest neighbors and (3) assign the target's class according to the majority of the neighbors. The key issue in such a system is a computational realization of similarity, usually represented as a distance function that, given a pair of objects, returns a nonnegative number measuring the distance between the objects, subject to the constraint that the distance between an object and itself is zero. If there are n objects in the training set and m objects in the performance set, then it is necessary (in general) to evaluate the distance function for $O(mn)$ object pairs, which usually requires a very fast (parallel) machine. For some similarity measures, there may be efficient algorithms requiring fewer than $O(mn)$ such computations.

Many distance measures are possible. The simplest is *Hamming distance,* which is the number of attributes in which two objects differ. A simple variant is the weighted Hamming distance, in which each attribute is assigned a weight measuring its importance in determining the class of the object. An attribute that was very important would have a weight of 1, whereas an attribute that was totally unimportant would have a weight of 0. Importance can be determined by a variety of statistical measures. If the attributes are numeric, then distance can be measured by the Euclidean distance metric, treating each attribute as an axis in a vector space. The above forms of MBR have been well known in the fields of nonparametric statistics and pattern recognition as the nearest-neighbor and k nearest-neighbor classification methods (Atkeson, 1990). Until the advent of parallel computing, such methods were generally considered computationally too expensive for practical use, but their application may now be considered possible. All these methods use a simple global distance measure: the weight given to each feature is uniform across the entire set of objects. It is possible, however, to use a nonglobal weighting scheme. For example, it is possible to (1) determine the importance of each feature, (2) find the 100 nearest neighbors, (3) determine the importance of each feature in that neighborhood, and (4) recompute the nearest neighbors. In effect, every time a new object is to be classified, a new metric that is appropriate to that object's neighborhood will be created. This ability to dynamically create metrics is an important difference between MBR and the traditional statistical methods. The application of MBR to classification has been studied for the pronunciation of English words (Stanfill and Waltz, 1986; Stanfill, 1987), optical character recognition, the interpretation of job category description on census forms (Creecy and co-workers, 1991), and protein secondary structure prediction (Zhang, Waltz, and Mesirov, 1988).

MBR has also been applied to the task of approximating continuous functions (Atkeson, 1990). In this problem, each object in the training and performance sets is associated with a numeric value. Furthermore, it is known that this numeric value is a continuous function of the features used to describe the objects. Values of this function are known for data in the training set, but not for data in the performance set. The task is to predict, as accurately as possible, the values for the performance set. The simplest application of MBR to this task would be to (1) choose an object from the performance set as the target object, (2) find its nearest neighbor in the training set, and (3) assign the value of the nearest neighbor to the target. The problem with this method is that it can only produce target values that are present in the training set. The resulting function will thus have discontinuities. The solution is to introduce some method of interpolating the results. The best approach at this point appears to be (1) assign each object in the training set a weight depending on its distance from the target object, with the nearer objects receiving higher weights; (2) perform a *weighted quadratic regression,* which produces a quadratic function from the object features to the object values that minimized error in the neighborhood of the target point; and (3) apply the quadratic function to the target point, producing a predicted value. This method has been demonstrated on robotic kinematic problems (Atkeson, 1989). For example, the state of a robot arm can be characterized at any point in time by a set of numbers measuring the angles, angular velocities, torques, and angular accelerations of each of its joints. The function to be approximated would be the state of the arm at a time t milliseconds in the future. The training set then consists of a set of snapshots of the state of the arm at various points in time. Locally weighted quadratic regression, as described above, is then employed.

It is possible to combine primitive MBR units into systems exhibiting fairly complex behavior. One such system modeled a first-grade child learning to read (Stanfill, 1988). The system had an auditory memory (a set of words known by sound, represented phonetically), an initially empty set of words known by spelling, and a set of simple pronunciation rules. When confronted with a new word, the system would first check its spelling memory to see whether the word was already known. If not, it would try to sound out the word by remembering pronunciation rules or by remembering words with similar spellings. It would then check its auditory memory for a word that sounded like its pronunciation and, if one was found, create a new item in its spelling memory linked with the corresponding entry in the auditory memory.

MBR can be thought of as occupying one extreme of a continuum of learning and reasoning methods, varying from abstraction-based to memory-based. At the abstraction-based extremes are traditional rule-based systems, where a human translates knowledge of a domain into an abstraction (a set of rules). Slightly less abstraction-based is machine learning, in which a set of rules are inferred from a set of data; the data themselves are not used at run time. Traditional (parametric) statistics plus nontraditional methods (such as back-propagation learning) produce numerical functions rather than symbolic rules, but are at approximately the same point in the abstraction- to memory-based scale. Radial basis functions use points in a training set in forming the basis of a vector space and are yet more memory based. Nearest-neighbor techniques using global metrics are the next steps in the continuum, because the data themselves are used in conjunction with a precomputed abstraction (the global metric). Finally, nearest-neighbor techniques with metrics constructed on the fly are the most extremely memory based. MBR may be considered a subclass of case-based reasoning. However, many case-based reasoning systems employ significant amounts of domain-specific (abstract) knowledge to organize memory, to retrieve items from memory, and to adapt memories to the specific purposes at hand. It may be possible to consider case-based reasoning as a high level architecture and memory-based reasoning as a low level inference technique.

BIBLIOGRAPHY

C. Atkeson, *Memory-Based Approaches to Approximating Continuous Functions,* in *Proceedings of the Sixth Yale Workshop on Adaptive and Learning Systems,* New Haven, Conn., 1990.

C. Atkeson, "Learning Arm Kinematics and Dynamics," *Ann. Rev. Neurosci.* **12,** 157–183 (1989).

R. H. Creecy, B. Masand, S. Smith, and D. Waltz, "Trading MIPS and Memory for Knowledge Engineering: Automatic Classification of Census Returns on a Massively Parallel Computer," Technical Report TMC-192, Thinking Machines Corp., Cambridge, Mass., 1991.

C. Stanfill, "Memory-Based Reasoning Applied to English Pronunciation," in *Proceedings of the Sixth National Conference on Artificial Intelligence,* Seattle, Wash., AAAI, Menlo Park, Calif., 1987.

C. Stanfill and D. L. Waltz, "Toward Memory-Based Reasoning," *Commun. ACM* **29**(12), 1213–1228 (1986).

C. Stanfill and D. L. Waltz, "Learning to Read: A Memory-Based Model," in *Proceedings of the Case-Based Reasoning Workshop,* Clearwater Beach, Fla., May 1988.

D. L. Waltz, "Is Indexing Used for Retrieval?" in *Proceedings of the Case-Based Reasoning Workshop,* Pensacola Beach, Fla., 1989.

X. Zhang, D. L. Waltz, and J. Mesirov, "Protein Structure Prediction Using Memory-Based Reasoning: A Case Study of Data Exploration," Technical Report RL 88-3, Thinking Machines Corporation, 1988.

Craig Stanfill
Thinking Machines Corporation

REASONING, NONMONOTONIC

In many situations, appropriate reasoning involves drawing default conclusions based on incomplete data. The results of such reasoning are in general unsound, ie, they are not necessarily true even though the data on which they are based may be true. Nonetheless, such reasoning can be important. For instance, a robot may assume that in the absence of information to the contrary its arms may be used to transport objects in the normal manner. Similarly, in normal conversation, the appropriate response to "Did you understand that article?" is not "I understood 90% of it," if in fact the latter speaker understood it all. Even though the indicated response is not strictly false, it is misleading because it encourages the questioner to conclude (unsoundly) that only 90% of the article was understood. That is, in the absence of information to the contrary, the questioner might normally and usefully assume that the respondent is being cooperative rather than misleading. These and other examples strongly suggest that reasoning by default is prevalent throughout commonsense reasoning.

Minsky (1975) coined the term nonmonotonic logic in developing an argument that tools of formal logic are inadequate to the task of representing commonsense reasoning. His argument involved an analysis of the use of defaults. As indicated above, these are conclusions based on the absence of contrary information. Consider the example of concluding from the knowledge that Tweety is a bird that Tweety can fly. Such a conclusion is not necessarily true, although it certainly can be very useful in some situations. It is possible to try to specify precisely those situations in which such a conclusion is sound, but any effort to do so quickly leads to despair. The possible circumstances in which any presumed correct line of reasoning can be defeated astounds: Tweety may be an ostrich, may have a broken wing, may be chained to a perch, may be too weak, etc. Indeed, the problem is virtually the same as that of the well-known qualification problem: the special conditions relevant to determining what may be the case in a complex environment defies precise specification. Although there appears to be a meaningful sense of certain typical situations (such as typical birds being ones which, among other things, can fly), it is notoriously hard to define typicality.

However, Minsky's claim was more than this: he argued that, first, conclusions of the sort given in the Tweety example are contingent on what else is known (eg, if it is already known that Tweety cannot fly, than the opposite conclusion is not made) and that, second, such conclusions do not obey the customary phenomenon of monotonicity of formal systems of logic. That is, a standard logic L has the property that if ϕ is a theorem of L and if L is augmented to L^* by additional axioms, then ϕ remains a theorem of L^*. Indeed the same proof of ϕ in L is a proof of ϕ in L^*. However, commonsense reasoning seems to allow a proof (at least in the form of a tentative supposition) that Tweety can fly, given only that Tweety is a bird, whereas in the augmented state in which it is known also that Tweety cannot fly, no such proof is forthcoming. In effect, account seems to be taken of what the reasoner does not know, an issue already much studied in the area of databases in the context of the closed-world assumption (Reiter, 1978a), in which any atomic formula not explicitly present in the database is intended (or assumed) to be false. Similarly, inheritance hierarchies provide mimicked traits (eg, flying) for subclasses (eg, robins) of other classes (eg, birds) unless made exceptional (eg, ostriches).

It is true that any straightforward attempt to represent such reasoning in terms of sentences in a traditional monotonic logic (in which the stated conclusions are theorems) will fail for the simple reason that these logics will necessarily have the original theorem (Tweety can fly) carried over to the augmented theories by virtue of their monotonicity. Several questions then arise:

1. Are there other formal logics that can represent such reasoning?
2. Has commonsense reasoning been fairly portrayed here or are there other factors involved that might change the assessment of the role of nonmonotonicity?
3. Might not a clever use of monotonic logic allow the effect of nonmonotonic deductions?

Minsky seems to have concluded that formal methods per se are inappropriate to capture such reasoning, whereas others have taken his ideas as a challenge by which to find more powerful formal methods. Out of this challenge has arisen a substantial field of research in nonmonotonic reasoning.

In fact, vigorous efforts have been made toward answering each of the three questions above, and the terrain has by now shown itself to be a rich and varied one involving ideas from divers parts of artificial intelligence, logic, natural language, and philosophy. One theme that seems to have emerged is that a key element in commonsense reasoning dealing with uncertainty (due to the abundance of special conditions defying specification) is self-reference: the reasoning entity uses information about the extent of its own knowledge. Indeed, answers suggested to the above three questions can be viewed in terms of their approach to representing such self-reference. This will be explored in what follows.

The topic of nonmonotonic reasoning has undergone an explosion of new material in recent years, so that it will not be possible to do justice to it in this brief survey. More information is available (Ginsberg, 1987; Reinfrank and co-workers, eds., 1988; Brachman and co-workers, 1989; Genesereth and Nilsson, 1987).

NONMONOTONIC FORMALISMS

Two distinct formalisms emerged around 1980 that attempted to capture the essence of nonmonotonic reasoning by providing a new kind of logical framework. One (McDermott and Doyle, 1980) simply bears the name nonmonotonic logic, and the other (Reiter, 1980) is called default logic. Both employ inferential tools making explicit use of information about what information the formalism itself has available to it. In both cases new syntactic and inferential constructs are developed. Each of these will be discussed in turn.

Nonmonotonic Logic

Nonmonotonic logic (NML) (McDermott and Doyle, 1980) takes as point of departure the desire to represent axiomatically such notions as "If an animal is a bird then, unless proven otherwise, it can fly." To do this a modal operator M is introduced into the language (initially a first-order language) so that if p is a formula then so is Mp (read "p is consistent"). Now in this language it is possible to write formulas that seem to express the kind of reasoning given earlier. For instance, the formula

$$(x)[\mathrm{Bird}(x)\ \&\ M\ \mathrm{Flies}(x)\ .\!\rightarrow \mathrm{Flies}(x)]$$

appears to convey information appropriate to concluding of typical birds that they can fly. A means is needed to characterize deductions with formulas containing the operator M, however, and McDermott and Doyle (1980) went to some length to develop this. As it is essential to their treatment, some time will be spent examining it. To provide an example for the following discussion, let A be the theory {Bird(x) & M Flies(x) → Flies(x), Bird(Tweety)}.

At first blush, it would appear easy to state what is wanted. For if indeed the formula p is consistent (with the rest of the axioms of the particular instance of NML that is to be used) and if in typical situations (ie, ones in which p is consistent) the formula q happens to be true, then a rule such as "from Mp deduce q" seems appropriate. However, McDermott and Doyle chose M to be a part of the language itself, ie, Mp is a formula as well as p. This means that a mechanism is needed to make it possible to prove formulas such as Mp, and this is problematic because proofs of consistency are not only notoriously hard in general but in fact are usually impossible within the same axiomatic system with respect to which consistency is sought. To deal with this problem, McDermott and Doyle extended the notion of proof to allow a kind of consistency test, at the expense of effectiveness. In fact, all formal approaches to nonmonotonic reasoning seem to run into this same issue. Their notion of proof is as follows: If L is a first-order language modified by the addition of a modal operator M, A is a theory in the language L, and S is a set of formulas in the language L of A, let

$$NM_A(S) = Th(A \cup As_A(S))$$

where

$$As_A(S) = \{Mq\text{:q}\in\text{L and } \neg q \notin S\} - Th(A)$$

Here $Th(A)$ is the usual set of first-order consequences of A, and $As_A(S)$, the so-called set of assumptions from S, consists of those formulas Mq not in $Th(A)$ for which $\neg q$ is not in S. Intuitively, an Mq that is not already proven is to be considered an assumption on the basis of S if S does not rule q out, ie, Q is considered to be possible. The idea is to adjoin assumptions to A and find all (usual) consequences, this producing the set $NM_A(S)$. S, of course, could be A itself, or even empty. However, when NM_A is formed, new formulas are thereby available for use (ie, they are considered proven) and these may themselves provide the basis for another round of assumptions. So S plays the role of a recursion variable, and a fixed point of $NM_A(S)$ is sought. It is desired then to consider as theorems precisely those formulas contained in all fixed points S. However, some theories A have no such fixed points, and for these such a definition will not do. McDermott and Doyle settled on the entire language L in such cases, thereby defining the set of theorems nonmonotonically derivable from A as

$$TH(A) = \cap(\{L\} \cup \{S\text{:}NM_A(S)\text{=}S\})$$

In terms of the example theory {Birds(x) & MFlies(x) → Flies(x), Bird(Tweety)}, every fixed point S will contain the sentence Flies(Tweety). Intuitively, because ¬Flies(Tweety) is not initially in the theory, and each stage of generating new assumptions will produce only additional sentences such as MFlies(Tweety) as well as their ordinary consequences (such as Flies(Tweety)), then Flies(Tweety) will be remain present in all iterations of the assumption process. Thus Flies(Tweety) will be a nonmonotonic theorem of A.

Note that any attempt to calculate $TH(A)$ leads to consistency tests. For in iterating $NM_A(S)$ for S initially empty, it is immediately necessary to determine whether, for any given q, Mq is in $Th(A)$. This is in general undecidable and amounts precisely to determining whether $A \cup \{\neg Mq\}$ is inconsistent. McDermott and Doyle acknowledged this difficulty and showed that in very restricted cases, essentially propositional logic, there is a remedy. [They also defined a notion of model for NML; however, there is some dispute as to the completeness of their definition (Davis, 1980).]

McDermott (1982) tried to strengthen NML to overcome certain weaknesses in the original version, in particular the fact that Mp and $\neg p$ are not contradictory. The newer effort makes fuller use of the modal character of the language, but the case he explores most collapses into equivalence with an ordinary monotonic logic. More recent work suggests, however, that in other cases this collapse need not occur (Marek and Truszczynski, 1989b).

A Distinction: Semantic Approaches

Moore (1983) reexamined the underlying goals of NML and concluded that two ideas were being conflated: typicality on the one hand and beliefs about beliefs on the other. He distinguished between concluding Tweety can fly on the basis that it is not known that Tweety cannot fly and that typically birds can fly, and concluding Tweety can fly on the basis that it is not known that Tweety cannot fly and that "I would know it if Tweety could not fly." Moore argued that the former is intended to be approximate and error prone, whereas the latter (which he called autoepistemic reasoning) is intended to be sound. He devised a logical semantics (usually denoted AEL) for the latter form of reasoning.

It does appear that autoepistemic reasoning forms a part of commonsense reasoning. The example above is not as striking as one given by Moore: "I would know it if I had an elder brother." Here he is presumably not merely stating a belief about typicality (that people typically know their older brothers, although that seems true enough) but rather a belief that "I" specifically do know of all "my" brothers. Admittedly this is arguable, because situations exist in which an older brother may be unknown, but they are not likely to be taken seriously, so that again a kind of typicality may be present here.

Moore pointed out that in autoepistemic beliefs there is a possibility of failure, ie, the belief can be false (I may have an elder brother after all) in which case I must alter that belief, whereas in the case of typicality I may merely conclude that I am atypical regarding knowledge of brothers and yet preserve the belief that typically elder brothers are known. Still, if I do discover to my surprise that such a brother exists, it would seem likely that I would conclude immediately that I was wrong about my autoepistemic belief but that the belief still applies to most people, ie, there seems a very fine and tenuous line between the two forms of beliefs. It seems possible to move back and forth between explicit typicality beliefs in which uncertainty is acknowledged and more stubborn autoepistemic ones, for the same assertions, depending on context, and the willingness of people to alter their position when challenged may attest to an implicit default character even in autoepistemic cases.

It is of interest that both forms of reasoning, however, like all nonmonotonic formalisms, depend at least implicitly on a determination that in fact certain formulas are not theorems of the formalism in question. Note that in Moore's example it must somehow be determined that in fact an elder brother is not known before using the autoepistemic belief and *modus ponens* to conclude there is no such brother. Again, this self-referential or consistency aspect of the reasoning seems the most striking characteristic, and the one presenting the greatest formal difficulty. Related approaches have been published (Halpern and Moses, 1984; Shoham, 1988; Bell, 1990).

Default Logic

Reiter (1980) introduced a logic for default reasoning (DL). In specifically singling out default reasoning, Reiter identified his concern as that of studying typicality rather than other possible nonmonotonic forms of reasoning. His formalism in fact bears close resemblance to NML, the most obvious difference being that the language is strictly first order, with the operator M playing a role only in rules of inference rather than in axioms. Specifically, Reiter allowed inference rules (default rules) such as

$$\frac{\mathrm{Bird}(x)\colon M\mathrm{Flies}(x)}{\mathrm{Flies}(x)}$$

where MFlies(x) is intended not as an antecedent theorem to the consequent Flies(x) but instead as a condition that must be met before Flies(x) can be concluded from Bird(x). The condition is, roughly (and as in all nonmonotonic formalisms) that Flies(x) be consistent with the rest of the axiomatic framework. Thus if the above rule and the axiom Bird(Tweety) are employed, the conclusion Flies(Tweety) results. As with NML, formalizing the notion of consistency for the indicated purpose requires care. Making this precise and showing it to be useful is the bulk of the task Reiter undertook. He employed a hierarchy of iterations along lines similar to that of NML, also arriving at a fixed point, in determining a notion of proof for default rules.

Reiter and Criscuolo (1981) also considered what they called interacting defaults, ie, default rules that separately might lead to opposed conclusions, such as in "Richard Nixon is a Quaker and a Republican" where it is known, say, that typically Quakers are pacifists and Republicans are not. This appears to be a substantial difficulty for any form of nonmonotonic reasoning that pretends to deal with typicality. Along the lines of interacting defaults, yet another approach has gotten much attention: inheritance hierarchies (Horty, 1990; Horty and co-workers, 1987; Selman and Levesque, 1989; Touretzky, 1984a, 1984b; Touretzky and co-workers, 1987; Etherington and Reiter, 1983).

REMAINING WITHIN FIRST-ORDER LOGIC

McCarthy (1980) has devised an ingenious means for representing and calculating knowledge about situations involving minimization of particular notions. He called this technique circumscription (which we will denote CL). It is noteworthy in the present context because it seems able to handle many of the kinds of reasoning with self-reference found in nonmonotonic approaches and yet stays within first-order logic. McCarthy managed this by use of an axiom schema that partially captures the notion of a model of a set of sentences, similar to (and in fact generalizing) the familiar manner of defining the natural numbers by a minimizing schema applied to the successor operation. Much work has followed his original paper. In particular, by introducing a predicate for abnormal McCarthy (1986) has been able to capture some of the intuitions about reasoning about typicality, that is, circumscribing that predicate can lead to conclusions to the effect that Tweety can fly, because it is abnormal for a bird not to fly and because abnormality is (intended to be) minimized by circumscription. Some positive and negative results on this have been published (Etherington and co-workers, 1985; Perlis and Minker, 1986).

PROBLEMATIC ASPECTS UNDERLYING NONMONOTONIC REASONING

Kowalski (1979) and Israel (1980) suggested that something was missing from Minsky's account of commonsense reasoning under uncertainty: the reasoning entity creating the "proof" (say that Tweety flies) must know that it does not know certain facts (such as that Tweety is an ostrich), and furthermore, that when this knowledge of self is properly represented, the reasoning is no longer nonmonotonic, thereby rendering unnecessary the development of new (nonmonotonic) logics. Their argument is as follows: a default rule such as "if X is not known then Y" (eg, if Tweety doesn't fly is not known then Tweety flies) is at least implicit in nonmonotonic proofs, and so the reasoning must make use, in some fashion, of X not being known, before concluding Y. This means there must be an additional mechanism M to determine that in fact X is not known. But then if the system is augmented by coming to know X (Tweety is an ostrich and, therefore, cannot fly, say) then the system can no longer derive "X is not known" as long as the mechanism M for such derivations is faithful to the facts. That is, not only has the system been augmented by now knowing X, it has had an old piece of knowledge removed (and properly so, for it no longer is true), namely that "X is not known," and that piece of information was precisely what previously allowed the now inadmissable conclusion Y.

What has happened in such a scenario is that one logic has been replaced by another that contains additional information but also has lost some information (namely information that no longer is true because of the very presence of the new information). In effect, a reasoning system that is to know about its own reasoning would appear to require temporal changes reflecting the fact that its previous states obeyed different truths. If a system first does not know X and then later does, it was true at first that "X is not known" and later this is false. If the system itself is to have this knowledge represented (as Kowalski and Israel argued) then Minsky's argument for nonmonotonicity fails, because there is no longer a strict augmentation of the original axioms. Israel, in particular, argued that a sequence of logics is a better way to view the situation, in which axioms are constantly added and subtracted as new facts become known and that this process is not one of deduction but of interaction with the happenstance environment. To a certain extent this acknowledges Minsky's point that logic is not (entirely) what is involved here. However, the insight provided by making explicit the default knowledge and mechanism M that utilizes it suggests that logic still may do all, and that other processes invoke the necessary self-referential inspections to determine whether or not X is still known. In fact, just such an approach has been undertaken in experimental reasoning systems (Perlis, 1984; Elgot-Drapkin, 1988).

Israel also argued that a sufficiently perceptive agent that uses nonmonotonic reasoning will necessarily entertain (at least occasional) inconsistencies. Perlis (1987) formalized this into specific challenges for the three formalisms DL, NML, and CL. In particular, the presence of beliefs to the effect that some of the nonmonotonic conclusions are false can lead to inconsistency or to the blocking of desired default conclusions. This has been investigated and partial solutions have been found (Etherington and co-workers, in press).

The issue of determining "what is not known" to a reasoner, as indicated above, appears to be central to all current formalizations of nonmonotonic reasoning. This is subject to computability constraints (undecidability in the general case) but also others. For instance, Perlis (1987) has shown that if a first-order reasoner is able to introspect both positively and negatively (Known(P) is inferred whenever P is itself known, and not-Known(P) is inferred whenever P is not known) and also has even fairly mundane arithmetical knowledge, then the reasoner is inconsistent. This also obtains if *known* is treated as a modality.

INTERCONNECTIONS

Given that there are three or four principal formalisms already in the literature aimed at capturing the informal notion of nonmonotonic reasoning, it is natural to ask about comparisons between them. Some results have been established along these lines, relating AEL, DL, circumscription, and logic programming (Reiter, 1982; Konolige, 1989, 1987; Marek and Subrahmanian, in press; Marek and Truszczynski, 1989a, in press; Przymusinski, 1989).

APPLICATIONS AND RELATED WORK

As with much of commonsense reasoning techniques, formal nonmonotonic modes of reasoning naturally present themselves as candidates for a reasoning mechanism that could in principle be used in an intelligent robot, for instance, in conjunction with a theorem prover. So far, little has been done in a concrete way to address these issues. Some preliminary work has been presented (Perlis, 1984).

One rather specific application of nonmonotonic reasoning that has received much attention is temporal persistence in the presence of conflicting defaults. This problem was most vividly noted in connection with the Yale Shooting Problem (Hanks and McDermott, 1986). Here each of two states of affairs (a person's being alive and a gun's being loaded) alone tends to persist. But together it is possible to negate the other (if at some point the gun is aimed at the person and the trigger is pulled). Has the gun remained loaded up to that point? Has the person remained alive up to that point? What general principles lead to intuitive conclusions here? Hanks and McDermott showed that the problem is nontrivial, indeed, they further argued that the problem illustrates a fundamental inappropriateness of any formal techniques to commonsense reasoning, thereby aligning themselves with Minsky's original position. But, not surprisingly, many proposals were forthcoming that provided various formalisms designed to solve this sort of problem. A survey of this entire area has been published (Haugh, 1989).

BIBLIOGRAPHY

J. Bell, "The Logic of Non-Monotonicity," *Artif. Intell.* **41,** 365–374 (1990).

R. Brachman, H. Levesque, and R. Reiter, eds., *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, 1989.

M. Davis, "The Mathematics of Non-Monotonic Reasoning," *Artif. Intell.* **13**(1–2), 73–80 (1980).

J. Elgot-Drapkin, *Step-Logic: Reasoning Situated in Time*, Ph.D. dissertation, University of Maryland, College Park, 1988.

