Excerpt from "Exploring Artificial Intelligence: Survey Talks from the National Conferences on Artificial Intelligence," Ed. H. Shrobe, Morgan-Kaufmann, 1988.

Chapter

10

Evidential Reasoning Under Uncertainty

Judea Pearl
Cognitive Systems Laboratory
Computer Science Department
University of California, Los Angeles

1 *Introduction*

1.1 *Overview*

One can hardly identify a field in AI that doesn't use some sort of evidential reasoning, namely, processes leading from evidence or clues to guesses and conclusions under conditions of partial information. Therefore, to avoid having to cover the entire field of AI, the topic will be limited to evidential reasoning tasks in which the uncertainty is given a specific notation, namely, it is represented explicitly by some sort of measure or degree.

Constrained by this guideline, I will not be able to give a full account of the heuristic approaches to evidential reasoning [Cohen, 1985; Clancey, 1985] nor to works in truth-maintenance systems and nonmonotonic reasoning that, essentially, address the same sort of problems. The latter are given full coverage by other surveys (see this volume), and will only be touched on briefly to point out their fundamental ties to other formalisms.

Additionally, it will not be possible to survey everything that anyone has said or written about uncertainty, nor would I be able to summarize the intricacies of powerful programs such as MYCIN [Shortliffe, 1976], INTERNIST [Miller et al., 1982], PROSPECTOR [Duda et al., 1976], MEDAS [Ben-Bassat et al., 1980], RUM [Bonissone et al., 1987], MUM [Cohen et al., 1987a] and MDX [Chandrasakaran and Mittal, 1983] that have embodied practical solutions to various aspects of reasoning with uncertainty. This survey focuses on a select set of issues, trends, and principles that have emerged from these past works and which I hope to describe in a unifying perspective and in greater depth than a more general survey would permit. For more extensive surveys, the reader is referred to [Thompson, 1985; Prade, 1983; Stephanou and Sage, 1987], and the works collected in [Kanal and Lemmer, 1986]. Expanded technical treatments of the topics discussed in this survey can be found in [Pearl, 1988a].

The thrust of this survey is shown in Figure 1—it depicts my perception of current approaches to evidential reasoning and is, in fact, a summary of this discussion. I will spend the first part discussing the general needs and difficulties of managing uncertainty, and then talk about two diametrically opposed approaches to the problem; one called *extensional*, the other *intensional* .[1] The extensional approach, also known as production systems, rule-based systems, or procedure-based systems, treats uncertainty as a generalized truth value attached to formulas and, following the tradition of classical logic, computes the uncertainty of any formula as a function of the uncertainties of its subformulas. It is characterized by computationally attractive features, but is semantically sloppy. In the intensional approach, also known as declarative or model-based approach, uncertainty is attached to "states of affairs" or subsets of "possible worlds." It is semantically clear but computationally clumsy. Naturally, there have been attempts from both sides to rectify their respective deficiencies. I will briefly discuss (Section 2) some movements from the extensional to the intensional, and will spend most of the time on movements with which you are more familiar, namely, attempts to make intensional approaches computationally more attractive (Section 3).

In this vein, I will discuss the central role of *belief networks* representations, both the Bayesian type and the Dempster-Shafer type. Finally, I will speculate (Section 4) on the middle ground toward which the two approaches will hopefully converge in the next few years. This area, I believe, will involve the issues of encoding context-dependent information, the formalization of relevance, and network decomposition techniques.