D. Etherington, S. Kraus, and D. Perlis, "Limited Scope and Circumscriptive Reasoning," in K. Ford and P. Hayes, eds., *Advances in Human and Machine Cognition*, Vol. 1, JAI Press, in press.

D. Etherington, R. Mercer, and R. Reiter, "On the Adequacy of Predicate Circumscription for Closed-World Reasoning," *J. Comput. Intell.* **1,** 11–15 (1985).

M. Genesereth and N. Nilsson, *Logical Foundations of Artificial Intelligence*, Morgan-Kaufmann, San Mateo, Calif., 1987.

M. Ginsberg, *Readings in Non-Monotonic Reasoning*, Morgan-Kaufmann, San Mateo, Calif, 1987.

J. Halpern and Y. Moses, "Towards a Theory of Knowledge and Ignorance: Preliminary Report in *Proceedings of the Workshop on Nonmonotonic Reasoning*, New Paltz, N.Y., 1984, pp. 125–143.

S. Hanks and D. McDermott, "Default Reasoning, Nonmonotonic Logics and the Frame Problem," in *Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, AAAI, Menlo Park, Calif., 1986, pp. 328–333.

B. Haugh, *Nonmonotonic Formalisms for Commonsense Temporal Causal Reasoning*, Ph.D. dissertation, University of Maryland, College Park, 1989.

J. Horty, *Some Direct Theories of Nonmonotonic Inheritance*, Technical Report, University of Maryland Institute for Advanced Computer Studies, College Park, 1990.

J. Horty, R. Thomason, and D. Touretzky, "A Skeptical Theory of Inheritance in Nonmonotonic Semantic Networks, in *Proceedings of the Sixth National Conference of Artificial Intelligence*, Seattle, Wash., AAAI, Menlo Park, Calif., 1987, pp. 358–363.

D. Israel, "What's Wrong with Non-Monotonic Logic," in *Proceedings of the First National Conference on Artificial Intelligence*," Stanford, Calif., AAAI, Menlo Park, Calif., 1980, pp. 99–101.

K. Konolige, "On the Relation between Default Theories and Autoepistemic Logic," in *Proceedings of the Tenth IJCAI*, Milan, Italy, Morgan-Kaufmann, San Mateo, Calif., 1987.

K. Konolige, "On the Relation between Autoepistemic Logic and Circumscription," in *Proceedings of the Eleventh IJCAI*, Detroit, Mich., Morgan-Kaufmann, San Mateo, Calif., 1989.

R. Kowalski, *Logic for Problem Solving*, North-Holland, Amsterdam, The Netherlands, 1979.

J. McCarthy, "Circumscription—A Form of Non-Monotonic Reasoning," *Artif. Intell.* **13**(1–2), 27–39 (1980).

J. McCarthy, "Applications of Circumscription to Formalizing Commonsense Knowledge," *Artif. Intell.* **28,** 89–116 (1986).

D. McDermott, "Nonmonotonic Logic II: Non-Monotonic Modal Theories," *JACM,* **29**(1), 33–57 (1982).

D. McDermott and J. Doyle, "Non-Monotonic Logic I," *Artif. Intell.* **13**(1–2), 41–72 (1980).

W. Marek and V. S. Subrahmanian, "The Relationship between Stable, Supported, Default and Auto-Epistemic Semantics for General Logic Problems," *Theor. Comput. Sci.*, in press.

W. Marek and M. Truszczynski, "Relating Autoepistemic and Default Logics," in R. Brachman, H. Levesque, and R. Reiter, *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, 1989a, pp. 276–288.

W. Marek and M. Truszczynski, "Stable Semantics for Logic Programming and Default Logic," in *Proceedings of the North American Conference on Logic Programming*, Cleveland, Ohio, 1989b.

W. Marek and M. Truszczynski, "Autoepistemic Logic," *JACM*, in press.

M. Minsky, "A Frameword for Representing Knowledge," in P. Winston, ed., *The Psychology of Computer Vision*, McGraw-Hill Book Co., Inc., New York, 1975.

R. Moore, "Semantical Considerations on Non-Monotonic Logic," in *Proceedings of the Eighth IJCAI*, Karlsruhe, FRG, Morgan-Kaufmann, San Mateo, Calif., 1983, pp. 272–279.

D. Perlis, "Non-Monotonicity and Real-Time Reasoning," in Halpern and Moses, 1984.

D. Perlis, "On the Consistency of Commonsense Reasoning," *Comput. Intell.* **2,** 180–190 (1987).

D. Perlis and J. Minker, "Completeness Results for Circumscription," *Artif. Intell.* **28,** 29–42 (1986).

T. Przymusinski, "Three-Valued Formalizations of Non-Monotonic Reasoning and Logic Programming," in Brachman and co-workers, 1989, pp. 341–348.

M. Reinfrank, J. de Kleer, M. Ginsberg, and E. Sandewall, eds., *Non-Monotonic Reasoning, Proceedings of the Second International Workshop*, Grassau, Springer-Verlag, New York, 1988.

R. Reiter, "A Logic for Default Reasoning," *Artif. Intell.* **13**(1–2), 81–132 (1980).

R. Reiter, "On Closed World Databases," in H. Gallaire and J. Minker, eds., *Logic and Databases*, Plenum, New York, 1987a, pp. 55–76.

R. Reiter, "On Reasoning by Default," in the *Proceedings of the Second TINLAP*, Urbana, Ill., 1987b.

R. Reiter and G. Criscuolo, "On Interacting Defaults," in *Proceedings of the Seventh IJCAI*, Vancouver, B.C., Morgan-Kaufmann, San Mateo, Calif., 1981.

B. Selman and H. Levesque, "The Tractability of Path-Based Inheritance," in *Proceedings of the Eleventh IJCAI*, 1989.

Y. Shoham, *Reasoning about Change*, MIT Press, Cambridge, Mass., 1988.

D. Touretzky, "Implicit Ordering of Defaults in Inheritance Systems," in *Proceedings of the Fourth National Conference on Artificial Intelligence*, Austin, Tex., Morgan-Kaufmann, San Mateo, Calif., 1984a.

D. Touretzky, *The Mathematics of Inheritance Systems*, Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, Pa., 1984b.

D. Touretzky, H. Horty, and R. A. Thomason, "A Clash of Intuitions: The Current State of Non-Monotonic Multiple Inheritance Systems," in *Proceedings of the Tenth IJCAI*, 1987.

General References

K. Clark, "Negation as Failure," in H. Gallaire and J. Minker, eds., *Logic and Databases*, Plenum Press, New York, 1978, pp. 293–322.

J. Doyle, "A Truth Maintenance System," *Artif. Intell.* **12,** 231–272 (1979).

D. Etherington, *Reasoning with Incomplete Information*, Morgan-Kaufmann, San Mateo, Calif., 1988.

B. Grosof, "Default Reasoning as Circumspection," in *Proceedings of the Workshop on Nonmonotonic Reasoning*, New Paltz, N.Y., 1984.

K. Konolige, "Circumscriptive Ignorance," in *Proceedings of the Second International Conference on Artificial Intelligence*, Pittsburgh, Pa., AAAI, Menlo Park, Calif., 1982, pp. 202–204.

K. Konolige, *Belief and Incompleteness*, SRI Technical Note 319, SRI International, Menlo Park, Calif., 1984.

I. Kramosil, "A Note on Deduction Rules with Negative Premises," in *Proceedings of the Fourth IJCAI*, Tbilisi, USSR, Morgan-Kaufmann, San Mateo, Calif., 1975, pp. 53–56.

S. Kraus, D. Lehmann, and M. Magidor, "Nonmonotonic Reasoning, Preferential Models and Cumulative Logics," *AIJ*, in press.

D. Kueker, "Another Failure of Completeness for Circumspection," paper presented at Week on Logic and Artificial Intelligence, University of Maryland, College Park, 1984.

H. Levesque, "Incompleteness in Knowledge Bases," *SIGART Newslett.* **74,** 150 (1981a).

H. Levesque, "The Interaction with Incomplete Knowledge Bases: A Formal Treatment," in *Proceedings of the Seventh IJCAI*, Vancouver, B.C., Morgan-Kaufmann, San Mateo, Calif., 1981b.

H. Levesque, *A Formal Treatment of Incomplete Knowledge Bases*, Ph.D. dissertation, University of Toronto, Toronto, Canada, 1981.

V. Lifschitz, "Some Results on Circumscription," in *Proceedings of the Workshop on Nonmonotonic Reasoning*, New Paltz, N.Y., 1984.

W. Lipski, "On the Logic of Incomplete Information," in *Lecture Notes in Computer Science*, Vol. 53, Springer-Verlag, New York, 1977, pp. 374–381.

W. Lukaszewicz, "General Approach to Nonmonotonic Logics," in *Proceedings of the Eighth IJCAI*, Karlsruhe, FRG, Morgan-Kaufmann, San Mateo, Calif., 1983, pp. 352–354.

J. McCarthy, "Applications of Circumscription to Formalizing Common Sense Knowledge," in *Proceedings of the Workshop on Nonmonotonic Reasoning*," New Paltz, N.Y., 1984.

J. Minker, "On Indefinite Databases and the Closed-World Assumption," in *Lecture Notes in Computer Science*, Vol. 138, Springer-Verlag, New York, 1982, pp. 292–308.

J. Minker and D. Perlis, "Protracted Circumscription," in *Proceedings of the Workshop on Nonmonotonic Reasoning*, New Paltz, N.Y., 1984a.

J. Minker and D. Perlis, "Applications of Protected Circumscription," in *Lecture Notes in Computer Science*, Vol. 170, Springer-Verlag, New York, 1984b, pp. 414–425.

R. Moore, *Reasoning from Incomplete Knowledge in Procedural Deduction System*, Memo **347,** MIT Artificial Intelligence Laboratory, Cambridge, Mass., 1975.

D. Nute, "Conditional Logic," in D. M. Gabbay and F. Guenthner, eds., *Handbook of Philosophical Logic*, Reidel, Dordrecht, 1984, pp. 387–439.

J. Nutter, *Default Reasoning in AI Systems*, M.Sc. thesis, SUNY at Buffalo, 1983a.

J. Nutter, "What Else Is Wrong with Nonmonotonic Logics? Representational and Informational Shortcomings," in *Proceedings of the Fifth Cognitive Science Conference*, Rochester, N.Y., 1983b.

M. A. Papalaskaris and A. Bundy, "Topics for Circumscription," in *Proceedings of the Workshop on Nonmonotonic Reasoning*, New Paltz, N.Y., 1984.

D. Perlis, "Languages with Self-Reference II: Knowledge, Belief, and Modality," *Artif. Intell.* **34,** 179–212 (1988).

J. Pollack, "A Refined Theory of Counterfactuals," *J. Philosph. Logic* **10,** 239–266 (1981).

R. Reiter, "Equality and Domain Closure in First-Order Databases," *JACM* **27**(2), 235–249 (1980).

R. Reiter and G. Criscuolo, "Some Representational Issues in Default Reasoning," *J. Comput. and Maths. with Applications,* (special issue on computational linguistics) **9,** 1–13 (1983).

E. Rich, "Default Reasoning as Likelihood Reasoning," in *Proceedings of the Third National Conference on Artificial Intelligence*, Washington, D.C., AAAI, Menlo Park, Calif., 1983.

E. Sandewall, *Partial Models, Attribute Propagation Systems, and Non-Monotonic Semantics*, LITH-IDA-R-83-01, Linkoping University, Linkoping, Sweden.

R. Stalnaker, *A Note on Non-Monotonic Modal Logic*, Department of Philosophy, Cornell University, Ithaca, N.Y., 1980.

D. Perlis
University of Maryland

REASONING, PLAUSIBLE

In the past, the management of uncertainty in expert systems (qv) has usually been left to *ad hoc* representations and combining rules lacking either a sound theory or clear semantics. However, the aggregation of uncertain information (facts) is a recurrent need in the reasoning process of an expert system. Facts must be aggregated to determine the degree to which the premise of a given rule has been satisfied, to verify the extent to which external constraints have been met, to propagate the amount of uncertainty through the triggering of a given rule, to summarize the findings provided by various rules or knowledge sources or experts, to detect possible inconsistencies among the various sources, and to rank different alternatives or different goals.

COPING WITH UNCERTAINTY IN EXPERT SYSTEMS

Over the past few years, uncertainty management has received a vast amount of attention from the researchers in the field, leading to the establishment of two well-defined approaches based on probability and possibility theory, respectively. In this article these approaches will be illustrated and compared.

Sources of Uncertainty

In a survey of reasoning with uncertainty (Bonissone and Tong, 1985), it was noted that there are two major types of uncertainty: randomness and fuzziness. Randomness deals with the uncertainty of whether a given element belongs or does not belong to a well-defined set (event). Fuzziness deals with the uncertainty derived from the partial membership of a given element to a set whose boundaries are not sharply defined.

These two types of uncertainty can be introduced in reasoning systems by a variety of sources: the reliability of the information, the inherent imprecision of the representation language in which the information is conveyed, the incompleteness of the information, and the aggregation or summarization of information from multiple sources.

The first source type is related to the reliability of information: uncertainty can be present in the factual knowledge (ie, the set of assertions or facts) due to inaccuracy and poor reliability of the instruments used to make the observations. Uncertainty can also occur in the knowledge base (ie, the rule set) as a result of using weak implications. Unlike categorical rules (describing set subsumption relationships) weak implications or plausible rules are typically used to describe likely interpretations of situations. By their very nature, these rules are less reliable than categorical rules and are used when the expert or model builder is unable to establish an exact correlation between premise and conclusion. In most expert systems the degree of implication is expressed as a scalar value on an interval (certainty factor, conditional probability, degree of sufficiency, etc). This value represents the change from the strict implication for all x, $A(x) \rightarrow B(x)$, to the weaker statement for most x, or usually, for all x, $A(x) \rightarrow B(x)$. The latter statement is not categorical and allows the possibility of exceptions to the rule. Thus the logical implication has now been changed into a plausible implication or disposition (Zadeh, 1985a, 1988). A natural way to express such a degree of implication is achieved by using fuzzy quantifiers such as *most, almost all,* etc (Zadeh, 1983a, 1984a). A fuzzy quantifier is a fuzzy number representing the relative cardinality of the subset of elements in the universe of discourse that usually satisfy the given property, ie, the implication. Uncertainty in the data can be compounded by aggregating uncertain data in the premise, by propagating certainty measures to the conclusion, and by consolidating the final certainty measure of conclusions derived from different rules. Triangular norms and conorms (Schweizer and Sklar, 1963; Dubois and Prade, 1984) can be used to generalize the conjunction and disjunction operators that provide the required aggregation capabilities. A description of their characteristics is provided under "Triangular Norm Based Reasoning Systems," below.

The second type of uncertainty is caused by the inherent imprecision of the facts and rules representation language. Observations can contain ill-defined concepts. Rules can contain vague predicates describing tests that cannot be expressed by Boolean expressions (eg, a great change in heading). As a result, these rules cannot be interpreted exactly. This problem has been partially addressed by the possibilistic theory of approximate reasoning that, in light of imprecise fact and rule descriptions, allows weaker inferences to be made based on a generalized *modus ponens* (Zadeh, 1975).

The third type of uncertainty is caused by the incompleteness of the information. This type of uncertainty has generally been modeled by nonnumerical characterizations, such as Doyle's (1983) reasoned assumptions.

The fourth type of uncertainty arises from the aggregation of information from different knowledge sources or experts. When unconditional statements (facts) are aggregated, three potential problems can occur: the closure of the representation may no longer be preserved when the facts to be aggregated have different granularity (the single-valued certainty measures of the facts may change into an interval-valued certainty measure of the aggregated fact), the aggregation of conflicting statements may generate a contradiction that should be detected, and the rule of evidence combination may create an overestimated certainty measure of the aggregated fact, if a normalization is used to eliminate or hide a contradiction (Zadeh, 1984b, 1985b). The first two problems are typical of single-valued numerical approaches, whereas the last problem is found in the two-valued approach (Dempster, 1967).

All these approaches will be discussed in the following section. The state of the art of techniques for reasoning with uncertainty will be reviewed. The numerical approaches will be emphasized, and probabilistic and possibilistic methods will be compared and evaluated against a list of requirements.

State of the Art of Reasoning with Uncertainty

The existing approaches to representing uncertainty can be subdivided into two basic categories according to their quantitative or qualitative characterizations of uncertainty.

Among the quantitative approaches, are two types of reasoning that differ in the semantics of their numerical representation. One is the probabilistic reasoning approach, based on probability theory. The other one is the possibilistic reasoning approach, based on the semantics of many-valued logics. Some of the more traditional techniques found among the approaches derived from probability are based on single-valued representations. These techniques include Bayes rule (Peark, 1982, 1985, 1988a), modified Bayesian rule (Duda and co-workers, 1976), and confirmation theory (Shortliffe and Buchanan, 1975). A more recent trend among the probabilistic approaches is represented by approaches based on interval-valued representations such as Dempster (1967) and Shafer (1976) theory; evidential reasoning (Lowrance and co-workers, 1986); probability bounds, ie, consistency and plausibility (Quinlan, 1983); and evidence space (Rollinger, 1983).

Over the last five years, considerable efforts have been devoted to improve the computational efficiency of Bayesian belief networks for trees and small polytrees (Pearl, 1988b) and for directed acyclic graphs (influence diagrams) (Howard and Matheson, 1984; Schachter, 1986; Agogino and Rege, 1987). Problem decomposition techniques (eg, loopcuts and cliques) (Lauritzen and Spiegelhalter, 1988) and approximate methods (eg, conditioning, clustering, bounding interval, and simulations) (Henrion, 1989) have been derived to handle multiconnected Bayesian belief networks (Pearl, 1988b).

Among the approaches anchored on many-valued logics, the most notable are based on a fuzzy-valued representation of uncertainty. These include necessity and possibility theory (Zadeh, 1978, 1979a), the linguistic variable approach (Zadeh, 1979b, 1983b), and the triangular-norm based approach (Bonissone, 1987a, 1990; Bonissone and Decker, 1986; Bonissone and co-workers, 1987).

With numerical representations, it is possible to define a calculus that provides a mechanism for propagating uncertainty through the reasoning process. Similarly, the use of aggregation operators provides summaries that can then be ranked to perform rational decisions. Such a numerical representation, however, cannot provide a clear explanation of the reasons that led to a given conclusion. The typical available explanations are usually annotated traces of the reasoning paths followed by the inference engine.

Models based on qualitative approaches, on the other hand, are usually designed to handle the aspect of uncertainty derived from the incompleteness of the information, such as reasoned assumptions (Doyle, 1983) and default reasoning (Reiter, 1980). With a few exceptions, they are generally inadequate to handle the case of imprecise information, as they lack any measure to quantify confidence levels (Doyle, 1983). A few approaches in this group have addressed the representation of uncertainty, using either a formal representation, such as knowledge and belief (Halpern and Moses, 1985), or a heuristic representation, such as the theory of endorsements (Cohen, 1985; Cohen and Grinberg, 1983a).

The formal approach has a corresponding (modal) logic theory that determines the mechanism by which inferences (theorems) can be proven or believed to be true. The heuristic approach has a set of context-dependent rules to define the ways by which framelike structures (endorsements) can be combined, added, or removed. The symbolic representations are more suitable for providing a trace from the sources of the information through the various inference paths to the final conclusions. However, no calculus can be defined for the propagation, aggregation, and ranking of such uncertain information. The only available partial solution is the use of context-dependent rules to determine how each piece of evidence can be compared or summarized.

In this article, the qualitative approaches will be briefly covered. However, most of the discussion will be focused on describing and comparing quantitative approaches. In particular the probabilistic and possibilistic reasoning systems will be analyzed.

APPROXIMATE REASONING SYSTEMS

Reasoning systems must attach a truth value to statements about the state or the behavior of a real world system. When this hypothesis evaluation is not possible due to the lack of complete and certain information, approximate reasoning techniques are used to determine a set of possibilities (possible worlds) that are logically consistent with the available information. These possible worlds are characterized by a set of propositional variables and their associated values. Because it is generally impractical to describe these possible worlds to an acceptable level of detail, approximate reasoning techniques seek to determine some properties of the set of possible solutions or some constraints on the values of such properties (Ruspini, 1987, 1989a, 1989b).

A large number of approximate reasoning techniques have been developed over the past decade to provide these solutions, and a survey has been published (Pearl, 1988a). The similarities and differences between the two most common approximate reasoning techniques, probabilistic and possibilistic reasoning, will be highlighted.

Probabilistic Reasoning

Probability-based, or probabilistic, reasoning seeks to describe the constraints on the variables that characterize the possible worlds by indentifying their conditional probability distributions given the evidence in hand. Its supporting formalisms are based on the concept of set measures, additive real functions defined over certain subsets of some space. Probabilistic methods seldom make categorical assertions about the actual state of the system being investigated. Rather, they indicate that there is an experimentally determined (or believed) tendency or propensity for the system to be in some specified state. Thus they are oriented primarily toward decisions that are optimal in the long run, describing the tendency or propensity of truth of a proposition without assuring its actual validity. Depending on the nature of the information, probabilistic reasoning estimates the frequency of the truth of a hypothesis as determined by prior observation (objectivist interpretation) or a degree of gamble based on the actual truth of the hypothesis (subjectivist interpretation).

From a practical computational viewpoint, probabilistic methods suffer from problems associated with the reliable determination of all required joint and conditional probabilities. In complex systems, many variables interrelate in ways that cannot be expressed in terms of simpler interactions. In these cases, the complexity of probabilistic inference is exponential in the size of the largest subgraph into which the system can be decomposed.

Possibilistic Reasoning

Conversely, possibilistic reasoning, which is rooted in fuzzy set theory (Zadeh, 1965) and many-valued logics, seeks to describe the constraints on the values of the variables of the possible worlds in terms of their similarity to other sets of possible worlds. The supporting formalisms are based on the mathematical concept of metrics instead of set measure. These methods focus on single situations and cases. Rather than measuring the tendency of the given proposition to be valid, they seek to find another related, similar proposition that is valid. This proposition is usually less specific and resembles (according to some measure of similarity) the original hypothesis of interest.

The notion of similarity is based on the concept of metric or distance. Distances are functions that assign a number greater than zero to pairs of elements of some set (for sake of simplicity, it will be assumed that the range of this function is the interval [0,1]). Distances are reflexive, commutative, and transitive. Similarity can be defined as the complement of distance, ie,

$$S(A,B) = 1 - d(A,B) \tag{1}$$

The basic structural characteristics of the similarity functions is an extended notion of transitivity that allows the computation of bounds on the similarity between two objects A and B on the basis of knowledge of their similarities to a third object C:

$$S(A,B) \geq T(S(A,C),S(B,C)) \tag{2}$$

where T is a triangular norm (Bonissone and Decker, 1986; Bonissone, 1987a). Any continuous triangular norm $T(A,B)$ falls in the interval $\text{Max}(0,A + B - 1) \leq T(A,B) \leq \text{Min}(A,B)$. Thus it can be observed that if the lower bound of the range of T norms is used in the expression describing the transitivity of similarity (eq. 2), the triangular inequality for distances is obtained. If the upper bound is used, the ultrametric inequality is obtained.

This similarity notion is a direct extension of the notion of accessibility relation that is of fundamental importance in modal logics. This notion is further described by Ruspini (1990). In summarizing Ruspini's results, it can be observed that the notion of accessibility captures the idea that whatever is true in some world w, is true, but in a modified sense, in another w' that is accessible from it. When considering multiple levels of accessibility (indexed by a number between 0 and 1), this relation, measuring the resemblance between two worlds, may be used to express the extent by which considerations applicable in one world may be extended to another world.

The basic inferential mechanism, underlying the generalized *modus ponens* (Zadeh, 1979b), makes use of inferential chains and the properties of a similarity function to relate the state of affairs in the two worlds that are at the extremes of an inferential chain.

Given the duality of purpose and characteristics between probabilistic and possibilistic methods, it can be concluded that these technologies ought to be regarded as being complementary rather than competitive.

PROBABILISTIC APPROACHES

Having contrasted probabilistic and possibilistic reasoning techniques, selected representative approaches will now be examined. Among the probabilistic techniques to be analyzed are the Bayesian approaches (Bayesian, modified Bayesian, and Bayesian belief networks), confirmation theory (certainty factors), and the Dempster-Shafer (belief) theory.

Bayes Rule

Given a set of hypotheses $H = \{h_1, h_2, \ldots, h_n\}$ and a sequence of pieces of evidence $\{e_1, e_2, \ldots, e_m\}$, Bayes rule (see BAYESIAN INFERENCE METHODS), derived from the formula of conditional probability, states that the posterior probability $P(h_i \mid e_1, e_2, \ldots, e_m)$ can be derived as a function of the conditional probabilities $P(e_1, e_2, \ldots, e_m | h_i)$ and the prior probability $P(h_i)$:

$$P(h_i| \, e_1, e_2, \ldots, e_m) = \frac{P(e_1, e_2, \ldots, e_m | h_i)P(h_i)}{\sum_{i=1}^{n} P(e_1, e_2, \ldots, e_m | h_i)P(h_i)} \tag{3}$$

The Bayesian approach is based on two fundamental assumptions. Each hypothesis h_i is mutually exclusive with any other hypothesis in the set **H** and the set of hypotheses **H** is exhaustive, ie,

$$P(h_i,h_j) = 0 \quad \text{for } i \neq j \tag{4}$$

$$\sum_{i=1}^{n} P(h_i) = 1 \tag{5}$$

Second, each piece of evidence e_j is conditionally independent under each hypothesis, ie,

$$P(e_1, e_2, \ldots, e_m | h_i) = \prod_{j=1}^{m} P(e_j | h_i) \tag{6}$$

Note that equations 4 and 5 are required to derive Bayes rule from the formula of conditional probability. Equation 6, on the other hand, is an assumption usually made to alleviate the difficulty of determining the conditional joint probability required by equation 3. Thus under equation 6, equation 3 becomes computationally feasible.

This method requires a large amount of data to determine the estimates for the prior and conditional probabilities. Such a requirement becomes manageable when the problem can be represented as a sparse Bayesian network that is formed by a hierarchy of small cluster of nodes. In this case the dependencies among variables (nodes in the network) are known and only the explicitly required conditional probabilities must be obtained (Pearl, 1985).

Modified Bayes Rule

In addition to equations 4 and 5 (for derivational needs) and equation 6 (for operational convenience) needed by the original Bayes rule, the modified Bayesian approach, used in PROSPECTOR, also requires that each piece of evidence e_j be conditionally independent under the negation of each hypothesis, ie,

$$P(e_1, e_2, \ldots, e_m | \neg h_i) = \prod_{j=1}^{m} P(e_j | \neg h_i) \tag{7}$$

The modified Bayesian approach is based on a variation of the odds–likelihood formulation of Bayes rule. When all the pieces of evidence are certainly true, this formulation defines the posterior odds as:

$$O(h_i|e_1, e_2, \ . \ . \ . \ , e_m) = \frac{P(e_1|h_i)}{P(e_1|\neg h_i)}\frac{P(e_2|h_i)}{P(e_2|\neg h_i)} \ . \ . \ . \ \frac{P(e_n|h_i)}{P(e_n|\neg h_i)}\frac{P(h_i)}{P(\neg h_i)} \quad (8)$$

$$= \lambda_{1,i}\lambda_{2,i}O(h_i)$$

where

$$\lambda j,i = \frac{P(e_j|h_i)}{P(e_j|\neg h_i)}$$

is the likelihood ratio of e_j for hypothesis h_i and

$$O(h_i) = \frac{P(h_i)}{P(\neg h_i)}$$

is the odds on hypothesis h_i. An analogous odds–likelihood formulation is derived for the case when all the pieces of evidence are certainly false:

$$O(h_i|\neg e_1, \neg e_2, \ . \ . \ . \ , \neg e_m) = \frac{P(\neg e_1|h_i)}{P(\neg e_1|\neg h_i)}\frac{P(\neg e_2|h_i)}{P(\neg e_2|\neg h_i)} \ . \ . \ . \ \frac{P(\neg e_n|h_i)}{P(\neg e_n|\neg h_i)}\frac{P(h_i)}{P(\neg h_i)} \quad (9)$$

$$= \lambda^*_{1,i}\lambda^*_{2,i} \ . \ . \ . \ \lambda^*_{n,i}O(h_i)$$

The likelihood ratio $\lambda j,i$ measures the sufficiency of a piece of evidence e_j to prove hypothesis h_i. Similarly, $\lambda^*_{j,i}$ measures the necessity of such a piece of evidence to prove the given hypothesis (Pearl, 1982).

Equations 8 and 9 assume that evidence e_j is precise (ie, $P(e_j) \in \{0,1\}$). This is not the case in most expert system applications. Therefore, the above equations must be modified to accommodate uncertain evidence. This is accomplished by using a linear interpolation formula. For the case of single evidence, the posterior probability $P(h_i|e'_j)$ is computed as:

$$P(h_i|e'_j) = P(h_i|e_j)P(e_j|e'_j) + P(h_i|\neg e_j)P(\neg e_j|e'_j) \quad (10)$$

where $P(e_j|e'_j)$ is the user's assessment of the probability that the evidence e_j is true, given the relevant observation e'_j. An effective likelihood ratio, $\lambda'_{j,i}$, is calculated from the posterior odds:

$$\lambda'_{j,i} = \frac{O(h_i|e'_j)}{O(h_i)} \quad (11)$$

The posterior odds for all the evidence is then computed as:

$$O(h_i|e'_1, e'_2, \ . \ . \ . \ , e'_m) = O(h_i) \prod_{j=1}^{m} \lambda'_{j,i} \quad (12)$$

Equation 10, however, requires a modification, because it overconstrains the input requested from the user. In fact, the user must specify: $O(h_i)$, the prior odds on h_i from which $P(h_i)$ can be derived; $\lambda_{j,i}$, the measure of sufficiency from which $P(h_i|e_j)$ can be derived; $\lambda^*_{j,i}$, the measure of necessity from which $P(h_i|\neg e_j)$ can be derived; and $O(e_j)$, the prior odds on e_j from which $P(e_j)$ can be derived. These requirements are equivalent to specifying a line in the space $[P(e|e'),P(h_i|e')]$ by specifying three points:

$$(0,P(h_i|\neg e_j))$$

$$(P(e_j),P(h_i))$$

$$(1,P(h_i|e_j))$$

The modification adopted in this approach to prevent the user's inconsistencies is to change equation 10 into a piecewise linear function defined by two line segments passing through the above three points (Duda and co-workers, 1976).

In an analysis of this approach (Pednault and co-workers, 1981), it was concluded that for the cases of more than two hypotheses, equations 6 and 7, requiring conditional independence of the evidence both under the hypotheses and their negation, were inconsistent with equations 4 and 5, requiring an exhaustive and mutually exclusive space of hypotheses. Specifically, it was proved that, under these assumptions, no probabilistic update could take place, ie,

$$P(e_j|h_i) = P(e_j|\neg h_i) = P(e_j) \ \forall i,j \quad (13)$$

However, a pathological counterexample to equation 13 was obtained (Glymour, 1985), and a fault was found in the original proof of Hussain's theorem that constituted the basis for Pednault and co-workers' results. Johnson (1986) extended this analysis by first showing that there are also nonpathological counterexamples that refute Pednault's results. However, Johnson proved that under the same assumptions used in Pednault's work, for every hypothesis h_i there is at most one piece of evidence e_j that produces updating for h_i. Further studies (Cheng and Kashyap, 1986) have also indicated that there are at least max $[0,(m - [n/2])]$ pieces of evidence that are irrelevant to all the hypotheses in the system. An evidence e_j is said to be irrelevant to the hypothesis h_i if $P(h_i|e_j) = P(h_i)$. This lower bound is for a system satisfying equations 4 and 5, in which n is the number of mutually exclusive exhaustive hypotheses ($n > 2$), and m is the number of evidence. The conclusion was that equation 7 should be dropped.

Pearl (1985) has argued that equation 7, requiring the conditional independence of the evidence under the negation of the hypotheses, is overrestrictive. By discarding this assumption, Pearl has derived new, more promising results. However, equation 6, requiring the conditional independence of the evidence under the hypotheses, is still required for computational efficiency.

The Bayesian approach has various shortcomings. The assumptions on which it is based are not easily satisfied, eg, if the network contains multiple paths linking a given evidence to the same hypothesis, the independence equations 6 and 7 are violated. Similarly, equations 4 and 5, requiring the mutually exclusiveness and exhaustiveness of the hypotheses, are not very realistic; equation 4 would not hold if more than one hypothesis could occur simultaneously and is as restrictive as the single-fault assumption of the simplest diagnosing systems. Equation 5 implies that every possible hypothesis is *a priori* known, and it would be violated if the problem domain were not suitable to a closed-world assumption. Perhaps the most restrictive limitation of the Bayesian approach is its inability to represent ignorance (ie, noncommitment) as illustrated by its two-way betting interpretation (Giles, 1982). The two-way betting interpretation of the Bayesian approach consists of regarding the assignment of probability p to event A as the willingness of a rational agent to accept any of the two following bets:

1. If you pay me \$$p$ then I agree to pay you \$1 if A is true (for $p \in [0,1]$).
2. If you pay me \$$(1 - p)$ then I agree to pay you \$1 if A is false.

The first bet represents the belief that the probability of A is not larger than p, the second bet represents the belief that the probability of A is not smaller than p.

Instead of being explicitly represented, ignorance is hidden in prior probabilities. Further shortcomings are represented by the fact that it is impossible to assign any probability to disjunctions, ie, to nonsingletons, which implies the requirement for a uniform granularity of evidence. This problem is usually solved with an approximation, using the maximum entropy principle (MEP). According to MEP, the probability assigned to the disjunct (a subset of singletons in the sample space) is equally divided among the singletons in the subset. This approximation, however, creates an interpretation of the original information, which may not always been appropriate. Finally, as has been pointed out (Quinlan, 1983), in this approach conflictive information is not detected but simply propagated through the network.

Confirmation Theory (Certainty Factors)

The certainty factor (qv) (CF) approach (Shortliffe and Buchanan, 1975), used in MYCIN, is based on confirmation theory. The certainty factor CF(h,e) of a given hypothesis h is the difference between a measure of belief $MB(h,e)$ representing the degree of support of a (favorable) evidence e, and a measure of disbelief $MD(h,e)$ representing the degree of refutation of an (unfavorable) evidence e. MB and MD are monotonically increasing functions that are respectively updated when the new evidence supports or refutes the hypothesis under consideration. The certainty factor CF(h,e) is defined as:

$$\mathrm{CF}(h,e) = \begin{cases} 1 & \text{if } P(h) = 1 \\ MB(h,e) & \text{if } P(h|e) > P(h) \\ 0 & \text{if } P(h|e) = P(h) \\ -MD(h,e) & \text{if } P(h|e) < P(h) \\ -1 & \text{if } P(h) = 0 \end{cases} \qquad (14)$$

The measures of belief MB and measure of disbelief MD could be interpreted as a relative distance on a bounded interval. Given an interval $[A,B]$ and a reference point R within the interval, the relative distance $d(X,R)$ between any arbitrary point X within the interval and the reference R can be defined as:

$$d(X,R) = \begin{cases} \dfrac{(X - R)}{(B - R)} & \text{if } X > R \\ 0 & \text{if } X = R \\ \dfrac{(R - X)}{(R - A)} & \text{if } X < R \end{cases} \qquad (15)$$

By making the following substitutions in equation 15

$$A = 0$$
$$B = 1$$
$$R = P(h)$$
$$X = P(h|e)$$

the definition of the measure of belief (MB) and measure of disbelief (MD) can be obtained.

$$MB(h,e) = \begin{cases} \dfrac{P(h|e) - P(h)}{1 - P(h)} & \text{if } P(h|e) > P(h) \\ 0 & \text{otherwise} \end{cases} \qquad (16)$$

$$MD(h,e) = \begin{cases} \dfrac{P(h) - P(h|e)}{P(h)} & \text{if } P(h|e) < P(h) \\ 0 & \text{otherwise} \end{cases} \qquad (17)$$

The CF was originally interpreted as the relative increase or decrease of probabilities. In fact, from equations 14, 16, and 17, it can be shown that

$$P(h|e) = P(h) + \mathrm{CF}(h,e)[1 - P(h)] \quad \text{for } \mathrm{CF}(h,e) \geq 0 \qquad (18)$$

$$P(h|e) = P(h) - |\mathrm{CF}(h,e)|P(h) \quad \text{for } \mathrm{CF}(h,e) \leq 0 \qquad (19)$$

Too often the CF paradigm has been incorrectly used in reasoning systems, interpreting the CF as absolute rather than incremental probability values. The original interpretation of the CF as a probability ratio, however, can no longer be preserved after the CF have been aggregated using the heuristic combining functions provided in MYCIN (Shortliffe and Buchanan, 1975).

Ishizuka and co-workers (1982) have shown that these combining functions were an approximation of the classical Bayesian updating procedure in which a term had been neglected (Ishizuka, 1982). In their analysis it was concluded that the assumption of mutual independence of evidence was required for the correct use of this approach. The original definition of certainty factor is asymmetric and prevents commutativity. Another source of concern in the use of CF is caused by the normalization of MB and MD before their arithmetic difference is computed. This normalization hides the difference between the cardinality of the set of supporting evidence and that of the set of refuting evidence.

Buchanan and Shortliffe (1984) have proposed a change to the definition of CF and its rules of combination:

$$\mathrm{CF}(h,e) = \frac{MB(h,e) - MD(h,e)}{1 - \min(MB(h,e),MD(h,e))} \tag{20}$$

$$\mathrm{CF}_{\mathrm{combine}}(x,y) = \begin{cases} x + y - xy & \text{for } x > 0,\, y > 0 \\ \dfrac{x + y}{1 - \min(|x|,|y|)} & \text{for } x < 0,\, y > 0 \\ & \text{or } x > 0,\, y < 0 \\ -\mathrm{CF}_{\mathrm{combine}}(-x,-y) & \text{for } x < 0,\, y < 0 \end{cases} \tag{21}$$

where $CF(h,e_1) = x$ and $\mathrm{CF}(h,e_2) = y$. This new definition avoids the problem of allowing a single piece of negative (positive) evidence to overwhelm several pieces of positive (negative) evidence. However, it has even less theoretical justification or interpretation than the original formulas.

Recently, Heckerman (1986) has derived a new definition for the CF that does allow commutativity and has a consistent probabilistic interpretation. The new definition is

$$\mathrm{CF}(h,e) = \frac{P(h|e) - P(h)}{P(h|e)[1 - P(h)] + P(h)[1 - P(h|e)]} \tag{22}$$

There are still numerous serious problems that characterize this approach: the semantics of the CF, ie, the interpretation of the number (ratio of probability, combination of utility values and probability); the assumptions of independence of the evidence; and the inability of distinguishing between ignorance and conflict, both of which are represented by the assignment CF = 0.

This type of representation of uncertainty has also been advocated by Rich (1983), as an alternative to default reasoning. Rich claims that default reasoning could actually better be interpreted as likelihood reasoning, providing a uniform representation for statistical, prototypical, and definitional facts.

Bayesian Belief Networks

An efficient propagation of belief on Bayesian networks was originally proposed by Pearl (1982). Pearl described an efficient updating scheme for trees and, to a lesser extent, for polytrees (1988b). However, as the complexity of the graph increases from trees to polytrees to general graphs, so does the computational complexity. The complexity for trees is $O(n^2)$, where n is the number of values per node in the tree. The complexity for polytrees is $O(K^m)$, where K is the number of values per parent node and m is the number of parents per child. This number is the size of the table attached to each node. Because the table must be constructed manually (and updated automatically), it is reasonable to assume that it is small. The complexity for multiconnected graphs is $O(K^n)$, where K is the number of values per node and n is the size of the largest nondecomposable subgraph. To handle such complexity, techniques such as moralization and propagation in a tree of cliques (Lauritzen and Spiegelhalter, 1988) and loop cutset conditioning (Suermondt and co-workers, 1990; Stillman, 1990) are typically used to decompose the original problem (graph) into a set of smaller problems (subgraphs). When this problem decomposition process is not possible, exact methods must be abandoned in favor of approximate methods. Among these methods the most common are clustering, bounding conditioning (Horvitz and co-workers, 1989), and simulation techniques (logic samplings and Markov simulations). Figure 1 illustrates a taxonomy of these Bayesian inference mechanisms.

Dempster-Shafer (Belief Theory)

The belief theory (Shafer 1976) was developed within the framework of Dempster's work on upper and lower probabilities induced by a multivalued mapping (see DEMPSTER-SHAFER METHOD). The one-to-many nature of the mapping is the fundamental reason for the inability of applying the well-known theorem of probability that determines the probability density of the image of one-to-one mappings. In fact, given a differentiable strictly increasing or strictly decreasing function ϕ on an interval I, and a continuous random variable X with a density f, such that $f(x) = 0$ for any x outside I, then the density function g can be computed as:

$$g(y) = f(x)\left|\frac{dx}{dy}\right|$$

$$y \in \phi(I)$$

$$x = \phi^{-1}(y)$$

In this context, the lower probabilities have been identified as epistemic probabilities and associated with a degree of belief. This formalism defines certainty as a function that maps subsets of a space of propositions θ on the [0,1] scale. The sets of partial beliefs are represented by mass distributions of a unit of belief across the propositions in θ. This distribution is called basic probability assignment (BPA). The total certainty over the space is 1. A non-zero BPA can be given to the entire space θ to represent the degree of ignorance. Given a space of propositions θ, referred to as frame of discernment, a function $m : 2^\theta \rightarrow$

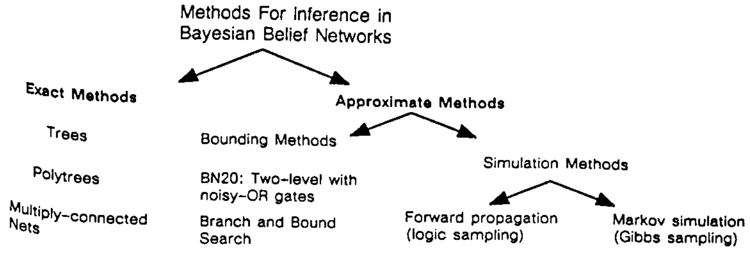

Methods For Inference in Bayesian Belief Networks
Exact Methods
Trees
Polytrees
Multiply-connected Nets
Approximate Methods
Bounding Methods
BN20: Two-level with noisy-OR gates
Branch and Bound Search
Simulation Methods
Forward propagation (logic sampling)
Markov simulation (Gibbs sampling)

Figure 1. Taxonomy of inference mechanisms for Bayesian belief networds. Courtesy of M. Henrion.

[0,1] is called a basic probability assignment if it satisfies the following three conditions:

$$m(\phi) = 0 \tag{23}$$

where ϕ is the empty set

$$0 < m(A) < 1 \tag{24}$$

$$\sum_{A \subseteq \theta} m(A) = 1 \tag{25}$$

The certainty of any proposition B is then represented by the interval $[Bel(B), P^*(B)]$, where $Bel(B)$ and $P^*(B)$ are defined as:

$$Bel(B) = \sum_{x \subseteq B} m(x) \tag{26}$$

$$P^*(B) = \sum_{x \cap B \neq \phi} m(x) \tag{27}$$

From the above definitions the following relation can be derived

$$Bel(B) = 1 - P^*(\neg B) \tag{28}$$

If m_1 and m_2 are two BPA induced from two independent sources, a third BPA, $m(C)$, expressing the pooling of the evidence from the two sources, can be computed by using Dempster's rule of combination:

$$m(C) = \frac{\sum_{A_i \cap B_j = C} m_1(A_i) m_2(B_j)}{1 - \sum_{A_i \cap B_j = \phi} m_1(A_i) m_2(B_j)} \tag{29}$$

Dempster's rule of combination normalizes the intersection of the bodies of evidence from the two sources by the amount of nonconflictive evidence between the sources. This amount is represented by the denominator of the formula.

There are two problems with the belief theory approach. The first problem stems from computational complexity: in the general case, the evaluation of the degree of belief and upper probability requires time exponential in $|\theta|$, the cardinality of the hypothesis set (frame of discernment). This is caused by the need of (possibly) enumerating all the subsets and supersets of a given set. Barnett (1981) showed that, when the frame of discernment is discrete (and simple support functions are used), the computational time complexity could be reduced from exponential to linear by combining the belief functions in a simplifying order. Strat (1984) proved thεt the complexity could be reduced to $O(n^2)$, where n is the number of atomic propositions, ie, intervals of unit length, when the frame of discernment is continuous. In both cases, however, these results were achieved by introducing various assumptions about the type and structure of the evidence to be combined and about the hypotheses to be supported. As a result, in addition to the requirements of mutual exclusive hypotheses and independent evidence that are needed by this approach, the following constraints must be included. For the case of discrete frame of discernment, each piece of evidence is assumed to support only a singleton proposition or its negation rather than disjunctions of propositions (ie, propositions with larger granularity), and for the case of continuous frame of discernment, only contiguous intervals along the number line can be included in the frame of discernment and thus receive support from the evidence.

The second problem in this approach results from the normalization process present in both Dempster's and Shafer's work. Zadeh (1984b, 1985b) has argued that this normalization process can lead to incorrect and counterintuitive results. By removing the conflictive parts of the evidence and normalizing the remaining parts, important information is discarded rather than being dealt with adequately. A proposed solution to this problem is to avoid completely the normalization process by maintaining an explicit measure of the amount of conflict and by allowing the remaining information to be subnormal (ie, $Bel(\theta) < 1$). Zadeh (1985b) has proposed a test to determine the conditions of applicability of Dempster's rule of combination. Dubois and Prade (1985) have also shown that the normalization process in the rule of evidence combination creates a sensitivity problem, where assigning a zero value or a very small value to a BPA causes very different results. It should be noted that this behavior also occurs in other probabilistic schemes, where the assignment of a value of zero to a prior probability would prevent any subsequent updating.

Ginsberg (1984) has proposed the use of the Dempster-Shafer approach as an alternative to nonmonotonic logic. This suggestion is an extension to Rich's (1983) idea of interpreting default reasoning as likelihood reasoning. Ginsberg provides a rule for propagating the lower and upper bounds through a reasoning chain or graph. The result is based on the interpretation of a production rule as a conditional probability rather than as a material implication. Smets (1981, 1988) has further explained the relations between belief functions, plausibilities, necessities, and possibilities and has extended Dempster's concepts to handle the case when the evidence is a fuzzy set (Zadeh, 1965).

Evidential Reasoning

Evidential reasoning (Garvey and co-workers, 1981; Lowrance and Garvey, 1983; Lowrance and co-workers, 1986) adopts the evidential interpretation of the degrees of belief and upper probabilities. Fundamentally based on Dempster-Shafer's theory, this approach defines the likelihood of a proposition A as a subinterval of the unit interval [0,1]. The lower bound of this interval is the degree of support of the proposition $S(A)$, and the upper bound is its degree of plausibility $Pl(A)$. The likelihood of a proposition A is written as $A_{[S(A), Pl(A)]}$. Table 1 illustrates a sample of interval-valued likelihoods and their interpretation.

Given two statements $A_{[S(A), Pl(A)]}$ and $B_{[S(B), Pl(B)]}$, the set of inference rules corresponding to the logical operations on these statements are defined (Garvey and co-workers, 1981) as follows.

Table 1. Sample of Interval-Valued Likelihoods and Their Interpretations

$A_{[0,1]}$	No knowledge at all about A
$A_{[0,0]}$	A is false
$A_{[1,1]}$	A is true
$A_{[0.3,1]}$	The evidence partially supports A
$A_{[0,0.7]}$	The evidence partially supports $\neg A$
$A_{[0.3,0.7]}$	The evidence simultaneously provides partial support for A and $\neg A$
$A_{[0.3,0.3]}$	The probability of A is exactly 0.3

$$\text{Intersection: } AND(A,B)_{[\max(0,S(A)+S(B)-1),\min(Pl(A),Pl(B))]} \tag{30}$$

$$\text{Union: } OR(A,B)_{[\max(S(A),S(B)),\min(1,Pl(A)+Pl(B))]} \tag{31}$$

$$\text{Negation: } NOT(A)_{[1-Pl(A),1-S(A)]} \tag{32}$$

This approach, embodied in GISTER (Lowrance and coworkers, 1986), implements Dempster-Shafer theory. When distinct bodies of evidence must be pooled, this approach uses the same Dempster-Shafer techniques, requiring the same normalization process that was criticized by Zadeh.

Evidence Space

Evidence space (Rollinger, 1983) represents the uncertainty of a statement as a point in a two-dimensional space. The (X,Y) coordinates of this space represent the positive or supporting evidence ($E+$) and the negative or disconfirming evidence ($E-$) available for any given proposition, respectively. The evidence space is a $[0,1]x[0,1]$ square whose four vertices represent ignorance (0,0), absolute certainty in the support (1,0), absolute certainty in the refutation (0,1), and maximum conflictive evidence (1,1). The diagonal line defined by the equation $x + y - 1 = 0$ represents the locus of probability points, the sum of whose coordinates is 1.

It is interesting to note that if the dimensions of the evidence space ($E+$, $E-$) represent the necessary evidence, ie, the lower bounds of the degree of support and refutation ($S(E)$, $S(-E)$), the evidence space is reduced to the lower left triangle. Its three vertices (0,0), (1,0), and (0,1) represent ignorance, absolute support, and absolute refutation, respectively. The maximum amount of conflict is given by the point (0.5, 0.5). On the other hand, if the dimensions of the evidence space ($E+$, $E-$) represent the possible evidence, ie, the upper bounds of the degree of support and refutation ($Pl(E)$,$Pl(\neg E)$), the evidence space is reduced to the upper right triangle. Its three vertices (1,1), (1,0), and (0,1) represent ignorance, absolute support, and absolute refutation, respectively. The maximum amount of conflict is again given by the point (0.5,0.5). If the lower bounds are equated to the upper bounds, ie, $(S(E),S(\neg E)) = (Pl(E),Pl(\neg E))$, a new set of coordinates $(P(E),P(\neg E))$, representing Bayesian probability, can be obtained. In this new set of coordinates, the evidence space collapses to the diagonal line $x + y - 1 = 0$ that is the intersection of the two triangles and that indeed represents the probability line.

Rollinger suggests the use of a distance to verify the validity of any given premise in a rule. This approach, however, does not suggest any way of aggregating evidence, propagating uncertainty through an inference chain, selecting an appropriate metric of similarity between patterns and data, etc.

POSSIBILISTIC APPROACHES

Among the possibilistic reasoning techniques, the ones based on many-valued logic operators (triangular norms, or T-norms) and the generalized *modulus ponens* will be discussed.

Triangular Norm Based Reasoning Systems

These possibilistic techniques have been implemented in a reasoning with uncertainty module (RUM) (Bonissone and Decker, 1986; Bonissone and Wood, 1989).

Uncertainty in RUM is represented in both facts and rules. A fact represents the assignment of a value to a variable. A rule represents the deduction of a new fact (conclusion) from a set of given facts (premises). Facts are qualified by a degree of confirmation and a degree of refutation. For a fact A, the lower bound of the confirmation and the lower bound of the refutation are denoted by $L(A)$ and $L(\neg A)$, respectively. As in the case of Dempster's (1967) lower and upper probability bounds, the following identity holds: $L(\neg A) = 1 - U(A)$, where $U(A)$ denotes the upper bound of the uncertainty in A and is interpreted as the amount of failure to refute A. Note that $L(A) + L(\neg A)$, need not necessarily be equal to 1, as there may be some ignorance about A, that is given by $(1 - L(A) - L(A))$. The degree of confirmation and refutation for the proposition A can be written as the interval $[L(A),U(A)]$.

RUM provides a natural representation for plausible rules. Rules are discounted by sufficiency (s), indicating the strength with which the antecedent implies the consequent, and necessity (n), indicating the degree to which a failed antecedent implies a negated consequent. Note that conventional strict implication rules are special cases of plausible rules with $s = 1$ and $n = 0$. RUM's inference layer is built on a set of five triangular norms (T-norms) based calculi (Bonissone and Decker, 1986; Bonissone, 1987a). T-norms and T-conorms are two-place functions from $[0,1]x[0,1]$ to $[0,1]$ that are monotonic, commutative and associative. They are the most general families of binary functions that satisfy the requirements of the conjunction and disjunction operators, respectively. Their corresponding boundary conditions satisfy the truth tables of the logical AND and OR operators. Five uncertainty calculi based on the following five T-norms are used in RUM:

$$T_1(a,b) = \max(0,a + b - 1)$$

$$\begin{aligned} T_{1.5}(a,b) &= (a^{0.5} + b^{0.5} - 1)^2 && \text{if } (a^{0.5} + b^{0.5}) \geq 1 \\ &= 0 && \text{otherwise} \end{aligned}$$

$$T_2(a,b) = ab$$

$$T_{2.5}(a,b) = (a^{-1} + b^{-1} - 1)^{-1}$$

$$T_3(a,b) = \min(a,b)$$

Their corresponding DeMorgan dual T-conorms, denoted by $S_i(a,b)$, are defined as

$$S_i(a,b) = 1 = T_i(1 - a, 1 - b)$$

These five calculi provide the user with an ability to choose the desired uncertainty calculus starting from the most conservative (T_1) to the most liberal (T_3). T_1 (T_3) is the most conservative (liberal) T-norm in the sense that for the same input certainty ranges of facts and rule sufficiency and necessity measures, T_1 (T_3) shall yield the minimum (maximum) degree of confirmation of the conclusion. For each calculus (represented by the above five T-norms), the following four operations have been defined in RUM.

Antecedent Evaluation. To determine the aggregated certainty range $[b,B]$ of the n clauses in the antecedent of a rule, when the certainty range of the ith clause is given by $[b_i,B_i]$:

$$[b,B] = [T_i(b_1,b_2, \ . \ . \ . \ ,b_n),T_i(B_1,B_2, \ . \ . \ . \ ,B_n)]$$

Conclusion Detachment: Modus Ponens. To determine the certainty range, $[c,C]$ of the conclusion of a rule, given the aggregated certainty range, $[b,B]$ of the rule premise and the rule sufficiency s and rule necessity n:

$$[c,C] = [T_i(s,b),1 - (T_i(n,(1 - B)))]$$

Conclusion Aggregation. To determine the consolidated certainty range $[d,D]$, of a conclusion when it is supported by m ($m > 1$) paths in the rule deduction graph, ie, by m rule instances, each with the same conclusion aggregation T-conorm operator. If $[c_i,C_i]$ represents the certainty range of the same conclusion inferred by the ith proof path (rule instance), then

$$[d,D] = [S_i(c_1c_2, \ . \ . \ . \ ,c_m),S_i(C_1,C_2, \ . \ . \ . \ ,C_m)]$$

Source Consensus. To determine the certainty range, $[L_{\text{tot}}(A),U_{\text{tot}}(A)]$ of the same evidence, A, obtained by fusing the certainty ranges, $[L_i(A),U_i(A)]$, of the ith information source out of a total of n different possible information sources:

$$[L_{\text{tot}}(A),U_{\text{tot}}(A)] = [\underset{i=1,\ldots,n}{\text{Max}} \ L_i(A), \ \underset{i=1,\ldots,n}{\text{Min}} \ U_i(A)]$$

The theory of possibilistic reasoning has been embedded in the reasoning with uncertainty module (Bonissone and co-workers, 1987) and the plausible reasoning modules (PRIMO) (Bonissone and co-workers, 1990).

Possibilistic Reasoning System: RUM

RUM's rule-based system integrates both procedural and declarative knowledge in its representation. This integration is essential for solving situation assessment problems, which involve both heuristic and procedural knowledge.

The expressiveness of RUM is further enhanced by two other functionalities: the context mechanism and belief revision. The context represents the set of preconditions determining the rule's applicability to a given situation. This mechanism provides an efficient screening of the knowledge base by focusing the inference process on small rule subsets. For instance, in SA, selected rules describe the behavior of friendly planes, whereas others should only be applied to unfriendly or unidentified ones. The rule's context provides this filtering mechanism.

RUM's belief revision is essential to the dynamic aspect of the classification problem. The belief revision mechanism detects changes in the input, keeps track of the dependency of intermediate and final conclusions on these inputs, and maintains the validity of these inferences. For any conclusion made by a rule, the mechanism monitors the changes in the certainty measures that constitute the conclusion's support. Validity flags are used to reflect the state of the certainty. For example, a flag can indicate that the uncertainty measure is valid, unreliable (because of a change in the support), too ignorant to be useful, or inconsistent with respect to the other evidence. These AI capabilities are used to develop a knowledge base, in conjunction with RUM's software engineering facilities, such as flexible editing, error checking, and debugging.

Possibilistic Reasoning System: PRIMO

The most recently developed technology embodying possibilistic reasoning techniques is the plausible reasoning module (PRIMO) (Bonissone and co-workers, 1990). PRIMO is a reasoning system that integrates the theories of plausible reasoning (based on monotonic rules with degrees of uncertainty) and defeasible reasoning (based on default values supported by nonmonotonic rules). The PRIMO system consists of a representation language that includes declarative specifications of uncertainty and default knowledge, reasoning algorithms, and an application development environment.

PRIMO, like its predecssor RUM, handles uncertain information by qualifying each possible value assignment to any given propositional variable with an uncertainty interval. The interval's lower bound represents the minimal degree of confirmation for the value assignment. The upper bound represents the degree to which the evidence failed to refute the value assignment. The interval's width represents the amount of ignorance attached to the value assignment. The uncertainty intervals are propagated and aggregated by triangular-norm–based uncertainty calculi (Bonissone and Decker, 1986; Bonissone, 1987a; Schweizer and Sklar, 1983, 1963). The uncertainty interval constrains intervals of subsequent dependent values.

PRIMO handles incomplete information by evaluating nonmonotonic justified (NMJ) rules. These rules are used to express the knowledge engineer's preference in cases of total or partial ignorance regarding the value assignment of a given propositional variable. The NMJ rules are used when there is no plausible evidence (to a given numerical threshold of belief or certainty) to infer that a given value assignment is either true or false. The conclusions of NMJ rules can be retracted by the belief revision system, when enough plausible evidence is available.

PRIMO uses the numerical certainty values generated by plausible reasoning techniques to quantitatively distinguish the admissible extensions generated by defeasible reasoning techniques. The method selects a maximally consistent extension (Bonissone and co-workers, 1990) given all currently available information.

For efficiency considerations some restrictions are placed on the language in which PRIMO rules can be expressed. The monotonic rules are noncyclic Horn clauses and are maintained by a linear belief revision algorithm operating on a rule graph. The NMJ rules can have cycles, but cannot have disjunctions in their conclusions.

By identifying sets of NMJ rules as strongly connected components (SCCs), the rule graph can be decomposed into a directed acyclic graph (DAG) of nodes, some of which are SCC with several input edges and output edges. PRIMO contains algorithms to efficiently propagate uncertain and incomplete information through these structures at run time. Treating the SCC independently can result in a significant performance improvement over processing the entire graph. However, this heuristic may result in loss of correctness in the worst case. These algorithms require finding satisfying assignments for nodes in each SCC, and are thus NP-hard in the unrestricted case. Tractability can be achieved by restricting the size and complexity of the SCCS, precomputing their structural information, and using run-time evaluated certainty measures to select the most likely extension.

Necessity and Possibility Theory

Necessity and possibility (Zadeh, 1979a, 1978) measure the degree of entailment and intersection of two fuzzy propositions represented by their normalized possibility distributions. Normal necessity and possibilities correspond to consonant belief and plausibility functions, respectively. Given two fuzzy propositions P and $D \subset U$, characterized by their possibility distributions $\mu_P(x)$ and $\mu_D(x)$, their degree of matching is represented by the interval [Nec($P|D$),Poss($P|D$)], where

$$\begin{aligned} N(P|D) &= \bigwedge_x (\mu_D(x) \rightarrow \mu_P(x)) \\ &= \bigwedge_x (\max[(1 - \mu_D(x)), \mu_P(x)]) \\ &= 1 - \bigvee_x (\min[(1 - \mu_P(x)), \mu_D(x)]) \qquad (33) \end{aligned}$$

$$\Pi(P|D) = \bigvee_x (\min[\mu_P(x), \mu_D(x)]) \qquad (34)$$

From the above definition, it is possible to derive for necessity and possibility the same duality observed between belief functions and upper probabilities (eq. 22)

$$\mathrm{Nec}(P|D) = 1 - \mathrm{Poss}(\neg P|D) \qquad (35)$$

The intersection of necessity measures and the union of possibility measures provide tighter bounds than those obtained by the intersection of belief functions and the union of plausibility functions (Prade, 1985).

QUALITATIVE REPRESENTATION

Among the nonnumerical representations of uncertainty, two approaches typify the characterization of uncertain information in a purely symbolic manner: reasoned assumptions and theory of endorsements.

Reasoned Assumptions

In the reasoned assumption approach (Doyle, 1983) the uncertainty embedded in an implication is (partially) removed by listing all the exceptions to that rule. When this is not possible, assumptions are used to show typicality of a value (default values) and defeasibility of a rule (liability to defeat of a reason). When an assumption used in the deductive process is found to be false, nonmonotonic mechanisms are used to keep the integrity of the data base of statements. Assumption based systems can cope with the case of incomplete information, but they are inadequate to handle the case of imprecise information. In particular, they cannot integrate probabilistic information with reasoned assumptions. Furthermore, these systems rely on the precision of the defaulted value. On the other hand, when specific information is missing, the system should be able to use analogous or relevant information inherited from some higher level concept. This surrogate for the missing information is generally fuzzy or imprecise and only provides some elastic constraints on the value of the missing information. Doyle (1983) recognized that assumptions based systems lack facilities for computing degrees of belief, which "may be necessary for summarizing the structure of large sets of admissible extensions as well as for quantifying confidence levels."

Theory of Endorsements

A different approach to uncertainty representation was recently proposed (Cohen and Grinberg, 1983a, 1983b), and is based on a purely qualitative Theory of Endorsements. Endorsements are based on the explicit recording of the justifications for a statement, as in a truth maintenance system. In addition, endorsements classify the justification according to the type of evidence (for and against a proposition), the possible actions required to solve the uncertainty of that evidence, and other related features. Endorsements provide a good mechanism for explanations, because they create and maintain the entire history of justifications (reasons for believing or disbelieving a proposition) and the relevance of any proposition with respect to a given goal. Endorsements are divided into five classes: rules, data, tasks, conclusions, and resolution endorsements. However, combination of endorsements in a premise, propagation of endorsements to a conclusion, and ranking of endorsements must be explicitly specified for each particular context, creating potential combinatorial problems.

COMPARISON OF APPROACHES FOR REASONING WITH UNCERTAINTY

From previous reviews of the state of the art of reasoning systems (Bonissone and Brown, 1986) and from previous analysis of applications (Bonissone, 1987a, 1987b; Bonissone and Wood, 1988), a desiderata (ie, a list of requirements to be satisfied by the ideal formalism for representing uncertainty and making inference with uncertainty) has been derived. In this section, the approximate reasoning technologies described above will be compared to the desiderata. This idea was first proposed by Quinlan (1983), who suggested a list of four requirements to illustrate the shortcomings of the Bayesian and confirmation theory approaches and to compare them with INFERNO, his proposed approach to uncertain inference. The requirements proposed by Quinlan are

- "An inference system should not depend on any assumptions about the probability distributions of the propositions."
- "It should be possible to assert common relationships between propositions . . . when the relationships are indeed known."
- "It should be possible to posit information about any set of propositions and observe the consequences for the system as a whole."
- "If the information provided to the system is inconsistent, this fact should be made evident along with some notion of alternative ways that the information could be made consistent."

Quinlan's work has been inspirational in the development of the following desiderata, which subsumes and extends Quinlan's initial list. As noted above, the proposed desiderata describes the requirements to be satisfied by the ideal formalism for representing uncertainty and making inference with uncertain information. To be consistent with the organizing principle typical of automated reasoning systems, the desiderata is subdivided into the same three layers of representation, inference, and control.

Representation Layer

1. There should be an explicit representation for the amount of evidence for supporting and for refuting any given hypothesis.
2. There should be an explicit representation of the information about the evidence, ie, meta-information, such as evidence source and creditibility, logical dependencies, etc.
3. The representation should allow the user to describe the uncertainty of information at the available level of detail, ranging from singletons to any subset of the universe of discourse. This property will be referred to as heterogeneous information granularity.
4. There should be an explicit representation of consistency. Some measure of consistency or compatibility should be available to detect trends of potential conflicts and to identify essential contributing factors in the conflict.
5. There should be an explicit representation of ignorance to allow the user to make noncommitting statements, ie, to express the user's lack of conviction about the certainty of any of the available choices or events.
6. The representation should be natural to the user to enable the description of uncertain input and to interpret uncertain output. The representation should also be natural to the expert to enable the elicitation of consistent weights representing the strength of the implication of each rule.

Inference Layer

7. The combining rules should not be based on global assumptions of evidence independence.
8. The combining rules should not be based on global assumptions of hypotheses exhaustiveness and exclusiveness.
9. The combining rules should maintain the closure of the syntax and semantics of the representation of uncertainty.
10. Any function used to propagate and summarize uncertainty should have clear semantics. This is needed both to maintain the semantic closure of the representation and to allow the control layer to select the most appropriate combining rules.

Control Layer

11. There should be a clear distinction between a conflict in the information (ie, violation of consistency), and ignorance about the information. To solve the conflict, the controller (meta-reasoner) must retract one or more elements of the conflicting set of evidence. To remove the ignorance, the controller must select a (retractable) default value or tag the information with an assumption.
12. The traceability of the aggregation and propagation of uncertainty through the reasoning process should be available to resolve conflicts, to explain the support of conclusions, and to perform meta-reasoning for control.
13. It should be possible to make pairwise comparisons of uncertainty because the induced ordinal or cardinal ranking is needed for performing any kind of decision-making activities.
14. It should be possible to select the most appropriate combination rule by using a declarative form of control (ie, by using a set of context-dependent rules that specify the selection policies).

EVALUATION OF THE APPROACHES

The above desiderata was used to guide the development of RUM and PRIMO. Table 2 summarizes the evaluation of the formalisms discussed in the previous section against this desiderata. The order in which the formal-

Table 2. Evaluation of Uncertainty Approaches against the Desiderata

	Representation						Inference				Control			
Approach	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Modified Bayesian	N	N	N	N	N	Y	N	N	Y	Y	N	N	Y	N
Confirmation	N	N	N	N	Y	N	N	Y	N	N	N	N	N	N
Dempster-Shafer	Y	N	Y	Y	Y	Y	N	N	Y	Y	Y	N	Y	N
Probability bounds	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y
Fuzzy necessity–possibility	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	N
Evidence space	Y	N	N	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	N
RUM–PRIMO	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Reasoned assumptions	N	Y	N	Y	N	Y	Y	Y	Y	Y	N	Y	N	Y
Endorsements	N	Y	N	Y	N	Y	Y	Y	Y	Y	N	Y	N	Y

isms appear in the table reflects their numeric or nonnumeric nature: the numeric formalisms are listed above RUM–PRIMO, the nonnumeric ones are shown below it. RUM–PRIMO is considered a hybrid, because it uses both numeric and symbolic information.

REAL-TIME APPROXIMATE REASONING SYSTEMS

This survey will conclude with a few remarks on the applicability of approximate reasoning systems (probabilistic and possibilistic) to many real-world problems requiring real-time performance. To achieve real-time performance levels, probabilistic reasoning systems need an efficient updating algorithm. The main problem consists in conditioning the existing information with respect to the new evidence: the computation of the new posterior probabilities in general belief networks is NP-hard (Cooper, 1990). A variety of solutions have been proposed, ranging from compilation techniques (to shift the burden from run time to compile time) to the determination of bounds of the posterior probabilities.

Horvitz (1988) and Breese and Frehling (1990) have established the applicability of decision–theoretic principles in defining bounded rationality for reasoning with limited resources. Heckerman and co-workers (1989) have provided a decision–theoretic based analysis of computation versus compilation. Given a description of the nature of evidential relationships in the domain, the utilities attached to alternative actions, the cost of run-time delay, and the cost of memory, their analysis determines the subset of evidence that is more cost-effective to compile. The analysis also determines the conditions under which run-time computation is preferable to look-up tables (generated at compile time).

Horvitz and co-workers (1989) have proposed a method to approximate the posterior probabilities of the variables in each subgraph of a belief network. This method, called bounded conditioning, defines the upper and lower bounds of these probabilities and, if given enough resources, converges on the final point probabilities.

A rather different approach has been suggested by D'Ambrosio. In contrast with the anytime algorithms discussed above (Dean and Boddy, 1988; Horvitz and co-workers, 1989), D'Ambrosio has proposed a design-to-time algorithm. Whereas the anytime algorithms try to yield a result any time they are interrupted, the design-to-time algorithms seek to "dynamically construct and execute a problem solving procedure which will [probably] produce a reasonable answer within [approximately] the time available" (D'Ambrosio, 1989). D'Ambrosio's (1988, 1990) initial development of the hybrid uncertainty management (HUM) seeks to provide an incremental and defeasible model, using an assumption-based truth maintenance system (ATMS) to maintain a mapping between symbolic structures (assumptions, logical support, and environments) and measures (numeric values for ranking and decision making). These ideas have been extended and integrated with a dynamic schema instantation (DSI), which, given a time bound, dynamically instantiates a qualitative probabilistic model of the problem.

Due to its different underlying theory, possibilistic reasoning does not exhibit the same complexity problems as probabilistic reasoning. Most of the efforts aimed at achieving real-time performance from possibilistic reasoning systems have been based on translation–compilation techniques (Pfau, 1987; Bonissone and Halverson, 1990) or hardware solutions (Corder, 1989; Watanabe and Dettloff, 1987).

Among the compilation techniques, a notable effort is RUMrunner, RUM's run-time system. The objective of RUMrunner is to provide a software tool that transforms the customized knowledge base generated during the development phase into a fast and efficient real-time application.

This goal is achieved by a combination of efforts: the translation of RUM's (development system) complex data structure into simpler and more efficient ones (to reduce overhead), the compilation of the rule set into a compiled network (to avoid run-time search), the load-time estimation of each rule's execution cost (to determine, at runtime, the execution cost of any given deductive path), and the planning mechanism for model selection (to determine the largest relevant rule subset that could be executed within a given time-budget).

An agenda mechanism is used to asynchronously receive any number of input tasks (such as backward-chaining on a goal or forward-chaining on a given piece of evidence) from various sources. Each task in the agenda receives a (static) priority number, determining the relative importance of the task with respect to the others. A time deadline, expressed in absolute time, is attached to the task to indicate its urgency (ie, its expiration time), which is used by the planning mechanisms described below.

A scheduler sorts the tasks by priority and, within the same priority level, by the shortest deadline. The highest priority task is then scheduled for execution by the forward or backward chainer (Durfee and Lesser, 1987). The results of these tasks are in turn isolated from external connecting systems via buffers or streams and a layer of interface functions.

External or internal interrupts, with reentrant reasoning, can supersede the current task. Because the state of the current knowledge base is dynamically maintained in the knowledge base nodes themselves, any changes to the knowledge base by the interrupting task will be automatically taken into account when the preempted task is resumed.

Among the hardware solutions to the problem of real-time performance for possibilistic reasoning systems, the most notable are the fuzzy chips (Corder, 1989; Watanabe and Dettloff, 1987). These chips are used in the application of approximate reasoning systems to industrial control. Fuzzy process controllers (Sugeno, 1985) represent one of the earliest instances of simple, but effective, knowledge-based systems successfully deployed in the field. Their main use has been the replacement of the human operator in the feedback control loop of industrial processes. Their applications range from the development of the controller of a subway train system (Yasunobu and Miyamoto, 1985) to the use of a predictive fuzzy controller for container crane operation (Yasunobu and Hasegawa, 1986) to their application in the control of a continuously variable automobile transmission (Kasia and Morimoto, 1988).

BIBLIOGRAPHY

A. Agogino and A. Rege, "Ides: Influence Diagram Based Expert System," *Math. Model.* **8**, 227–233 (1987).

J. Barnett, "Computational Methods for a Mathematical Theory of Evidence," in *Proceedings of the Seventh IJCAI,* Morgan-Kaufmann, San Mateo, Calif., Vancouver, B.C., 1981.

P. P. Bonissone, "Summarizing and Propagating Uncertain Information with Triangular Norms," *Int. J. Approx. Reas.* **1**(1), 71–101 (Jan. 1987a).

P. P. Bonissone, "Using T-Norm Based Uncertainty Calculi in a Naval Situation Assessment Application," in *Proceedings of the Third AAAI Workshop on Uncertainty in Artificial Intelligence,* AAAI, Menlo Park, Calif., July 1987b, pp. 250–261.

P. P. Bonissone, "Now that I Have a Good Theory of Uncertainty, What Else Do I Need?" in M. Henrion, R. Shachter, L. Kanal, and J. Lemmer, eds., *Uncertainty in Artificial Intelligence,* Vol. 5, North-Holland, Amsterdam, The Netherlands, 1990, pp. 237–253.

P. P. Bonissone and A. L. Brown, "Expanding the Horizons of Expert Systems," in T. Bernold, ed., *Expert Systems and Knowledge Engineering,* North-Holland, Amsterdam, The Netherlands, 1986, pp. 267–288.

P. P. Bonissone, D. Cyrluk, J. Goodwin, and J. Stillman, "Uncertainty and Incompleteness: Breaking the Symmetry of Defeasible Reasoning," in Bonissone, 1990, pp. 67–85.

P. P. Bonissone and K. S. Decker, "Selecting Uncertainty Calculi and Granularity: An Experiment in Trading-off Precision and Complexity," in L. Kanal and J. Lemmer, eds., *Uncertainty in Artificial Intelligence,* North-Holland, Amsterdam, The Netherlands, 1986, pp. 217–247.

P. P. Bonissone, S. Gans, and K. S. Decker, "RUM: A Layered Architecture for Reasoning with Uncertainty," *Proceedings of the Tenth IJCAI,* Milan, Italy, Morgan-Kaufmann, San Mateo, Calif., 1987, pp. 891–898.

P. P. Bonissone and P. C. Halverson, "Time-Constrained Reasoning Under Uncertainty," *J. Real Time Sys.* **2**(1–2), 22–45 (May 1990).

P. P. Bonissone and R. M. Tong, "Editorial: Reasoning with Uncertainty in Expert Systems," *Int. J. Man-Machine Stud.* **22**(3), 241–250 (Mar. 1985).

P. P. Bonissone and N. C. Wood, "Plausible Reasoning in Dynamic Classification Problems," in *Proceedings of the Validation and Testing of Knowledge-Based Systems Workshop,* AAAI, Menlo Park, Calif., Aug. 1988.

P. P. Bonissone and N. C. Wood, "T-Norm Based Reasoning in Situation Assessment Applications," in L. Kanal, T. Levitt, and J. Lemmer, eds., *Uncertainty in Artificial Intelligence,* Vol. 3, North-Holland, Amsterdam, The Netherlands, 1989, pp. 241–256.

J. S. Breese and M. R. Fehling, "Control of Problem Solving: Principles and Architecture," in R. Shachter, T. Levitt, L. Kanal, and J. Lemmer, eds., *Uncertainty in Artificial Intelligence,* Vol. 4, North-Holland, Amsterdam, The Netherlands, 1990, pp. 59–68.

B. Buchanan and E. Shortliffe, *"Rule-Based Expert Systems,"* Addison-Wesley Publishing Co., Inc., Reading, Mass., 1984.

Y. Cheng and R. Kashyap, *Irrelevancy of Evidence Caused by Independence Assumptions.* Technical Report TR-EE-86-17, School of Electrical Engineering, Purdue University, West Lafayette, Ind., 1986.

P. Cohen, *Heuristic Reasoning about Uncertainty: An Artificial Intelligence Approach.* Pittman, Boston, 1985.

P. Cohen and M. Grinberg, "A Framework for Heuristics Reasoning about Uncertainty," In *Proceedings of the Eighth IJCAI,* Karlsruhe, FRG, Morgan-Kaufmann, San Mateo, Calif., 1983a, pp. 355–357.

P. Cohen and M. Grinberg, "A Theory of Heuristics Reasoning about Uncertainty," *AI Mag.,* 17–23 (1983b).

G. Cooper, "The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks," *Artif. Intell.* **42**(2–3), 393–405 (1990).

R. Corder, "A High Speed Fuzzy Processor," in *Proceedings of the Third International Fuzzy Systems Association,* IFSA, Seattle, Wash., Aug. 1989, pp. 379–389.

B. D'Ambrosio, "A Hybrid Approach to Reasoning Under Uncertainty," *Int. J. Approx. Reas.* **2**(1), 29–45 (Jan. 1988).

B. D'Ambrosio, "Resource Bounded-Agents in an Uncertain World," in *Proceedings of the AAAI Workshop on Real-Time Artificial Intelligence Problems,* AAAI, Menlo Park, Calif., Aug. 1989.

B. D'Ambrosio, "Process Structure, and Modularity in Reasoning with Uncertainty," in R. Shachter, T. Levitt, L. Kanal, and J. Lemmer, eds., *Uncertainty in Artificial Intelligence,* Vol. 4, North-Holland, Amsterdam, 1990, pp. 15–25.

T. Dean and M. Boddy, "An Analysis of Time Dependent Planning," in *Proceedings of the Seventh National Conference on Artificial Intelligence,* St. Paul, Minn., AAAI, Menlo Park, Calif., 1988, pp. 49–54.

A. Dempster, "Upper and Lower Probabilities Induced by a Multivalued Mapping," *Ann. Math. Stat.* **38**, 325–339 (1967).

J. Doyle, "Methodological Simplicity in Expert System Construction: The Case of Judgements and Reasoned Assumptions, *AI Mag.* **4**(2), 39–43 (1983).

D. Dubois and H. Prade, "Criteria Aggregation and Ranking of Alternatives in the Framework of Fuzzy Set Theory," in H. Zimmerman, L. Zadeh, and B. Gaines, eds., *TIMS/Studies in the Management Science*, Vol. 20, Elsevier Science Publishing Co., Inc., New York, 1984, pp. 209–240.

D. Dubois and H. Prade, "Combination and Propagation of Uncertainty with Belief Functions—A Reexamination," in *Proceedings of the Ninth IJCAI*, Los Angeles, Calif., Morgan-Kaufmann, San Mateo, Calif., 1985, pp. 111–113.

R. Duda, P. Hart, and N. Nilsson, "Subjective Bayesian Methods for Rule-Based Inference Systems," *Proc. AFIPS* **45**, 1075–1082 (1976).

E. H. Durfee and V. R. Lesser, *Planning to Meet Deadlines in a Blackboard-Based Problem Solver*, Technical Report COINS-87-07, COINS, University of Massachusetts, Amherst, 1987.

T. Garvey, J. Lowrance, and M. Fischler, "An Inference Technique for Integrating Knowledge from Disparate Sources," in Barnett, 1981, 319–325.

R. Giles, "Semantics for Fuzzy Reasoning," *Int. J. Man-Machine Stud.* **17**(4), 401–415 (1982).

M. Ginsberg, "Non-Monotonic Reasoning Using Dempster's Rule," in *Proceedings of the Fourth National Conference on Artificial Intelligence*, Austin, Tex., AAAI, Menlo Park, Calif., 1984, pp. 126–129.

C. Glymour, "Independence Assumptions and Bayesian Updating," *J. Artif. Intell.* **25**, 95–99 (1985).

J. Y. Halpern and Y. Moses, "A Guide to Modal Logics of Knowledge and Belief," in *Proceedings of the Ninth International Conference on Artificial Intelligence*, AAAI, Menlo Park, Calif., 1985, pp. 480–490.

D. Heckerman, "Probabilistic Interpretations for MYCIN Certainty Factors," in Bonissone and Decker, 1986, pp. 167–196.

D. E. Heckerman, J. S. Breese, and E. J. Horvitz, "The Compilation of Decision Models," in *Proceedings of the Fifth AAAI Workshop on Uncertainty in Artificial Intelligence*, AAAI, Menlo Park, Calif., Aug. 1989, pp. 162–173.

M. Henrion, "Practical Issues in Constructing a Bayes' Belief Network," in Bonissone and Wood, 1989, pp. 161–173.

E. J. Horvitz, "Reasoning under Varying and Uncertain Resource Constraints," in Dean and Boddy, 1988, pp. 111–116.

E. J. Horvitz, H. J. Suermondt, and G. F. Cooper, "Bounded Conditioning Flexible Inference for Decisions under Scarce Resources," in Heckerman and co-workers, 1989, pp. 182–193.

R. Howard and J. Matheson, "Influence Diagrams," in R. Howard and J. Matheson, eds., *The Principles and Applications of Decision Analysis*, Vol. 2, Strategic Decisions Group, Menlo Park, Calif., 1984, pp. 719–762.

M. Ishizuka, "An Extension of Dempster-Shafer Theory to Fuzzy Sets for Constructing Expert Systems," *Seisan-Kenkyu* **34**, 312–315 (1982).

M. Ishizuka, K. Fu, and J. Yao, "A Rule-Based Inference with Fuzzy Set for Structural Damage Assessment," in M. Gupta and E. Sanchez, eds., *Fuzzy Information and Decision Processes*, North-Holland, Amsterdam, The Netherlands, 1982.

R. Johnson, "Independence and Bayesian Updating Methods," *J. Artif. Intell.* **29**, 217–222 (1986).

Y. Kasai and Y. Morimoti, "Electronically Controlled Continuously Variable Transmission," in *Proceedings of the International Congress on Transportation Electronics*, IEEE, 1988, pp. 33–42.

S. Lauritzen and D. Spiegelhalter, "Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems," *J. R. Stat. Soc. Ser. B* **50**, 157–224 (1988).

J. Lowrance and T. Garvey, *Evidential Reasoning: An Implementation for Multisensor Integration*, "Technical Report Note 307, SRI International, Artificial Intelligence Center, Menlo Park, Calif., 1983.

J. Lowrance, T. Garvey, and T. Strat, "A Framework for Evidential-Reasoning Systems," in *Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, Pa., AAAI, Menlo Park, Calif., 1986, pp. 896–903.

J. Pearl, "Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach," in *Proceedings of the Second National Conference on Artificial Intelligence*, Pittsburgh, Pa., AAAI, Menlo Park, Calif., August 1982, pp. 133–136.

J. Pearl, "How to Do with Probabilities What People Say You Can't," in *Proceedings of the Second Conference on Artificial Intelligence Applications*, IEEE, Dec. 1985, pp. 1–12.

J. Pearl, "Evidential Reasoning under Uncertainty," In H. E. Shrobe, ed., *Exploring Artificial Intelligence*, Morgan-Kaufmann, San Mateo, Calif., 1988a, pp. 381–418.

J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan-Kaufmann, San Mateo, Calif., 1988b.

E. Pednault, S. Zucker, and L. Muresan, "On the Independence Assumption Underlying Subjective Bayesian Updating, *J. Artif. Intell.* **16**, 213–222 (1981).

L. M. Pfau, *RUMrunner: Real-Time Reasoning with Uncertainty*, Master's thesis, Rensselaer Polytechnic Institute, Troy, N.Y., Dec. 1987.

H. Prade, "A Computational Approach to Approximate Reasoning and Plausible Reasoning with Applications to Expert Systems," *IEEE Trans. Pattern Anal. Machine Intell.* **7**(3), 260–283 (1985).

J. Quinlan, "Consistency and Plausible Reasoning," in *Proceedings of the Eighth IJCAI*, Karlsruhe, FRG, Morgan-Kaufmann, San Mateo, Calif., 1983, pp. 137–144.

R. Reiter, "A Logic for Default Reasoning," *Artif. Intell.* **13**, 81–132 (1980).

E. Rich, "Default Reasoning as Likelihood Reasoning," in *Proceedings of the Third National Conference on Artificial Intelligence*, AAAI, Menlo Park, Calif., 1983, pp. 348–351.

C. Rollinger, "How to Represent Evidence—Aspects of Uncertainty Reasoning, in Quinlan, 1983, pp. 358–361.

E. Ruspini, *The Logical Foundations of Evidential Reasoning*, Technical Note 408, Artificial Intelligence Center, SRI International, Menlo Park, Calif., 1987.

E. Ruspini, *On the Semantics of Fuzzy Logic*, Technical Note 475, Artificial Intelligence Center, SRI International, Menlo Park, Calif., 1989a.

E. Ruspini, "The Semantics of Vague Knowledge," *Rev. Syst.* **3**(4), 387–420 (1989b).

E. Ruspini, "Possibility as Similarity: The Semantics of Fuzzy Logic," in *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, Cambridge, Mass., 1990, pp. 281–289.

R. Schachter, "Evaluating Influence Diagrams," *Operations Res.* **34,** 871–882 (1986).

B. Schweizer and A. Sklar, "Associative Functions and Abstract Semi-Groups," *Publicationes Mathematicae Debrecen* **10,** 69–81 (1963).

B. Schweizer and A. Sklar, *Probabilistic Metric Spaces.* North-Holland, Amsterdam, The Netherlands, 1983.

G. Shafer, *A Mathematical Theory of Evidence,* Princeton University Press, Princeton, N.J., 1976.

E. Shortliffe and B. Buchanan, "A Model of Inexact Reasoning in Medicine," *Math. Biosci.* **23,** 351–379 (1975).

P. Smets, "The Degree of Belief in a Fuzzy Set," *Inform. Sci.* **25,** 1–19 (1981).

P. Smets, "Belief Functions," in P. Smets, A. Mamdani, D. Dubois, and H. Prade, eds., *Non-Standard Logics for Automated Reasoning,* Academic Press, Inc., New York, 1988.

J. Stillman, "On Heuristics for Finding Loop Cutsets in Multiply-Connected Belief Networks," in *Proceedings of the Sixth Conference on Uncertainty in AI,* 1990, pp. 265–272.

T. Strat, "Continuous Belief Functions for Evidential Reasoning," in Ginsberg, 1984, pp. 308–313.

J. Suermondt, G. Cooper, and D. Heckerman, "A Combination of Cutset Conditioning with Clique-Tree Propagation in the Pathfinder System," in *Proceedings of the Sixth Conference on Uncertainty in AI,* 1990, pp. 273–279.

M. Sugeno, ed., *Industrial Applications of Fuzzy Control,* North-Holland, Amsterdam, The Netherlands, 1985.

H. Watanabe and W. Dettloff, "Fuzzy Logic Inference Processor for Real Time Control: A Second Generation Full Custom Design," in *Proceedings of the Twenty-first Asilomar Conference on Signal, Systems & Computers,* IEEE, Nov. 1987, pp. 729–735.

S. Yasunobu and G. Hasegawa, "Evaluation of an Automatic Crane Operation System Based on Predictive Fuzzy Control," *Contr. Theor. Adv. Technol.* **2,** 419–432 (1986).

S. Yasunobu and S. Miyamoto, "Automatic Train Operation by Predictive Fuzzy Control," in Sugeno, 1985, pp. 1–8.

L Zadeh, "Fuzzy Sets," *Inform. Contr.* **8,** 338–353 (1965).

L. Zadeh, "Fuzzy Logic and Approximate Reasoning (in Memory of Grigor Moisil)," *Synthese* **30,** 407–428 (1975).

L. Zadeh, "Fuzzy Sets as a Basis for a Theory of Possibility," *Fuzzy Sets Sys.* **1,** 3–28 (1978).

L. Zadeh, "Fuzzy Sets and Information Granularity," In M. Gupta, R. Ragade, and R. Yager, eds., *Advances in Fuzzy Set Theory and Applications,* Elsevier Science Publishing Co., Inc., New York, 1979, pp. 3–18.

L. Zadeh, "A Theory of Approximate Reasoning," in P. Hayes, D. Michie, and L. Mikulich, eds., *Machine Intelligence,* Halstead Press, New York, 1979, pp. 149–194.

L. Zadeh, "A Computational Approach to Fuzzy Quantifiers in Natural Language," *Comput. Math.* **9,** 149–184 (1983).

L. Zadeh, "Linguistic Variables, Approximate Reasoning, and Dispositions," *Med. Inform.* **8,** 173–186 (1983b).

L. Zadeh, "A Computational Theory of Disposition," in *Proceedings of the International Conference of Computational Linguistics,* 1984a, pp. 312–318.

L. Zadeh, "Review of Books: A Mathematical Theory of Evidence," *AI Mag.* **5**(3), 81–83 (1984b).

L. Zadeh, "Syllogistic Reasoning in Fuzzy Logic and Its Application to Usuality and Reasoning with Dispositions," *IEEE Trans. Syst. Man Cybernet.* **15,** 754–765 (1985a).

L. Zadeh, *A Simple View of the Dempster-Shafer Theory of Evidence and Its Implications for the Rule of Combinations,* Technical Report 33, Institute of Cognitive Science, University of California, Berkeley, 1985b.

L. Zadeh, "Dispositional Logic," *Appl. Math. Lett.* **1**(1), 95–99 (1988).

P. Bonissone
General Electric

REASONING, SPATIAL

Human beings and other creatures spend much of their time solving spatial problems, such as finding their way around. Furthermore, people often seem to use spatial methods for solving problems analogically, as when they reason about graphs by drawing or imaging pictures. Research in this area tries to duplicate or mimic some of these abilities. Tentatively, it can be divided into these subheadings:

Visual object recognition.
Cognitive maps and path finding.
Simulations of human imagery.
Visualization for qualitative physical reasoning.

It seems a good guess that all of these problem areas share representations and algorithms. However, to date most of them have evolved in different directions. Not all of these areas are covered here. In particular, visual object recognition is omitted entirely. Surveys of research in this area have been published (Kak, 1988; Chen, 1990).

A key issue in spatial reasoning is qualitative shape representation. Most humans have little trouble visualizing objects and reasoning about them without precise knowledge about their dimensions. For instance, suppose a balloon landed on a pincushion. What might happen? Although the pins might puncture the balloon, it is quickly realized that they are unlikely to in this case because they are head side up. When most people solve a problem like this, they are obviously unaware of the exact shape of the pincushion or the detailed distribution of the pins. Yet they imagine a "picture" of the situation. Controversy has raged about what is really going on in the mind when this picturelike entity is experienced (see below). Fortunately for AI, the computational question can be asked how the knowledge about the shapes of objects like pins and pincushions is represented and used without an *a priori* commitment to any answer to questions about human visual imagery.

Various proposals have been made about representation of spatial information. Many of them limit themselves to two dimensions instead of three, either as a research tactic or because of a belief that it is desirable for efficiency to reduce three-dimensional problems to two dimensions when possible. Shape representations tend to fall into various categories: part whole, volumetric, and surface descriptions. Overviews are available (Ballard and Brown, 1982; Davis, 1990).

Part–Whole Descriptions

Objects are described in terms of the parts that make them up. Typically these descriptions employ some kind of associative network. There is nothing special about this use of associative networks; the resulting descriptions would be similar to those in nonspatial domains, such as descriptions of corporate organizations. So the pincushion description might mention the presence of zero or more pins as parts.

The representations becomes more spatial when coordinates and other parameters are added to it. For instance, with each pin might be stored its approximate length and position with respect to the pincushion. It is often useful to invert the resulting data structure. For instance, given a table of cities and their locations, it might be desirable to find a city near some location. Rather than search all the cities, it is possible to use a discrimination tree(2), in which objects are sorted by discriminating on their X and Y coordinates (Fig. 1). Other quantitative and symbolic discriminators can be introduced, such as the population or shape of each object. [The term "*k*-d tree" has been used by Bentley and Friedman (1979) for a tree discriminated on several numerical coordinates.] To find an object in such a tree, given as a key an X interval and a Y interval, the computer can start at the top and follow only branches compatible with the key intervals, so only a subset of the cities are ultimately compared with the key interval.

Volumetric Descriptions

Objects are described as combinations of volumes. The volumes are often overlapping and often do not necessarily designate distinct parts of the overall shape. For example, a milk bottle might be described as a cylinder topped by truncated cone. The dimensions and relative locations and orientations of the volumes must be specified (Marr and Nishihara, 1978; Brooks, 1981). The component volumes may be drawn from a vocabulary of primitives or constructed by sweeping surfaces along axes (so-called generalized cylinders) (Agin and Binford, 1976). In some systems volumes may be subtracted as well as added. For example, a spool might be described as a solid cylinder with a small parallel cylinder subtracted from its axis. This approach is often called constructive solid geometry (Requicha, 1980).

One of the simplest ways of describing volumes (or, in two dimensions, areas) is with arrays representing space (up to some grain), whose cells are labeled with the object filling the part of space. These are often called *occupancy arrays* and their cells are called *pixels* in two dimensions (a term borrowed from computer graphics), or *voxels* in three dimensions. In Figure 2a the shape of a house is represented by putting an H in every cell the house fills. This is a two-dimensional projection; a three-dimensional array could be used if required. However, even the two-dimensional version is costly. The quadtree device allows the information to be compressed in an elegant way (Klinger and Rhodes, 1979; Samet, 1984). The grid is represented as a tree whose nodes are square areas of the picture. The top node represents the entire picture. Each node has zero or four children. If the square region corresponding to a node lies entirely inside or outside the house, it is a leaf of the tree and is labeled with an H or E (for empty). Otherwise, it has four children representing the four quadrants of its square. The division stops at some convenient grain (Fig. 2b). The same ideas applied to three dimensions yields the octree. A natural extension of the occupancy-array idea is to store a more complex vector of data at each pixel. For example, in a military application, the elevation, slope, and vegetation might be stored at each pixel (Antony, 1990; Thorpe and co-workers, 1988).

Another interesting generalization was proposed by Moravec (1988). Instead of classifying each pixel as occu-

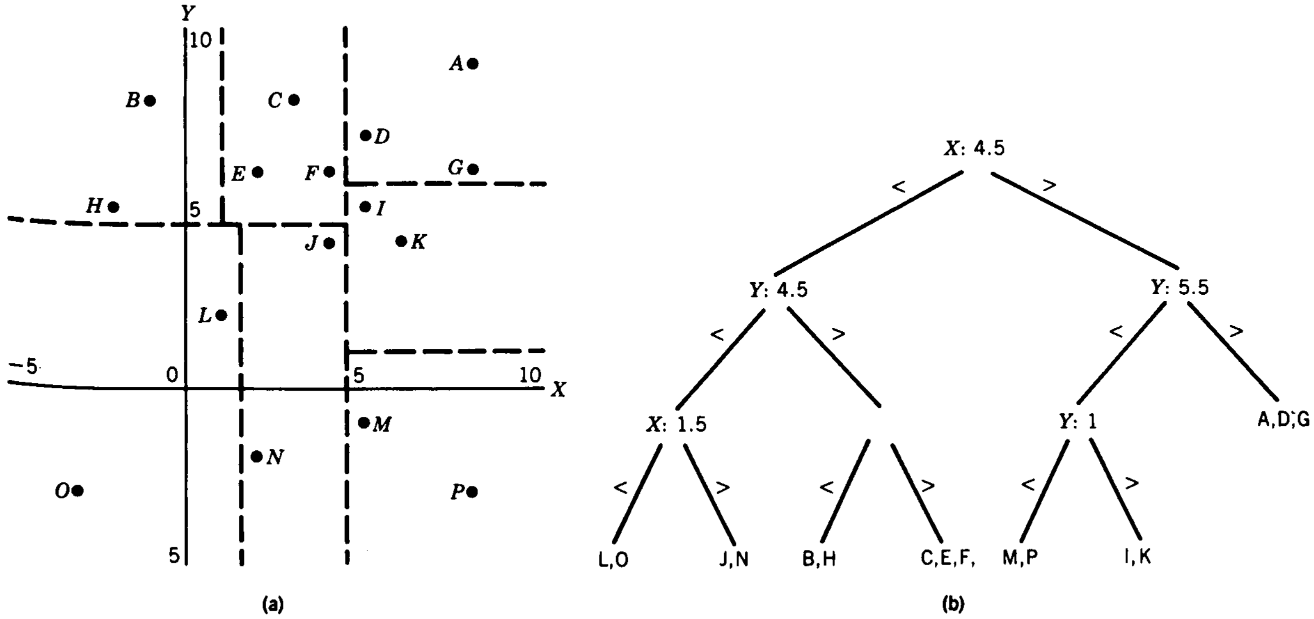


Figure 1. An X—Y discrimination tree: (*a*) cities; (*b*) discrimination tree.

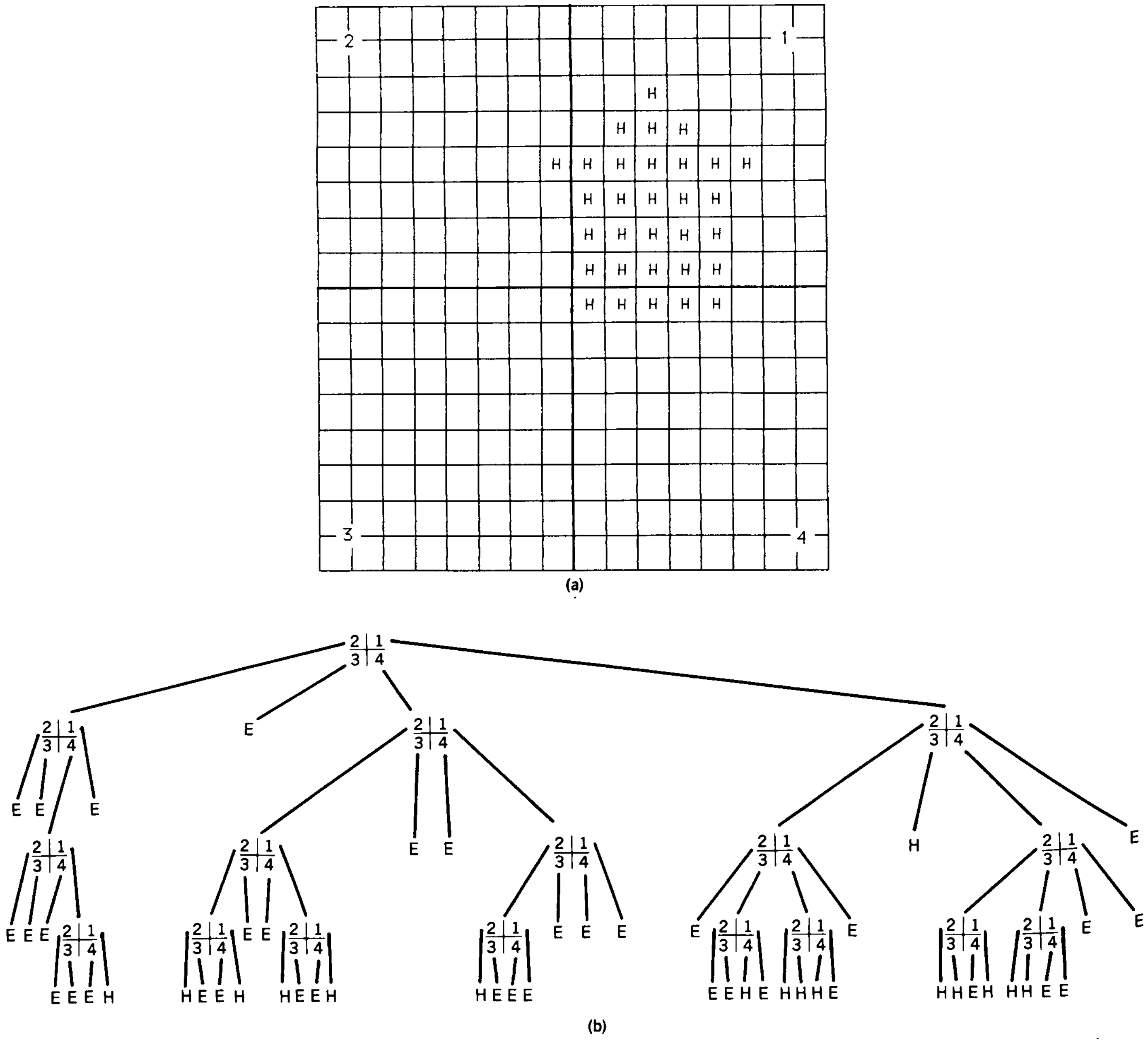


Figure 2. Grid representation and quadtree: (*a*) occupancy array; (*b*) corresponding quadtree.

pied or not, the probability of its being occupied is stored. The resulting representation is called a certainty grid. The probabilities are derived from sensor data. New sensor readings are combined with old using Bayesian techniques. Data obtained at different levels of resolution can be stored using different pixel densities. For example, areas farther from the sensor would normally be broken into a few large pixels, whereas areas nearby could be broken into many small ones.

Boundary Description

Objects are described by their bounding surfaces. The bounding surfaces are often planar, which limits the description to polyhedral objects or to polyhedral approximations of curved objects (Baumgart, 1975). In two dimensions the bounds are curves, often approximated as polygons or "polylines" (chains of lines, not necessarily closed curves) (Davis, 1985).

Many of these shape-representation ideas are borrowed from machine vision and graphics. If the goal is to draw a picture of an object, the application of these representations is well understood. Unfortunately, for the kind of qualitative spatial reasoning discussed above, much of the information provided in these formats is useless, and ideas have been lacking on what is needed instead. For instance, devices such as cubic splines are excellent for parameterizing a curve or surface with just a few numbers. However, the resulting data compression is of little use in spatial reasoning because the numbers say little about the way the surface behaves. In spatial reasoning it is more likely that information such as *can be used as a conduit* or *is almost horizontal* is desired. It is often necessary or desirable to approximate spatial knowledge. If a wall is almost flat, for many purposes there is no need to keep its slight deviations from flatness in mind, even if they are known. Hence the model used by the system for reasoning will usually be a simplified version of the truth. There may be multiple models for different purposes.

The most common approach to this problem is to use a simple symbolic vocabulary. If an object is known to be a cylinder, and nothing else is known about it, then it is described by the symbol *cylinder.* Usually the machine knows something about the dimensions of the object (Brooks, 1981), so it might be represented as

```
cylinder
   length > width
   axis-curvature = 0
```

This idea can be thought of as sorting objects into qualitative "bins" with labels like *cylinder.* Within each bin, objects are distinguished by different values of parameters such as length and axis curvature. This approach is by far the most common; for specialized applications, such bins are usually easy to find. For instance, in a route-finding application objects might be classified as streets, buildings, regions, rivers, etc, each with its own set of parameters. The boundaries between classes would seldom be crossed. The classes are useful because there is a large set of inferences that pertain to just the objects belonging to a given class. For instance, an object classified as a street can be used to get somewhere; an object classified as a river requires finding a bridge, etc.

The difficulty with this scheme is generalizing it to handle more than one application area. In the general case, the following problems are encountered.

1. The qualitative-bin notation has trouble with detailed descriptions of objects. Once an object has been classified as a cylinder, it may be necessary to describe it further as "slightly flattened on one side" or "peppered with thousands of holes." Little is known about how to turn such natural language descriptions into something more formal.

2. An object may fall into more than one bin. The big advantage of the qualitative-bin idea is that similar objects have similar representations. The two objects in Figures 3a and 3b are obviously similar because they are both classed as cylinders with different axis curvatures. But if the sequence of similar objects is continued, this ultimately leads to Figure 3d, which would have been classed as a torus (with a gap represented somehow). Such qualitative discontinuities may make it necessary to maintain multiple descriptions of objects.

3. An object may fall into no bins. For any given application it is usually easy to find qualitative classes that include every object of interest. It is much harder to find a set of classes that works just as well for every application. If you simply take all the classes that have ever been proposed, many objects will fall into more than one or none at all.

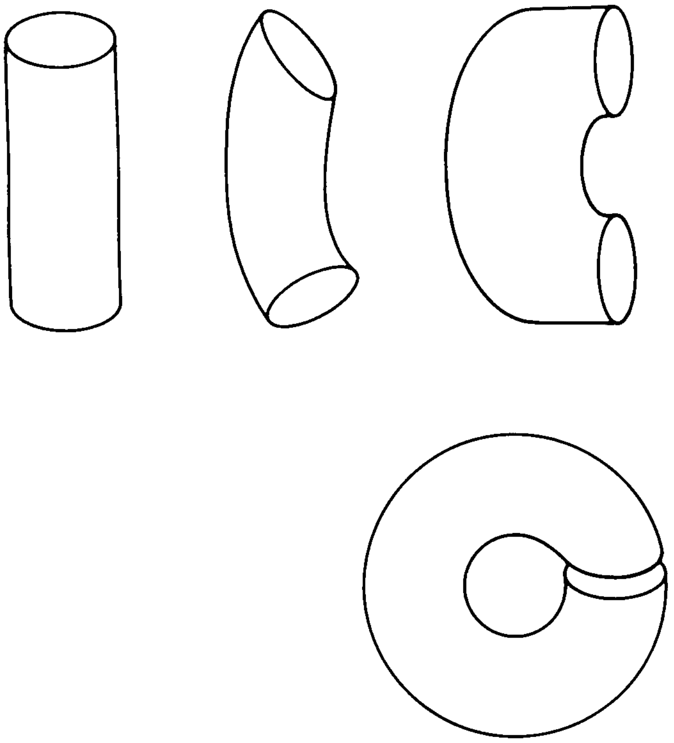


Figure 3. Object with multiple descriptions.

All these problems may be evidence that there is no solution to the general spatial-reasoning problem. As elsewhere in AI, the mere ability to discern a problem may not mean it actually has a solution. Still, it is hard on introspective grounds to believe that humans have a collection of independent task-oriented spatial representations in their heads, and this is reason to keep looking for a solution to the general case.

In what follows, various specialized problem areas are examined, keeping some of these problems in mind.

ROUTE FINDING AND EXPLORATION

The route-finding problem is planning a route from one place to another and then following it. It assumed that the planner (henceforth referred to as the robot) has a cognitive map of its surroundings that it can consult for this purpose. This map is incomplete, so that the planner may need to ask directions or explore as it goes. In this section it is assumed that the robot is small compared to the spaces through which it is navigating. A somewhat different line of research assumes that the robot is large but that the shapes and positions of all obstacles in the space are known so that intricate reasoning is required to squeeze it through. This is called the robot motion-planning problem (see VISUAL MOTION ANALYSIS). The assumption is also made that the robot is moving on a two-dimensional surface. The objects it sees are assumed to approximately prismatic and perpendicular to this surface, so that they can be described by specifying their cross sections and heights.

There are several approaches to the route-finding problem, based on quite different assumptions about what the problem is. There is no obvious definition of what a *place* is; it cannot be taken to be a point in Cartesian space. The coordinate system for such a point would be underdetermined, although in many applications there is an obvious choice. A more basic problem is that a place must be bigger than a single point, both because space is intuitively divided up that way and because otherwise it would be impossible for a robot to visit the same place twice. Another problem, usually simply neglected, is that some places, such as the interiors of airplanes and elevators, do not correspond to a fixed location with respect to a larger coordinate frame.

One solution to these conundrums is to define places in terms of stable perceptions (Kuipers and Byun, 1988; Kuipers and Levitt, 1988). A place is an area within which some perceptual invariant is preserved, such as "I see four corridors at 90° angles." This definition requires some refinement, because two places can look identical and because the same point in space can change its appearance over time. Another solution is to impose a grid on a coordinate system, so that a place is defined as an arbitrary square region (Moravec, 1988). This approach is related to the occupancy-array representation described above. A third approach is to put places aside and focus on the locations and orientations of identifiable objects in the region of interest (Chatila and Laumond, 1985; Smith and Cheeseman, 1986). Places can then be picked out if necessary by relating them to these objects. A place might be defined as "the region between two known walls."

A fourth "approach" is to let places be whatever humans find natural to designate as places, such as "Apartment 3G at 200 York Street," "The vacant lot on my block," or even "Highway 61." ("Abe said, Where you want this killing done? God said, Out on Highway 61," Bob Dylan.) This idea says little about how to carve up space into places, but it does raise the issue of why carve it up at all. If the task is to have a robot get orders from humans to go somewhere and do something, then at some point the robot must have matched up human labels with its own percepts. The alternative is to assume that the robot has its own goals that require it to know its way around, but so far the only goal studied is simply to learn the map, which leaves the choice of one definition of place versus another somewhat unjustified. In the remainder of this section, navigation methods will be examined using the various representations and then methods for learning maps.

Grid-based navigation methods are the most straightforward. Here it is assumed that the robot has methods for getting its approximate global position and orientation before it examines its surroundings perceptually. For example, in a military application, it can be assumed that satellites or airplanes can provide a rover with information about where it is in a terrain that is already well mapped to some resolution. It can further be assumed that the robot's objective is given with respect to the same global coordinate frame.

The route-finding problem in a grid-oriented representation is to plot a path from one pixel in the grid to another. A natural approach is to use the A*, or best-first search, algorithm (Hart and co-workers, 1968), treating a single-pixel move as an operator, and straight-line distance as a heuristic estimator (see A* ALGORITHM; SEARCH, BEST-FIRST). Figure 4 shows an example somewhat schematically. Occupied, untraversable pixels are shaded. A line from the center of one pixel to the center of an adjacent pixel indicates a single-pixel operator application. There are several little detours branching off the final path found, but the straight-line distance estimate keeps the algorithm on track fairly well. It does worse when it has to back out of cul de sacs, because the estimator is least accurate in those situations. The A* approach has the advantage of being simple and adaptable to a variety of situations. In any particular context, a more efficient algorithm can be found. A survey of a wide variety of such algorithms has been published (Mitchell, 1988).

Usually other criteria are added to the estimator so that something other than the shortest path is found. Moravec (1988) describes an algorithm for finding such a route in a certainty grid, where the criteria include minimizing the probability of encountering an obstacle. In a military context, the occupancy grid can be used to store information about roads and vegetation, and a path can be sought that minimizes opportunities for interdiction by the enemy.

Another context in which good positional information is known is a stored street or highway map. In this case, places are taken to be addresses or intersections, and it is assumed that the map has been completely inputted before route finding begins. Elliot and Lesk (1982) describe an *A**-style algorithm that provides automatic directions to places in a city. The heuristic evaluation function favors large streets and penalizes routes containing many left turns so that the directions it finds tend to be the sort humans like to follow.

The frequent use of best-first-search algorithms in metric domains is no accident, because the estimated straight-

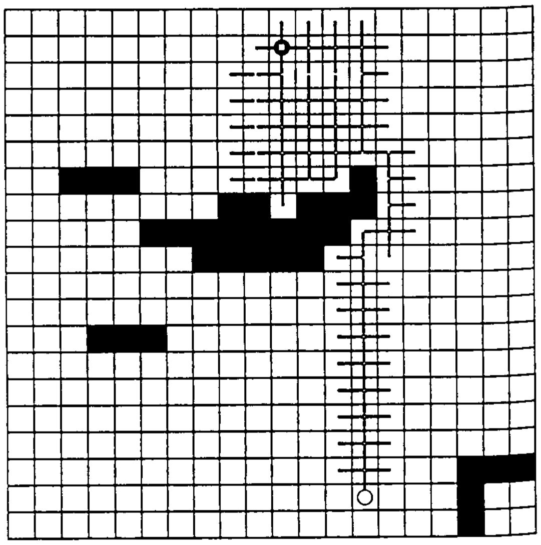


Figure 4. *A** search applied to route finding.

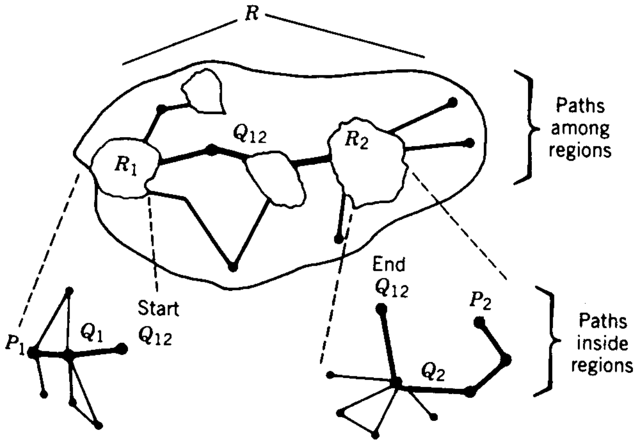


Figure 5. Finding routes between points in different regions.

line distance to a destination is such a good heuristic path-cost estimator, except in mazes (Lumelsky and Stepanov, 1987). This assertion remains true even when only approximate metric information is available (McDermott and Davis, 1984). However, in cases where metric information is assumed to be almost completely absent, other search strategies come to the fore, including searching through hierarchically organized graphs. The TOUR program (Kuipers, 1978) operates in the context of a city street network. The network is incomplete, and there are no global coordinates stored for anything. It maintains a hierarchical structure of regions. Paths are located inside regions and can run between regions. This structure makes it natural to find hierarchical plans for getting from one place to another (a hierarchical plan for a task consists of a short sequence of large steps, each of which is broken down into smaller plans if necessary). Here is a sketch of an algorithm for finding such plans (Fig. 5):

```
To find a path between two points $P_1$ and $P_2$:
    Find the smallest region $R$ including both points.
    Find the regions $R_1$ and $R_2$ just below $R$ that contain
        $P_1$ and $P_2$, respectively.
    Find a route $Q_{12}$ from $R_1$ to $R_2$ (by some search process).
    Recursively find a route $Q_1$ from $P_1$ to $Q_{12}$ in $R_1$ and
        a route $Q_2$ from $P_2$ to $Q_{12}$ in $R_2$.
    Return result $Q_1 - Q_{12} - Q_2$.
```

This algorithm is not guaranteed to return optimal routes, but it works without requiring any sort of global coordinates. All it requires is that the area actually be organizable into appropriate hierarchical structure of major routes between fairly well-defined regions.

Attention is now turned to the problem of exploring and learning an unknown area. As mentioned, it is possible to distinguish between perceptual approaches, in which the goal is to discern and relate places defined by perceptual invariants, and metric approaches, in which the goal is to discover objects and learn their shapes, positions, and orientations.

A survey of the first kind of approach is available (Kuipers and Levitt, 1988). The perceptual technique is often coupled with the assumption that the overall structure of the cognitive map is a graph whose edges correspond to paths and whose nodes correspond to places (Kuipers and Byun, 1988). Sometimes the paths are given by a road network, but in less structured domains paths can be defined in terms of robot control strategies. For example, if a robot can follow a wall, then a large open room might be organized as a set of paths around its walls, punctuated by places at the corners and doors. This picture makes the most sense when the robot's sensors and effectors are assumed to be reliable only over short distances, so that it has little hope of being able to make sense of its location after launching itself through the interior of the room.

Under such short-range assumptions, which are reasonable with today's robots, places can be defined as areas satisfying some distinctiveness criterion. For example, a cul de sac might be defined as a place where most of the viewing angles around the robot are occupied by solid material. A corridor would debouch on a room at a place where there is a 180° open area on one side and two nearby solid peaks on the other. If this distinctiveness criterion can be quantified, as a distinctiveness measure, then the center of a distinctive neighborhood can be reached by moving until a local maximum of distinctiveness is reached. Doing this hill climbing allows the robot to attain a canonical location from which to launch subsequent explorations.

In one model (Kuipers and Byun, 1988) exploration requires moving out into open space from the current place, then adopting an appropriate control strategy to move along a path to the next place. Eventually, in a closed world, the robot will come back to a place it has already visited, which will satisfy the same distinctiveness criterion as before. However, it is entirely possible that two places could look very similar, so some care must be taken in deciding whether to identify the current place with one seen earlier. Kuipers and Byun use the following heuristic for deciding whether two places, P_{old} and P_{new}, are the same. Because the robot has been to P_{old} and left, it knows what some neighboring places look like. Hence it can attempt to traverse a path to those neighbors. If it gets to places that look right, it assumes P_{old} and P_{new} are the same place, and builds the graph accordingly. If it gets to places that don't match, it treats P_{old} and P_{new} as unrelated places. This strategy requires being able to identify directions out of P_{new} with those out of P_{old}, which depends on either a global compass or an asymmetrical distinctiveness criterion. The procedure is not foolproof, both because the directions might fail to match up properly, and because two similar places might have neighboring places that look similar. The first problem would lead to failure to realize that P_{old} and P_{new} were the same; the second problem would lead to the opposite error. The term *identification problem* is used to refer to the possibility of these kinds of error.

Another perceptual technique has been developed (Levitt and co-workers, 1987) that is based on the assumption that a robot can reliably track landmarks at a distance. Over a given time interval, a stable set of such landmarks (towers, tall buildings, mountains) will be perceivable around the robot, in a stable order. Whenever two landmarks are simultaneously visible, the robot can know when it has crossed the line between them, called a *landmark pair boundary* (LPB). It can tell which side it is on by whether, in panning clockwise, it sees landmark 1 before landmark 2. Places can be defined as minimal polygons bounded by LPB, and routes can be plotted by running an A^* algorithm through the resulting tesselation of the plane. Currently the theory does not specify how landmarks are first noticed or how they are reliably tracked.

With metric approaches to route finding and exploration, the map-learning problem is to sort the world into objects and locate the objects in space. In special cases, today's sensors allow very impressive performance simply by sensing a scene repeatedly from a moving robot, generating a series of overlapping models, and matching up successive models. The NavLab project (Thorpe and coworkers, 1988), can create a map hundreds of meters long and a hundred meters wide by this method. The map is represented as a grid giving the elevation at discrete x, y coordinates over a wide area, and marking some pixels as containing obstacles. The method depends on using a laser range finder to generate high quality elevation maps and an inertial guidance system to help match up successive scenes.

A more general solution to the metric map-learning problem would rely more on passive vision, and would have to cope with uncertainty of robot position. It would also have to solve the problem of recognizing a familiar group of objects when they are approached from a new direction. First, it is necessary to represent the shapes of the objects. The usual approach is to classify each object as a standard shape (eg, a rectangle) with uncertain numerical attributes (eg, its length and width). A more ambitious representation has been used (Davis, 1985), which allowed objects to be approximated by arbitrary polygons whose sides were of uncertain length and joined at uncertain angles. Second, it is necessary to manage all this uncertainty. Typically, the task is to keep track of the intervals within which quantities lie (McDermott and Davis, 1984) or of probability distributions for those quantities (Smith and Cheeseman, 1986; Chatila and Laumond, 1985; Moutarlier and Chatila, 1989). As new information is gathered, the probability distributions become sharper (Fig. 6).

A fundamental question is what coordinate system to use for quantities like the x coordinate of an object. Even when there is an obvious coordinate system to use, there is the problem of capturing precise local information. For instance, the robot might know that objects A and B are close together without knowing much about their location in the global coordinate system. The approach taken by most probabilists is to store the probability distributions as means and variances (ie, to assume that they are Gaussian) and then to store a covariance matrix for all pairs of quantities in order to capture local relationships. Estimation theory supplies methods for updating these matrices as new observations occur. The alternative (McDermott and Davis, 1984) is to maintain many local coordinate systems, so that B's position could be stored with respect to A's frame. Obtaining B's position with respect to the global frame might then require composing the information about its position with respect to A and information about A's position with respect to the global frame. The fact that all these quantities are stored as intervals make the computations rather messy.

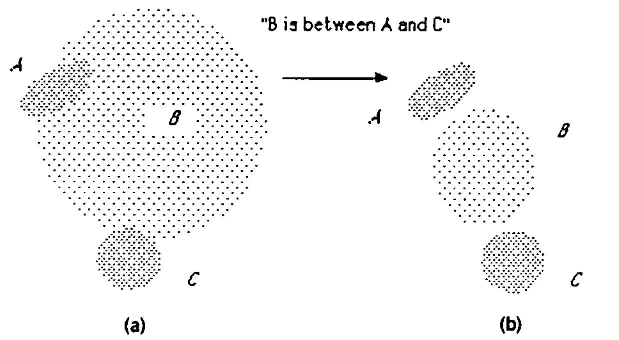


Figure 6. New information sharpens probability distributions on coordinates: (a) before new information; (b) after.

It is possible to accumulate information about the parameters of an object only if the object can be reliably recognized when encountered. This is just the identification problem again, which arose with perceptual approaches in connection with deciding when to equate perceptually similar places and landmarks. Many workers in the field assume that recognizing previously seen objects is a job for the vision module. Others assume that the current scene can be converted into a local map, which can then be merged into a global map, by searching for the piece of the global map that matches it best. Elfes (1989) described algorithms for matching certainty grids. Davis (1985) described an algorithm for matching polygonal maps, where the polygons are required only to approximate underlying shapes. Lavin (1979) developed one for a more specialized representation of Gaussian hills.

One issue that has received remarkably little attention is how to correct errors in learned maps. Once a distortion gets introduced, future attempts to make new information consistent with it tend to introduce worse and worse distortions. A discussion of some of the problems is available (Davis, 1988). It is hoped that probabilistic representations could tolerate errors by having later observations overwhelm them, pushing parameters back toward correct values, but that hope runs into the problem that parameter-value errors can make matching errors more likely, causing the system's map to get further and further from the truth.

PHYSICAL REASONING

An active area of AI research is the study of reasoning about the structure and function of physical systems. Structure inevitably includes spatial structure. A key problem type for algorithms developed in this area is: given a mechanism, what will it do? In many cases, physical-reasoning algorithms are expected to take qualitative descriptions of systems and draw qualitative conclusions about their behavior. For example, given the signs of initial values of quantities, infer whether the values ever become zero (de Kleer and Brown, 1984; Forbus, 1984) (see PHYSICS, QUALITATIVE).

Unfortunately, when spatial structure is important, it is hard to find cases where it is possible to infer much without knowing the details of the initial layout of a system. Hence, especially in recent research, it has been conceded that detailed quantitative knowledge of the shape and initial configuration of a mechanism are needed, even if all that is wanted is a qualitative description of its behavior over time.

When people solve spatial-reasoning problems in their heads, they often have a subjective feeling of seeing a picture. For example, if asked to name all 50 states, almost everyone reports visualizing a path through which the attention wanders. There is a controversy about what these reports are reports of. On one side are researchers such as Shepard and Metzler (1971) and Kosslyn (1980, 1983) who believe that there is an actual picturelike entity in the brain performing useful computations. On the other are critics such as Pylyshyn (1985) who believe that pictures are poor computing devices and that subjective impressions are misleading.

The first issue to settle in building a computational model of imagery is what the underlying picture medium is. It is possible to begin by assuming that a map of the United States was stored as a hierarchical pointer structure, with nodes representing large areas (like New England) pointing to smaller component areas (like states) and other, less familiar, areas pointing to a sparser set of subareas. Such a model might explain many facts [eg, certain distortions in subjects' memories of maps (Stevens and Coupe, 1978)], but would not itself explain why the data structure is experienced as a picture when it is traversed. Researchers who believe that this is one of the prime facts to be explained make their models pictorial from the start. A dangerous pitfall here is to assume that a mental image is nothing but a picture, so that some homunculus must "look" at it. To avoid the need for the homunculus, image theorists usually assume that the pictorial medium is active, capable of computation on its own.

There have been several attempts to avoid the need for detailed quantitative knowledge of spatial layouts, notably the work of Hayes (1985a, 1985b) on naive physics (See PHYSICS, NAIVE), the effort to formalize what "everyone knows" about the way objects in the world behave and interact. He presents axioms about the behavior of liquids in containers, where the containers are described in terms of their bounding surfaces. The notation omits most of the details of where these surfaces are located or how they are shaped and instead focuses on qualitative description of them as separating volumes into different functional parts. An open container, for instance, is described as a volume with just one free face, the top. The brim of the container is the "face of its top face," that is, the edge bounding its top face. Hayes argues that the way to bring time and change into this formalism is to analyze activity as four-dimensional histories of objects. For example, if an open container full of liquid is tilted, there will be a "leaving" history at the brim of the container that interfaces to a "falling" history in the free space nearby. A more recent attempt to axiomatize knowledge about space and change is available (Davis, 1990).

In practical programs, more specialized solutions often obviate a detailed representation. Figure 7 shows a problem solved by deKleer's (1979) program NEWTON. The program reasons about the qualitative shape of the roller coaster in order to realize that the object cannot get to point X. Other physics-problem solvers reason about situations of similar complexity (Novak, 1977; Bundy, 1978).

None of these programs rely on a general-purpose shape representation. All of them use some version of the qualitative-bin representation. For example, Novak's program accepts physics-problem statements in natural language. It turns a sentence like "a man stands on a ladder" into an internal representation in which a man modeled as a point mass is located on a ladder represented as a line. Each such internal entity has various parameters associated with it, such as the man's mass. In different problem statements the man would get modeled as some other kind of entity, with different parameters. Equations involving these parameters are set up and solved to produce the solution to the overall problem. The concern about these representations is that they presuppose so much about the problem class. At no point does the program possess a neutral statement of the geometric setup, which is independent of the question type to be asked. It is hard to imagine a purely qualitative representation of this kind.

The difficulty of finding an effective qualitative spatial representation has led researchers to assume that good quantitative information is available about the shapes and positions of objects. The problem then is to extract reasonable predictions about behavior from these initial conditions, where "reasonable" is hard to define, but tends to mean "concise and symbolic," as opposed to voluminous and numerical.

One way to get a quantitative representation of spatial change over time is to use a changing spatial-occupancy array. Each pixel now records its current state, and can change to a new state after communicating with its neighbors. Interesting inferences then occur by the combined action of all the pixels. This model is quite attractive to imagery theorists, because in the brain a natural imple-

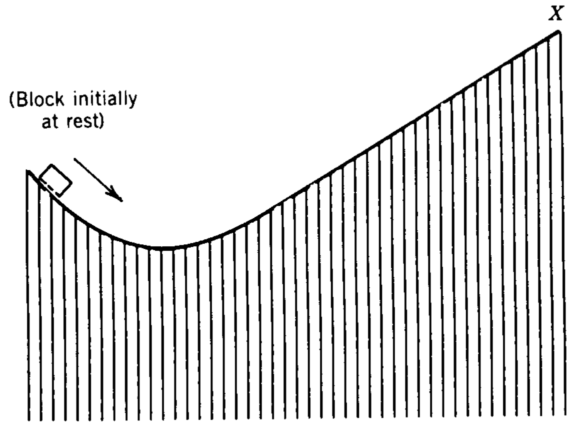


Figure 7. Problem solved by NEWTON: "Will the block reach point X?" (de Kleer, 1975).

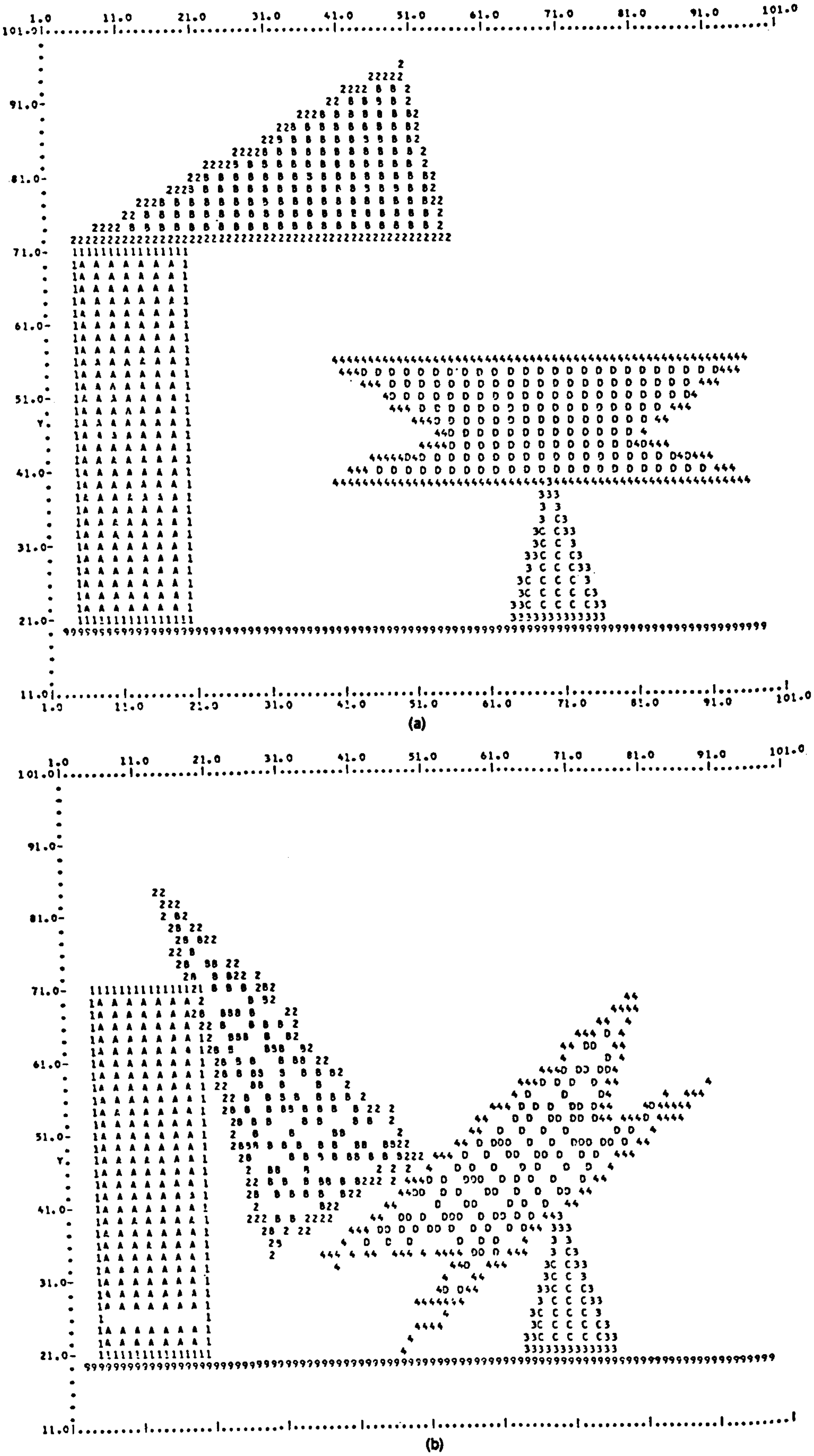


Figure 8. Use of blocks-world image to reason about stability: (*a*) initial snapshot; (*b*) final snapshot.

mentation would involve groups of neurons, probably laid out in an actual two-dimensional field. In a digital computer each pixel is a data structure arranged in an array, and a sequential algorithm simulates each in turn; but a special-purpose computer could be built that implemented the pixels as separate pieces of silicon, all running in parallel. If this could be made to work, it might be very fast.

An example of this kind of system is Funt's (1976, 1987) program WHISPER, which solves problems in the "blocks world," (Fig. 8). Block A is indicated by labeling the pixels it occupies with an A for pixels in A's interior, or a 1 for pixels on A's boundary. In that figure the program decides that block B is unstable and will cause D to fall when it does. The image machine helps in several ways. First, the center of gravity of B must be found, by having each pixel labeled B or 2 report its location to a central processor, which adds the coordinates and averages. The processor notes that the center of gravity of B is not above A, so B will fall. Then it simulates the fall by instructing each pixel of B to rotate around the upper right corner of A. This rotation occurs step by step [actually, two arrays of pixels are used (Funt, 1987)]. The collision between B and D is noticed when a 2-labeled pixel attempts to transfer its contents into one already labeled with a 4. The advantage of this approach to thinking about colliding blocks is that no intricate calculation about intersecting lines is required to detect a collision. A more recent example, involving simulating a wider array of physical phenomena is that of Gardin and Meltzer (1989).

Other spatial representations can be set in motion besides occupancy arrays. An active area of research is in modeling mechanisms, where the ubiquity of curved surfaces make volume and boundary descriptions more appropriate than occupancy arrays. The mechanism-envisioning problem involves starting with a detailed description of, say, a clock, and producing a "qualitative" description of its behavior. Because this work is still in an exploratory phase, it has encountered a common obstacle in AI: not being able to say exactly what the output is to be used for. Some researchers (Faltings, 1990) attempt to produce a classification of the possible positions the mechanism can be in. Others (Gelsey and McDermott, 1990) produce a symbolic description of some traces of machine behavior for typical initial conditions, without attempting an exhaustive analysis of the machine's possible configurations in advance.

The study of possible configurations is called *machine kinematics,* and, of course, has been carried out by human engineers for decades (Reuleaux, 1876). An important idea in this study is the configuration space of the machine, an abstract space with one dimension for each degree of freedom of the machine. For example, a machine with a single wheel rotating on an axle has one degree of freedom, and a single number (the angle the wheel makes with the axle) suffices to describe its state completely. A complex machine can have a configuration space with many dimensions, which can be expensive to analyze. However, the fact that the focus is on machines makes things better than they could be. In the abstract, a simple wheel has six degrees of freedom (three coordinates in space, plus three rotations to specify its orientation), and that is neglecting deformations. When it is stipulated that the wheel is a rigid body that makes permanent contact with a fixed axle, all but one of the degrees of freedom go away.

The configuration space for a machine can be divided into two regions: states of the machine in which parts would overlap and states where they would not. The second region consists of the states the machine can actually be in. The boundary between the two regions are the states in which parts are in contact, and it is here that most of the interesting behavior of the machine occurs, because forces are transmitted only during contacts.

A frequently studied example is the clock escapement. An example appears in Figure 9 (Faltings, 1990). The odd-shaped part at the top is called the lever. Its two ends catch the teeth of the escape wheel at the bottom, allowing it to turn only when the lever is disengaged, and hence synchronizing its turning rate to the period of the lever (which is connected to an oscillating balance wheel, not shown). This mechanism has two degrees of freedom, characterized by the angles of rotation of the lever and escape wheel around their axes. The configuration space of the mechanism is shown in Figure 10 (Faltings, 1990). It has two dimensions, reflecting the two degrees of freedom. Most of the space is gray, because it is impossible for the two angles to take on most of their possible values

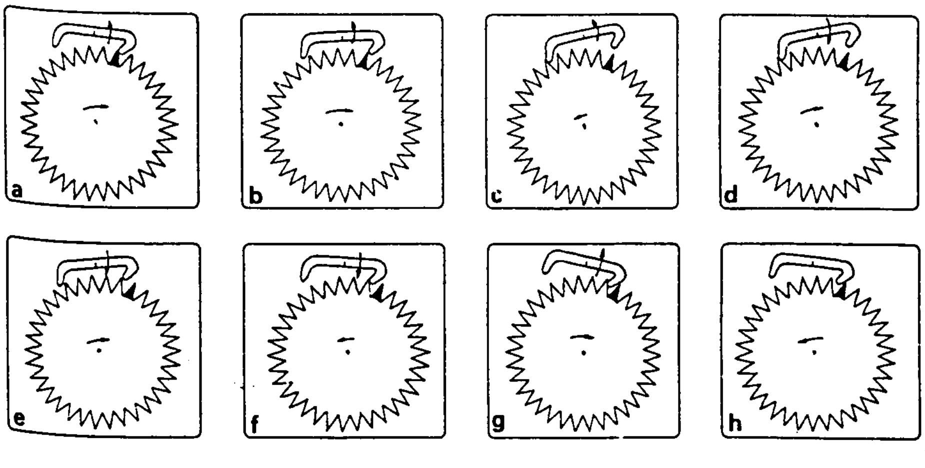


Figure 9. The behavior of a simple clock escapement (Faltings, 1990; courtesy of *Artificial Intelligence*).

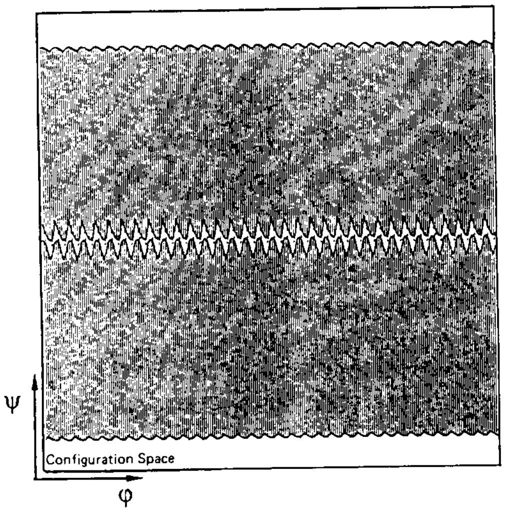


Figure 10. Configuration space for the mechanism of Figure 9 (Faltings, 1990; courtesy of *Artificial Intelligence*).

simultaneously without causing the lever and the escape wheel to overlap. The jagged white stripe in the middle is the escapement's normal region of operation. The open white space at the top and bottom is a normally unreachable region in which the lever is turned over, and never engages the escape wheel.

A variety of algorithms have been developed for making inferences about mechanisms like this, starting from a geometrical description. Joskowicz's (1987, 1988) algorithm finds a complete description of the boundary of the configuration space starting from a description of the boundaries of the parts. Faltings's (1990) algorithm divides the free region of configuration space into "places" in which the behavior of the machine has a simple qualitative description. This place vocabulary can then be used to support qualitative envisioning of the mechanism (Nielsen, 1988). The algorithms of Hoffman and Hopcroft (1987), Cremer (1989), and Gelsey (1990) avoid computing the configuration space at all, but go directly to quantitative simulation. Figure 11 shows a trace of the evolution of an escapement found by Gelsey's algorithm. This output is then analyzed to find repetitions of configurations, yielding a concise description of the basic loop the mechanism evolves through. Simulated experiments are then performed to determine how and when the loop comes to an end and a new regime begins, characterized by a change in the contacts responsible for the system's behavior. In the case of the escapement, the description says merely that the mechanism will oscillate at a constant period until the spring runs down.

BIBLIOGRAPHY

G. J. Agin and T. O. Binford, "Computer Descriptions of Curved Objects," *IEEE Trans. Comput.* **25**(4), 439–449 (1976).

R. T. Antony, "A Hybrid Spatial/Object-Oriented DBMS to Support Automated Spatial, Hierarchical, and Temporal Reasoning," in Chen, 1990, pp. 63–132.

D. Ballard and C. Brown, *Computer Vision*, Prentice-Hall, Englewood Cliffs, N.J., 1982.

B. G. Baumgart, *Geometric Modeling for Computer Vision*, Stanford Artificial Intelligence Lab Report STAN-CS-74-463, Stanford, Calif., 1975.

J. Bentley and J. H. Friedman, "Data Structures for Range Searching," *Comput. Surv.* **11**(4), 397–409 (1979).

R. A. Brooks, "Symbolic Reasoning Among 3-D Models and 2-D Images," *Artif. Intell.* **7**(1–3), 285–348 (1981).

A. Bundy, "Will It Reach the Top? Prediction in the Mechanics World," *Artif. Intell.* **10**(2), 129–146 (1978).

R. Chatila and J. Laumond, "Position Referencing and Consistent World Modeling for Mobile Robots," in *Proceedings of the*

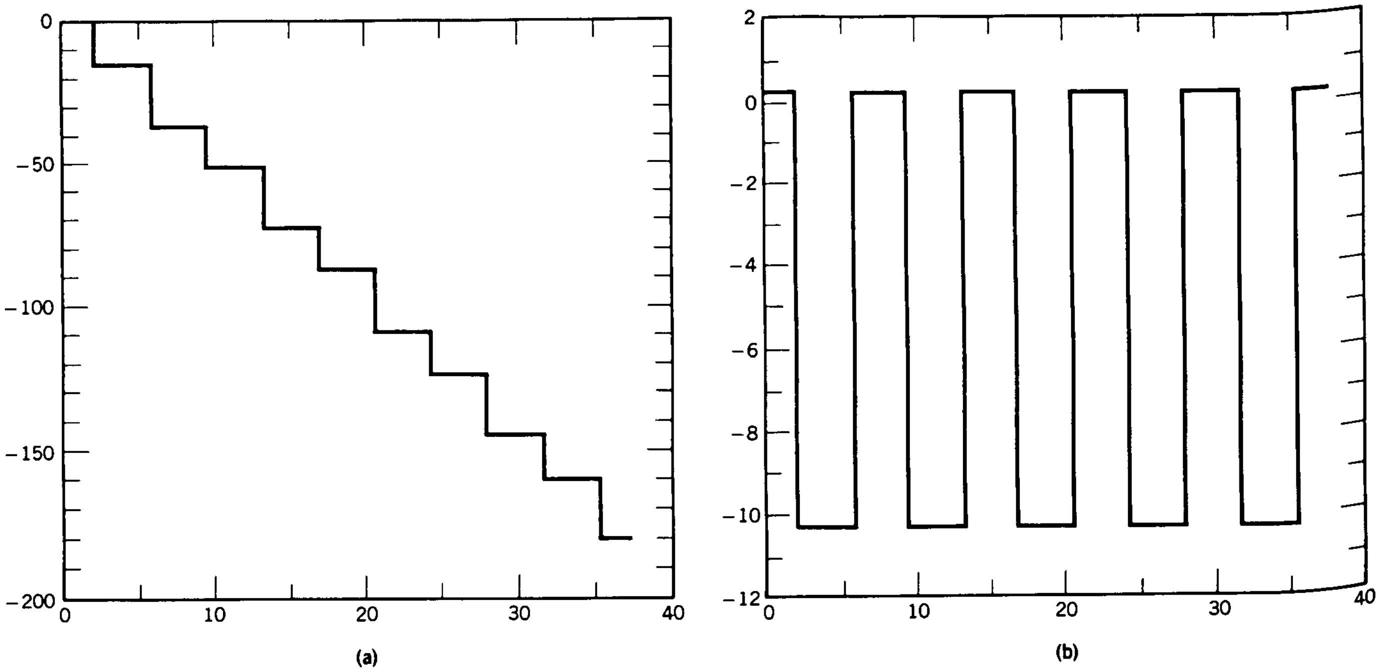


Figure 11. Simulated behavior of the escapement over time: (*a*) lever; (*b*) escape wheel.

IEEE International Conference on Robotics and Automation, IEEE Computer Society, Washington, D.C., 1985, pp. 138–170.

S. Chen, *Advances in Spatial Reasoning,* Vol. 1, Ablex Publishing Corp., Norwood, N.J., 1990.

J. F. Cremer, *An Architecture for General Purpose Physical System Simulation—Integrating Geometry, Dynamics, and Control,* Ph.D. dissertation, Cornell University, Ithaca, N.Y., 1989.

E. Davis, *Representing and Acquiring Geographic Knowledge,* Pitman Publishing, New York, 1985.

E. Davis, "Error Correction in Large-Scale Cognitive Maps," in *Proceedings of the SPIE Workshop on Sensor Fusion: Spatial Reasoning and Scene Interpretation,* Boston, Mass., 1988, pp. 332–337.

E. Davis, *Representations of Commonsense Knowledge,* Morgan-Kaufmann, San Mateo, Calif., 1990.

J. de Kleer, "Qualitative and Quantitative Reasoning in Classical Mechanics," in P. H. Winston and R. H. Brown, eds., *Artificial Intelligence: An MIT Perspective,* Vol. 1, MIT Press, Cambridge, Mass., 1979, pp. 11–30.

J. de Kleer and J. S. Brown, "A Qualitative Physics Based on Confluences," *Artif. Intell.* **24,** 7–83 (1984).

A. Elfes, "Using Occupancy Grids for Mobile Robot Perception and Navigation," *IEEE Comput.* (special issue), 46–58 (June 1989).

R. J. Elliot and M. E. Lesk, "Route Finding in Street Maps by Computers and People," in *Proceedings of the Second National Conference on Artificial Intelligence,* Pittsburgh, Pa., AAAI, Menlo Park, Calif., 1982, pp. 258–261.

B. Faltings, "Qualitative Kinematics in Mechanisms," *Artif. Intell.* **44**(1–2), 89–119 (1990).

K. Forbus, "Qualitative Process Theory," *Artif. Intell.* **24,** 85–168 (1984).

B. V. Funt, *WHISPER: A Computer Implementation Using Analogues in Reasoning,* Technical Report 76-09, University of British Columbia, Vancouver, 1976.

B. V. Funt, "Analogical Modes of Reasoning and Process Modelling," in N. Cercone and G. McCalla, eds., *The Knowledge Frontier,* Springer-Verlag, New York, 1987, pp. 414–428.

F. Gardin and B. Meltzer, "Analogical Representations of Naive Physics," *Artif. Intell.* **38**(2), 139–159 (1989).

A. Gelsey, *Automated Reasoning About Machines,* Report **785,** Yale University, New Haven, Conn., 1990.

A. Gelsey and D. McDermott, "Spatial Reasoning About Mechanisms," in Chen, 1990, pp. 1–33.

P. E. Hart, N. J. Nilsson, and B. Raphael," A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Trans. Sys. Sci. Cybernet.* 4(2), 100–107 (1968).

P. Hayes, "The Second Naive Physics Manifesto," in J. Hobbs and R. C. Moore, eds., *Formal Theories of the Commonsense World,* Ablex Publishing Corp., Norwood, N.J., 1985a, pp. 1–36.

P. Hayes, "Naive Physics I: Ontology for Liquids," in J. Hobbs and R. C. Moore, eds., *Formal Theories of the Commonsense World,* Albex Publishing Corp., Norwood, N.J., 1985b, pp. 71–107.

C. M. Hoffman and J. E. Hopcroft, "Simulation of Physical Systems from Geometric Models," *IEEE J. Robot. Automat.* 3(3), 194–206 (1987).

L. Joskowicz, "Shape and Function in Mechanical Devices," in *Proceedings of the Sixth National Conference on Artificial Intelligence,* Seattle, Wash., AAAI, Menlo Park, Calif., 1987, pp. 611–615.

L. Joskowicz, *Reasoning About Shape and Kinematic Function in Mechanical Devices,* Courant Institute of Mathematical Sciences Report 402, New York, 1988.

A. Kak, ed., Special Issue, *AI Mag.* **9** (Summer 1988).

A. Klinger and M. L. Rhodes, "Organization and Access of Image Data by Areas," *IEEE Trans. Patt. Anal. Machine Intell.* **1,** 50–60 (1979).

S. Kosslyn, *Image and Mind,* Harvard University Press, Cambridge, Mass., 1980.

S. Kosslyn, *Ghosts in the Mind's Machine: Creating and Using Images,* W. W. Norton, New York, 1983.

B. Kuipers, "Modeling Spatial Knowledge," *Cogn. Sci.* **2**(2), 129–154 (1978).

B. Kuipers and Y. Byun, "A Robust, Qualitative Method for Robot Spatial Reasoning," in *Proceedings of the Seventh National Conference on Artificial Intelligence,* St. Paul, Minn., AAAI, Menlo Park, Calif., 1988, pp. 774–779.

B. Kuipers and T. S. Levitt, "Navigation and Mapping in Large-Scale Space," *AI Mag.,* 25–43 (Summer 1988).

M. Lavin, "Analysis of Scenes from a Moving Viewpoint," in de Kleer, 1979, pp. 185–208.

T. S. Levitt, D. T. Lawton, D. M. Chelberg, and P. C. Nelson, "Qualitative Landmark-Based Path Planning and Following," in *Proceedings of the Sixth National Conference on AI,* AAAI, Menlo Park, Calif., 1987, pp. 689–694.

V. Lumelsky and A. Stepanov, "Path Planning Strategies for a Point Mobile Automaton Moving Amidst Unknown Obstacles of Arbitrary Shape," *Algorithmica* 3(4), 403–430 (1987).

D. Marr and H. K. Nishihara, "Representation and Recognition of the Spatial Organization of Three-Dimensional Shapes," *Proc. R. Soc. B* **200,** 269–294 (1978).

D. McDermott and E. Davis, "Planning Routes Through Uncertain Territory," *Artif. Intell.* **22,** 107–156 (1984).

J. S. B. Mitchell, "An Algorithmic Approach to Some Problems in Terrain Navigation," *Artif. Intell.* **37**(1–3), 171–201 (1988).

H. P. Moravec, "Sensor Fusion in Certainty Grids for Mobile Robots," *AI Mag.,* 61–74 (Summer 1988).

P. Moutarlier and R. Chatila, "Stochastic Multisensory Data Fusion for Mobile Robot Location and Environment Modelling," *Proc. Int. Symp. Robot. Res.* **5,** 207–216 (1989).

P. Nielsen, *A Qualitative Approach to Rigid Body Mechanics,* Ph.D. dissertation, University of Illinois, 1988.

G. Novak, "Representations of Knowledge in a Program for Solving Physics Problems," in *Proceedings of the Fifth IJCAI,* Morgan-Kaufmann, San Mateo, Calif., 1977, pp. 286–291.

Z. Pylyshyn, *Computation and Cognition: Toward a Foundation for Cognitive Science,* MIT Press, Cambridge, Mass., 1985.

A. A. G. Requicha, "Representations for Rigid Solids: Theory, Methods, and Systems," *ACM Comput. Surv.* **12,** 437–464 (1980).

F. Reuleaux, *The Kinematics of Machinery,* Macmillan and Co., London, 1876.

H. Samet, "The Quadtree and Related Hierarchical Data Structures," *Comput. Surv.* **16**(2), 187–260 (1984).

R. C. Smith and P. Cheeseman, "On the Representation and Estimation of Spatial Uncertainty," *Int. J. Robot. Res.* **5**(4), 56–68 (1986).

R. N. Shepard and J. Metzler, "Mental Rotation of Three-Dimensional Objects," *Science* **171,** 701–703 (1971).

A. Stevens and P. Coupe, "Distortions in Judged Spatial Relations," *Cogn. Psychol.* **10,** 422–437 (1978).

C. Thorpe, M. H. Hebert, T. Kanade, and S. Shafer, "Vision and

Navigation for the Carnegie Mellon Navlab," *IEEE Trans. Patt. Anal. Machine Intell.* 10(3), 362–373 (1988).

General References

N. Cercone and G. McCalla, eds., *The Knowledge Frontier,* Springer-Verlag, New York, 1987.

J. Hobbs and R. C. Moore, eds., *Formal Theories of the Commonsense World,* Ablex Publishing Corp., Norwood, N.J., 1985.

P. H. Winston and R. H. Brown, eds., *Artificial Intelligence: An MIT Perspective,* Vol. 1, MIT Press, Cambridge, Mass., 1979.

D. McDermott
Yale University

REASONING, TEMPORAL

It is hard to think of a research area within AI that does not involve reasoning about time in one way or another: medical-diagnosis systems try to determine the time at which the virus infected the blood system, circuit-debugging programs must reason about the period over which the charge in the capacitor increased, automatic programmers and program synthesizers must deduce that after procedure P is executed the value of variable X is zero, and robot programmers must make sure that the robot meets various deadlines when carrying out a set of tasks. One particular subfield of AI, which has become known as the area of temporal reasoning (TR), acknowledges this central role. Although most subfields merely employ temporal terminology, the very goal of TR is a general theory of time.

Of course, the passage of time is important only because changes are possible. In a world where no changes were possible (no viruses infecting blood systems, no electrical charges changing, no changes in program counters, not even changes in the position of the sun in the sky or the position of the hands on wristwatches) not only would there be no computational justification for keeping track of time but the very concept of time would become meaningless. Therefore, the ideal theory of time must meet two requirements. The first is that it provide a language for describing what is true and what is false over time. The second is that it provide a criterion of lawful change.

The work that is described here is mostly in the form of various formalisms, which usually means a logic with more or less well worked out syntax and semantics. The first section reviews the work done within AI, describing a somewhat idealized progression of such formalisms. The last section describes related work from philosophy and theoretical computer science.

How is temporal information represented? Suppose the task is to represent in logic the fact that the color of a particular house is red at time t. There are several options:

Time can simply be included as an argument to the predicate: COLOR(HOUSE,RED,T), where T is a time argument (a point, an interval, or otherwise).

The proposition can be "reified' (or, as McCarthy once proposed, "thingified"): HOLD(T,COLOR(HOUSE, RED); here COLOR serves as a function symbol rather than a relation symbol.

Time need not be mentioned at all! Instead, the interpretation of formulas is complicated. If in classic logic a formula φ is either true (written $\models \varphi$) or false (and in this article the discussion is confined to the propositional case), now a formula is either true at a given time t (written $t \models \varphi$) or false at that time. (Here again there is a choice of choosing t to be a point, an interval, or some other temporal entity.)

The first option is not acceptable from the standpoint of TR. If time is represented as an argument (or several arguments) so predicates, there is nothing general that can be said about it. For example, it cannot be said that "effects cannot precede their causes"; at most it can be said about specific causes and effects. Indeed, this first option accords no special status to time, neither conceptual nor notational, which goes against the grain of the TR spirit.

The second option fares better in this respect, and in one guise or another has been widely accepted in TR. Taking this approach seriously requires paying special attention to the meaning of the terms in the language, which now also serve the function of what otherwise would be relation symbols.

The third option, favored in modern philosophy and theoretical computer science, has for the most part been ignored in AI until very recently.

TEMPORAL REASONING IN AI

Situation Calculus, STRIPS, Histories, Intervals, and Chronicles

McCarthy and Hayes (1981) introduced the situation calculus (SC), a temporal formalism that to this day is the basis for many temporal representations. A situation is a snapshot of the universe at given moment. Actions are the means of transforming one situation into another. For example, by performing the action PICKUP(A) in the situation where ON(A, B) and ISCLEAR(A) are true, a new situation is arrived at where ISCLEAR(B) is true. The actual formal construction uses the function RESULT that accepts a situation and an action as arguments and returns a new situation. If in the above example the first situation is S1 and the second S2, then S2 = RESULT(S1,PICKUP(A)). It is possible, of course, to construct longer chains of action, as in S3 = RESULT(RESULT(S1,PICKUP(B)),PUTDOWN(B)).

SC makes several strong commitments. The first is about discreteness of time, which precludes discussion of continuous processes such as water flowing into a container and gradually filling it. The second is about contiguity of cause and effect, so to speak; the effects of an action are manifested at the very next situation. A further limitation of SC was that it did not allow concurrent actions, even in the framework of discrete time. The best known problem introduced by SC is the frame problem. Consider the same situations S1 and S2 as described above with the additional information that in S1 COLOR

(A,GREEN) is true. Is COLOR(A,GREEN) still true in S2? It is hoped that the answer is affirmative, but in fact this conclusion is not warranted by the theory. The RESULT function specifies only what changes as a consequence of taking an action but not what is true by virtue of having not changed. In order to be able to make those inferences, it is possible to add numerous frame axioms, specifying for each action what it does not affect. The first problem with this solution is that such axioms are numerous: Picking up a block does not change its color, does not affect any other block, does not change the president of the United States, etc. It is obviously impossible to explicitly list what is unaffected by an action. A further complication arises if concurrent actions are introduced. In this case frame axioms are simply wrong: Someone might paint the block as it is being PICKUPed.

As was mentioned before, despite these limitations SC has proved very influential. For example, it has been the basis for several planning (qv) systems. One of the first of such systems was STRIPS (Fikes and Nilsson, 1971), which embodied a natural solution to the frame problem. STRIPS is a name for both a formalism and a planner based on that formalism. This article is concerned primarily with the former. The STRIPS framework adopted the same view of time as SC but made the following addition. With each action STRIPS associated two lists. The addlist specified what becomes true as a result of the action, and the deletelist specified what ceases to be true after the action. The STRIPS assumption was that if action A transformed state S1 into state S2, then a proposition P was true in state S2 if and only if either P was in the addlist of A or P was true in S1 and was not in the deletelist of A. This assumption was the basis for the regression operator in the STRIPS planner (see PLANNING).

One of the strong advocates of formal reasoning about the commonsense world has been Hayes. Hayes (1984a, 1984b) offered a general justification of his approach as well as an actual formalism to describe the behavior of liquids (see PHYSICS, NAIVE). (These are slightly revised versions; the original papers were written in the 1970s.) In the latter paper Hayes introduced the notion of histories, which has had a strong influence on TR. A history is a connected piece of four-dimensional space–time. For example, a falling history is the space occupied by a liquid for the duration of its freefall. This view of the world is a radical departure from the SC paradigm. It acknowledges the continuous nature of time (and space, although that is not directly relevant to this discussion) and allows the representation of gradual change. There is no restriction to snapshots of the universe and a method for stringing them together. Instead an entire interval of time is described. Of course, it is possible to describe a snapshot of the universe by taking a slice through a history (which is the projection of the history onto the three-dimensional space at a given point in time). The theory of histories was partially applied by Forbus (1984) to reasoning about qualitative physics.

The spatial nature of Hayes's histories has drawn some criticism. For one thing, some occurrences do not have a well-defined spatial extent: What are the spatial boundaries of a conversation? Of an election? Of a confusion? Furthermore, histories are extensional in that the space–time of the history completely determines the history. But this cannot possibly be right, due to the fusion problem: more than one history can take place at the same space–time, eg, a concert history and a stale-air history. Nevertheless, Hayes's bold transition from states to interval inspired much of the later work, including Allen's interval calculus and McDermott's temporal logic.

Retaining the interval-based view of time, Allen (1984) proposed a theory of action and time and identified the 13 possible relations between two intervals (identity, the one totally preceding the other, overlapping, etc). The ontology proposed by Allen associates with an interval one of three objects: a property, an event, or a process. A property is something that is statically true or false, eg, "the pen is red." Allen represented this assertion by the formula HOLD(I, COLOR(PEN, RED)). Properties hold uniformly throughout an interval; a proposition holds for an interval exactly when it holds for all subintervals: $\forall\, I, P$ HOLD(I, P) $\equiv \forall\, I' \in I$ HOLD(I', P) (and in fact Allen required a slightly stronger axiom). Because properties are reified propositions, to retain their intuitive meaning, Allen simulated the logical connectives. For example, HOLDS(I, AND(P, Q)) $\equiv$ HOLDS(I, P) $\wedge$ HOLDS(I, Q) is an axiom in Allen's system.

Contrary to properties, events are holistic entities. If an event occurred over an interval, it did not occur over any subinterval. An example of an event is "I went to the shop"; such an event is repeatable but not divisible. The predicate denoting occurrence of events is OCCUR(I, EVENT). Processes are a hybrid case. An example of a statement describing a process is "I am walking"; if it is true for an interval, it must be true for some subinterval but need not be true for all of subintervals (I may rest during my hour-long walk).

Allen's proposal also included an account of causation. Event causation describes a relation between events (and associated intervals). The properties of this relation are exactly the following: the occurrence of causes entails the occurrence of effects, effects may not precede their causes, and the relation is transitive and antisymmetric. Other parts of Allen's proposal include the notions of agents, actions, level generation (Goldman, 1970), intentions, plans, commitment, knowledge, and belief.

McDermott (1978) began exploring the connection between problem solving (qv) and theories of time and action. He proposed a temporal logic to be used in the process of planning (McDermott, 1982a), in which he introduced the notion of chronicles. McDermott's construction takes as primitive the notion of a state, which is a point in time in some set of possible worlds. A fact type is a reified proposition that may be instantiated as a fact token. For example, "I am walking" is a fact type, and "I am walking on 1.1.2000 from 1:00 to 1:45" is a fact token. The construction is set theoretic: the assertion $T(S, P)$, denoting the existence of a particular fact token, is shorthand for $S \in P$. Similarly for an event type E, the assertion OCC(S1, S2, E), denoting the existence of an event token, is merely shorthand for $\langle$S1, S2$\rangle \in E$. The assertion TT(S1, S2, P) ("p is true throughout the interval [s1, s2]") is shorthand for [S1, S2] $\subset P$.

McDermott assumes that time, as well as being a partial order, is dense. He furthermore arranges time in chronicles, which are linear timelines that form a treelike structure: chronicles may branch into the future, and once they do, the different branches do not meet again. When taking an action is contemplated, it is necessary to compare the world (or chronicle) in which the action is taken to the world in which it is not. McDermott's formulation was the first one to provide a mechanism for such a comparison within the logic; different worlds are simply separate paths in the chronicle tree. In particular, he managed to give meaning to the notion of preventing.

McDermott also devotes a large part of the discussion to causation as a representation of rules governing change. The notion he considers are ECAUSE, for one event causing another, and PCAUSE, for an event causing a fact. The former allow for a delay between the cause and the effect. The latter are particularly interesting, because they introduce the notion of persistences. The idea is that the effect of a PCAUSE is a defeasible prediction. For example, the predicted effect of a boulder rolling to a particular location is that the boulder will be there for the next 50 years. This prediction will be violated if a construction company decides to erect a building on that particular site at an earlier time. Persistences, whose semantics are loosely based on nonmonotonic logic (see REASONING, NONMONOTONIC) are McDermott's (1982b) solution to the frame problem. Other topics covered by McDermott are continuous change, qualitative physics and potrans (potential transfer), planning, and the notion of a subtask. Further applications of temporal reasoning to planning have been published (McDermott, 1984).

Application to Qualitative Physics

Much of physical reasoning involves time for obvious reasons. It is possible to reason about what a physical system will do starting from initial conditions, or contrariwise it is possible to reason backward from final conditions to an explanation in terms of previous behavior. Most of the work has focused on the former process, which is called envisioning (de Kleer, 1975) or projection (Wilenskey, 1983; Simmons, 1983).

Envisioning is not the same as simulation. It is desired that the reasoner see interesting patterns in the behavior, that it recognize an oscillator, not just become one. Furthermore, it is important that the reasoning be able to proceed in the absence of the detailed quantitative information that a simulator would require. The resulting paradigm is called qualitative reasoning about physical systems. It has been pursued by de Kleer and Brown (1984), Forbus (1984), and Kuipers (1984) (see PHYSICS, NAIVE; REASONING, CAUSAL; REASONING, COMMONSENSE).

There are two phases to the reasoning in this paradigm. First the program must produce an abstract version of the structure of the system, then it must use this structure to reason. In almost every case, the abstract structure is analyzed as a set of devices with connections to each other. Causality flows through the connections from one device to another. The state of a device is represented by the values of a set of quantities. A device can have different behaviors depending on what ranges various quantities lie in. For instance, a thermostat-controlled heating system will be characterized by quantities such as room temperature and heat flow out of room. The furnace will be in state OPERATIVE or OFF depending on whether the room temperature is greater than the thermostat setting. If it is OPERATIVE and the tank is NONEMPTY, there will be a constant value of furnace heat flow into the room. And so forth.

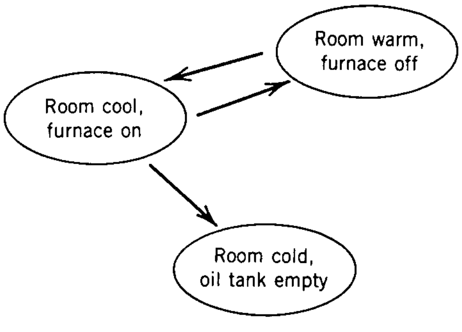


Figure 1. A simple state diagram.

Devices connect quantities together; they enforce certain relationships among them. Hence, from an initial set of quantity relationships, the temporal reasoner can deduce how the quantities will change. If the furnace is on, and it is not too cold outside, the room temperature will increase. Any such change, if allowed to continue, will eventually drive some quantity out of its current range, thus changing the behavior of some device. The reasoner must, therefore, reproject the behavior of the system under the altered circumstances. The new analysis will result in a different pattern of changes, leading to new behavior shifts, and so forth. The final analysis may be displayed as a graph, as in Figure 1. States of the system are shown as ovals. An arrow joins two states if the first may evolve into the second through quantity changes. A loop in the graph indicates that the system may oscillate between the two given states. If the analysis is correct and useful, it can be expected that a finite number of significantly different states will be found, and hence it is expected that every system to enter one state and will stay there or loop among some set of states. Note that the "furnace on" state has two potential successors: either the room becomes warm or the oil runs out. Both possibilities appear because the qualitative analysis shows two quantities changing: the room temperature is increasing and the oil level is decreasing. Without further information about the magnitudes of the quantities, the reasoner cannot decide which will reach its threshold first and cause the system to change state.

The early work by Hendrix (1973) anticipated much of current research in planning and qualitative physics. Hendrix's extension of the STRIPS formalism includes representation of continuous change (using real values for quantities) and concurrent actions. The central component of the system (which is a set of data structures and a skeleton of a simulation program rather than a logical formalism) is the process monitor. This module continuously attempts to identify "active processes" and compute their effects.

Rieger (1976) also addressed the issue of continuous change in his proposal of commonsense algorithms (1976). Among the many notions he considered are continuous causality and gated causal rules. His system too is couched in terms of data structures and algorithms, and McDermott (1982a) pointed out some of the difficulties in assigning meaning to the symbols in Rieger's system.

Another system that is primarily a serious attempt at incorporating one of the more sophisticated logics into a computer program is Dean's (1986) Time Map Managing System. The system, which can be viewed as a temporal reason-maintenance system, was designed to be used as part of an automated planner. It is loosely constructed around McDermott's temporal logic. A earlier attempt along similar, although more modest, lines is Vilain's (1982) system, a time-maintenance system based on a formalism that is similar to Allen's.

It is worth pointing out that the more "applied" work on TR has not been based to any great extent on the representational research described earlier. This can be viewed as evidence that the foundations of the applied work are shaky, or that the logic-level research has not yet become sophisticated enough to be of real value, or both.

WORK OUTSIDE AI

Both AI and theoretical computer science owe an intellectual debt to philosophy, where time, action, and causation have been studied for many years. Although it is not possible here to properly cover the relevant philosophical literature, two good expositions of the work done on causation are available (Mackey, 1974; Sosa, 1975). Other investigations into the nature of actions have been published (Goldman, 1970; Davidson, 1967).

Work done in formal philosophy on the logic of time is particularly relevant. It was mentioned in the introduction that one possibility for representing temporal information is to have time implicit in the interpretation of the formulas. The way this option is exercised is through modal logic, to which a modern introduction is Chellas (1980). The formulas of the logic are augmented by one or more modal operators; if P is a wff and $\bigcirc$ is a (unary) modal operator, then $\bigcirc P$ is a wff too. In the basic modal logic there is a single modal operator $\Box$, called "box" and pronounced "necessarily." Its dual operator $\Diamond$, called "diamond" and pronounced "possibly," is defined by $\Diamond P \equiv \neg \Box \neg P$. The widely accepted semantics for the resulting logic are possible world semantics introduced by Kripke (1963). When applied to temporal logic, possible worlds are equated with time points, and the modal operators are usually some variant of the following.

FP: p is true in some future time point.

GP: p is true in all future time points.

PP: p is true in some past time point.

HP: p is true in all past time points.

The outstanding feature of these systems is the indexicality of time: formulas are interpreted with respect to a time point (usually called now) and may contain reference to other time points through use of the modal operators. Thus if p is taken to mean "it is raining," the formula $\neg p \supset \mathrm{G} \neg p$ means "if it isn't raining now then it never will." From this logic a more traditional modal logic can be derived such as S4, for example by defining $\Box p$ to be $p \wedge \mathrm{G}p$. (The other well-known construction is define $\Box p$ to be $\mathrm{H}p \wedge p \wedge \mathrm{G}p$.)

Prior (1967) is the philosopher widely credited for first applying the principles of modal logic to temporal logic. He explored various possible properties of time (linearity, future branching, circularity, additivity, having a metric defined on it) and related the resulting logics to known modal systems. Rescher and Urquhart (1971) did the same, employing more conventional notation and using more recent results from logic. In particular, they discussed decision procedures based on the semantic tableau method. The most recent comprehensive study of temporal logics appears in a book by van Benthem (1983). He conveniently divides the discussion into the structure of time on the one hand and the nature of temporal assertions on the other. In each part he considers two cases: basing the logic on points and basing the logic on intervals, the latter possibility having recently gained currency in philosophy. As was said earlier, the philosophical literature on time is very rich; these references are merely initial pointers to it.

In theoretical computer science there has been considerable interest in TR, although until recently there was no overlap between that work and TR in AI. Pnueli (1977, 1979) was the first in computer science to apply modal temporal logic to program verification. There are several variants of modal temporal logic, stemming from different models of time (discrete or continuous, linear or branching) and different choices of modal operators (Emerson and Halpern, 1983). Discreteness is usually assumed, and "time points" correspond to the instants when the program interpreter is about to execute the next command. The modal operator $\bigcirc$ is the "next-state" operator: $\bigcirc P$ is true if and only if P is true the next time the interpreter is about to execute a command. Using the logic, various properties can be expressed very concisely, such as termination, freedom from deadlock, fair execution, and more (Manna and Pneuli, 1981). Temporal logic has been the framework in which much research on concurrent computation was done (Gabbay and co-workers, 1980).

Dynamic logic, first introduced by Pratt (1976) in conjunction with Moore, is the other major area of TR in theoretical computer science, and it too is geared toward reasoning about computer programs and digital devices. Rather than the usual modal operator of temporal logic, dynamic logic associates modal operators with each program. If α is a (nondeterministic) program, then $\langle\alpha\rangle P$ means that P holds after some possible execution of α. Similarly, $[\alpha]P$ means that P holds after all possible executions of α. For a systematic treatment of dynamic logic, from so-called simple dynamic logic to the full-fledged first-order one, see Harel (1979).

Both dynamic logic and temporal logic interpret statements over time points. Because in AI statements are often encountered that refer to time intervals rather than time points ("the robot performed the task," "I solved the problem"), neither formalism is completely adequate for AI. There have been several extensions of these formalisms to time intervals. For example, dynamic logic was generalized to process logic (Pratt, 1979; Harel and coworkers, 1982). In process logic formulas are interpreted over paths, or sequences of discrete time points. In Harel and co-workers' (1982) version, the two modal operators (in addition to the ones introduced by dynamic logic) are F ("first") and SUF (roughly, "until"). $S_1, \ldots, S_n \models \mathrm{F}\, p$ iff $S_1 \models p$. $S_1, \ldots, S_n \models p$ SUF q iff, for some j, $S_i, \ldots, S_n \models p$ for all i, $0 < i < j$, and $S_j, \ldots, S_n \models q$.

Interval temporal logic is a similar formalism, introduced by Moszkowski (1983) in conjunction with Halpern and Manna, which was applied to reasoning about digital devices. There are several other logics of time intervals, including a proposal by Halpern and Shoham (1986). This logic, which extends point-based temporal logic in a way that is analogous to the way process logic generalizes dynamic logic, is one of few temporal logics in computer science (whether point-based or interval-based) that are not committed to the discrete view of time.

It was mentioned at the beginning that to date neither temporal logic nor dynamic logic have had much influence on AI. Some exceptions, however, can be found (Fusaoka and co-workers, 1983; Moszkowski, 1985; Shoham, 1986; Mays, 1983; Georgeff and Lansky, 1985).

BIBLIOGRAPHY

J. F. Allen, "Towards a General Theory of Action and Time," *Artif. Intell.* **23**(2), 123–154 (July 1984).

B. F. Chellas, *Modal Logic,* Cambridge University Press, Cambridge, UK, 1980.

D. Davidson, "The Logical Form of Action Sentences," in N. Rescher, ed., *The Logic of Decision and Action,* Pittsburgh University Press, Pittsburgh, Pa., 1967.

T. Dean, *Temporal Imagery: An Approach to Reasoning about Time for Planning and Problem Solving,* Ph.D. dissertation, Yale University, New Haven, Conn., 1986.

J. de Kleer, *Qualitative and Quantitative Reasoning in Classical Mechanics,* Technical Report 352, MIT Artificial Intelligence Lab, Cambridge, Mass., 1975.

J. de Kleer and L. Seely Brown, *A Qualitative Physics Based on Confluences,* Technical Report, Xerox PARC Intelligent Systems Laboratory, Palo Alto, Calif., Jan. 1984.

E. A. Emerson and J. Y. Halpern, "'Sometimes' and 'Not Never' Revisited: on Branching versus Linear Time," in *Proceedings of the Tenth ACM Symposium on Principles of Programming Languages,* 1983, pp. 127–140.

R. Fikes and N. J. Nilsson, "STRIPS: A New Approach to Application of Theorem Proving to Problem Solving," *Artif. Intell.* **2,** 189–208 (1971).

K. D. Forbus, *Qualitative Process Theory,* Ph.D. dissertation, Artificial Intelligence Laboratory, MIT, Cambridge, Mass., 1984.

A. Fusaoka, H. Seki, and K. Takahashi, "A Description and Reasoning of Plant Controllers in Temporal Logic," in *Proceedings of the Eighth IJCAI,* Karlsruhe, FRG, Morgan-Kaufmann, San Mateo, Calif., 1983, pp. 405–408.

D. Gabbay and co-workers, "On the Temporal Analysis of Fairness" in *Proceedings of the Seventh ACM Symposium on Principles of Programming Languages,* 1980, pp. 163–173.

M. P. Georgeff and A. L. Lansky, "A Procedural Logic," in *Proceedings of the Ninth IJCAI,* Los Angeles, Calif., Morgan-Kaufmann, San Mateo, Calif., 1985, pp. 516–523.

A. Goldman, A *Theory of Human Action,* Princeton University Press, Princeton, N.J., 1970.

J. Y. Halpern and Y. Shoham, *A Propositional Modal Logic of Time Intervals, Logic in Computer Science,* Springer-Verlag, New York, June 1986.

D. Harel, "First-Order Dynamic Logic," in Goos and Hartmanis, eds., *Lecture Notes in Computer Science,* Vol. 68, Springer-Verlag, New York, 1979.

D. Harel, D. Kozen, and R. Parikh, "Process Logic: Expressiveness, Decidability, Completeness," *JCSS* **25**(2), 145–180 (Oct. 1982).

P. J. Hayes, "The Naive Physics Manifesto," in J. R. Hobbs and R. C. Moore, eds., *Formal Theories of the Commonsense World,* Ablex, Norwood, N.J., 1984.

P. J. Hayes, "Naive Physics 11: Ontology for Liquids," in J. R. Hobbs and R. C. Moore, eds., *Formal Theories of the Commonsense World,* Ablex, Norwood, N.J., 1984b.

G. G. Hendrix, "Modeling Simultaneous Actions and Continuous Processes," *Artif. Intell.* **4,** 145–180 (1973).

S. Kripke, "Semantical Considerations on Modal Logic," *Acta Philos. Fenn.* **16,** 83–94 (1963).

B. Kuipers, "Commonsense Reasoning about Causality: Deriving Behavior from Structure," *Artif. Intell.* **24**(1), 169–203 (1984).

J. M. McCarthy and P. J. Hayes, "Some Philosophical Problems from the Standpoint of Artificial Intelligence," in *Readings in Artificial Intelligence,* Tioga, Palo Alto, Calif., 1981, pp. 431–450.

D. V. McDermott, "Planning and Acting," *Cogn. Sci.* **2**(2), 71–109 (1978).

D. V. McDermott, "A Temporal Logic for Reasoning about Processes and Plans," *Cogn. Sci.* **6,** 101–155 (1982a).

D. V. McDermott, "Nonmonotonic Logic II: Nonmonotonic Modal Theories," *JACM,* **29**(1), 33–57 (1982b).

D. V. McDermott, "Reasoning about Plans," in J. R. Hobbs and R. C. Moore, eds., *Formal Theories of the Commonsense World,* Ablex, Norwood, N.J., 1984.

J. L. Mackey, *The Cement of the Universe: a Study of Causation,* Oxford University Press, Oxford, UK, 1974.

Z. Manna and A. Pneuli, *Verification of Concurrent Programs: Temporal Proof Principles,* Technical Report, Weizmann Institute, Department of Applied Mathematics, Rehovot, Israel, Sept. 1981.

E. Mays, "A Modal Temporal Logic for Reasoning about Change," in *Proceedings of the Annual Conference of the Association for Computational Linguistics,* Cambridge, Mass., June 1983.

B. C. Moszkowski, *Reasoning about Digital Circuits,* Ph.D. dissertation, Stanford, University, Stanford, Calif., July 1983.

B. C. Moszkowski, *Executing Temporal Logic Programs,* Technical Report **71,** University of Cambridge, Cambridge, UK, Aug. 1985.

A. Pneuli, "A Temporal Logic of Programs," in *Proceedings of the Eighteenth FOCS,* IEEE, Oct. 1977, pp. 46–57.

A. Pneuli, "The Temporal Semantics of Programs," *Theor. Comput. Sci.* **13,** 45–60 (1979).

V. R. Pratt, "Semantical Considerations on Floyd-Hoare Logic," in *Proceedings of the Seventeenth FOCS,* IEEE, Oct. 1976, pp. 109–121.

V. R. Pratt, "Process Logic," *Proceedings of the Sixth POPL,* ACM, Jan. 1979, pp. 93–100.

A. N. Prior, *Past, Present, and Future,* Clarendon Press, Oxford, UK, 1967.

N. Rescher and A. Urquhart, *Temporal Logic,* Springer-Verlag, New York, 1971.

C. Rieger, "An Organization of Knowledge for Problem Solving and Language Comprehension," *Artif. Intell.* **7** (1976).

Y. Shoham, *Reasoning about Change: Time and Causation from the Standpoint of Artificial Intelligence,* Ph.D. dissertation, Yale University, New Haven, Conn., 1986.

R. Simmons, "The Use of Qualitative and Quantitative Simulations," in *Proceedings of the Third National Conference on Artificial Intelligence,* Washington, D.C., AAAI, Menlo Park, Calif., 1983.

E. Sosa, *Causation and Conditionals,* Oxford University Press, Oxford, UK, 1975.

J. F. A. K. van Benthem, *The Logic of Time,* Reidel, Dordrecht, 1983.

M. B. Vilain, "A System for Reasoning about Time," in *Proceedings of the Second National Conference on Artificial Intelligence,* Pittsburgh, Pa., AAAI, Menlo Park, Calif., 1982, pp. 197–201.

R. Wilenskey, *Planning and Understanding,* Addison-Wesley Publishing Co., Inc., Reading, Mass., 1983.

Y. Shoham
D. V. McDermott
Yale University

RECOVERY. See Visual recovery.

RECURSION

In this entry recursion is just a self-referential feature for procedures (and similar constructs) in certain programming languages. It was first available in LISP (qv) (McCarthy and co-workers, 1962), the principal systems implementation language for AI, and is available in most modern programming languages; for example, Pascal (Wirth, 1976), Logo (qv) (Harvey, 1985), Ada (Wiener and Sincovec, 1983), and Modula 2 (Ogilvie, 1985). Rogers (1987) and Manna (1974) contain fine mathematical treatments of recursion's connections to mathematical recursive (or inductive) definitions, the subtleties of assigning formal semantics to recursion, and the use of proofs by mathematical induction to verify correctness of recursive procedures. Shapiro (1986) is an excellent introduction to programming in LISP that effectively teaches one how to think recursively.

Essentially, recursion simply allows the instructions of a computer procedure to invoke the procedure itself. This brings up questions as to how it is possible to implement a thing like that and why anyone would want to use it. Suppose such a feature is allowed in computer procedures; the entry sketches how it is actually implemented. First, the usefulness of recursion is motivated by presentation of a simple, but illustrative example.

Some programming tasks create headaches for the programmer by ostensibly requiring that he or she devise methods to handle tedious "bookkeeping" subtasks that are subsidiary to the main tasks. Many times in such situations recursion can be used to free the programmer's mind of such unpleasant details; the details are handled instead by the computer implementation of recursion. Recursion in these cases enables the human programmer to produce succinct, conceptually clean programs that are easy to understand and verify as being correct.

Sometimes in solving problems the intelligent thing to do, artificially or otherwise, is to systematically consider all the possibilities in a given situation. This can take the form of exhaustively and systematically searching some, perhaps complicated, structure (see Search). Two examples are searching a maze for desirable objects and searching a game tree (qv) for good moves. It is difficult in programming such searches to devise correct, clear strategies for the program to keep track of where it has already searched and where it still needs to go.

Here is an example. Imagine wanting to employ a programmable, electronic monkey to collect all the bananas in any "tree" of a certain type. A picture of the type of tree is presented in the tree-shaped diagram of Figure 1. This figure consists of dots and lines. The dots are called nodes. The bottom node in such a tree is called the root. For example, node 1 is the root of the tree pictured in Figure 1. Branching upward from each node are either two lines leading to two respective nodes or no lines leading to nodes. The entire tree is finite. The tree pictured in Figure 1 has 15 nodes that are numbered to facilitate some of the exposition. Assume that any such tree has bananas only at its nodes. These assumptions as to the type of tree to be considered are merely to simplify the problem. The monkey is initially placed on the root of a tree and is capable of understanding and performing the following primitive tests and instructions that have obvious corresponding meanings. The tests are: "bananas_present_at_current_node" and "there_is_a_node_above." The instructions are: "pick_up_bananas_present_at_current_node," "climb_up_one_node_to_the_left," "climb_up_one_node_to_the_right," and "climb_down_one_node." Assume the monkey can be programmed with a block-structured lan-

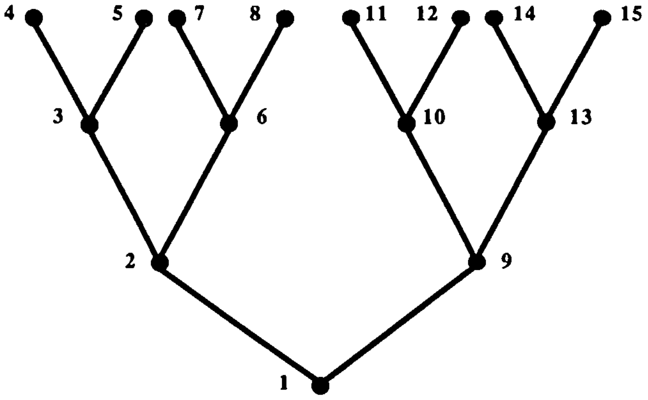


Figure 1. A tree is comprised of a root (*1*), nodes, and branches, and is finite.

ENCYCLOPEDIA OF ARTIFICIAL INTELLIGENCE

This extensively revised and expanded *Second Edition* of the *Encyclopedia of Artificial Intelligence* defines the discipline by bringing together the core of knowledge from all fields encompassed by AI. It covers the latest developments in current AI topics such as neural networks, fuzzy logic, machine vision, natural language generation, and many more. Includes:

- Over 450 articles—all entries written expressly for the *Encyclopedia*
- Over 5,000 literature references; 454 illustrations and color photographs
- Over 50% new and revised material
- Exemplary indexing and cross-referencing for easy, complete information access to all topics

Praise for the *First Edition*...

"The *Encyclopedia* is a wonder of clarity and scope: surprisingly easy to read ... the clarity is an especially pleasant surprise, considering the articles were all written by AI experts ... It's a treasure house of easily accessible knowledge."
—*Language Technology*

"Excellent bibliographies are attached to most of the articles, and diagrams and sketches are clear and helpful. The indexing and cross-indexing are exemplary. As the editor points out, the reader will be led by the extensive cross-references to almost every other article ..."
—*Artificial Intelligence Reporter*

"The *Encyclopedia* is a first-class piece of work that will be an indispensable part of any AI library."
—*Computing Reviews*

"... A tour de force ... a truly fantastic encyclopedia which no one in the field of artificial intelligence can afford to be without."
—*Systems Research & Information*

WILEY-INTERSCIENCE
John Wiley & Sons, Inc.
Professional, Reference and Trade Group
605 Third Avenue, New York, N.Y. 10158-0012
New York • Chichester • Brisbane • Toronto • Singapore
ISBN 0 471-50307-X (Two-Volume Set)
ISBN 0 471-50305-3 (Vol. 1)
ISBN 0 471-50306-1 (Vol. 2)

ISBN 0-471-50306-1
90000
9 780471 503064