1 This terminology is due to [Perez and Jirousek, 1985].

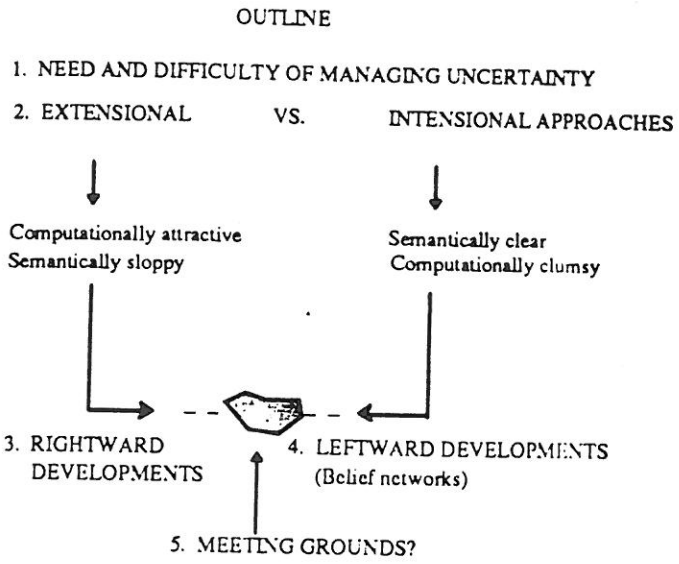


Figure 1 Outline of survey and relationships between extensional and intensional approaches to uncertainty.

1.2 *Why Bother with Uncertainty?*

Reasoning about any realistic domain always requires that some simplifications be made. By necessity, we leave many facts unknown, unsaid, or crudely summarized. For example, most rules used to encode knowledge and behavior have exceptions that one cannot afford to enumerate, and the situations in which the rules apply are usually ambiguously defined or hard to satisfy precisely in real life. Reasoning with exceptions is like navigating through a minefield; most steps are safe but some can be devastating. Given its location, each mine can be avoided or diffused, but we must start our journey with a map the size of a postcard, with no room to mark down the exact location of every mine or the way they are wired together. An alternative to the extremes of ignoring or enumerating exceptions, is to *summarize* them, i.e., provide some warning signs to indicate which areas of the minefield are more dangerous than others. Such summarization is essential if we wish to find a reasonable compromise between safety and speed of movement.

1.3 *Why Is It Hard?*

One way of summarizing exceptions is to assign to propositions numerical measures that combine according to uniform syntactic principles, similar to the way truth values are combined in logic. This approach has been adopted by first-generation expert systems, but often yields unpredictable and counterintuitive results, examples of which will soon be demonstrated. As a matter of fact, it is remarkable that this combination strategy went as far as it did, in view of the fact that uncertainty measures stand for something totally different than truth values. While truth values in logic characterize the formulas under discussion, uncertainty measures characterize exceptions, i.e., the invisible facts *not* shown in the formulas. Accordingly, while the syntax of the formula is a perfect guide for combining the visibles, it is close to useless when it comes to combining the invisibles. For example, the machinery of Boolean algebra gives us no clue as to how the exceptions to $A \rightarrow C$ interact with those of $B \rightarrow C$ to yield the exceptions to $(A \wedge B) \rightarrow C$. These invisible exceptions may interact in very intricate and clandestine ways, as a result of which we lose most of the computationally attractive features of classical logic, e.g., modularity and monotonicity.

Although in logic, too, formulas interact in intricate ways, the interactions are visible. This enables us to calculate the impact of each new fact *in stages*, by a process of derivation that resembles the propagation of a wave: We first compute the impact of the new fact on a set of syntactically related sentences, S_1, store the results, then propagate the impact from S_1 to another set of sentences, S_2, and so on, without having to come back and redo S_1. Unfortunately, this computational scheme, so common to logical deduction, cannot be justified under uncertainty unless one makes restrictive assumptions, that, in probabilistic terms, amount to *conditional independence*.

Another feature we lose in going from logic to shaded uncertainties is *incrementality*. What we would like to do when we have several items of evidence is to account for the impact of each of them individually: Compute the effect of the first item, then attend to the next, absorb its added impact, and so on. This, too, can only be done after making restrictive assumptions of independence. Thus, it appears that uncertainty reasoning represents a hopeless case of having to compute the impact of the entire set of past observations on the entire set of sentences in one global step. This, of course, is an impossible task.

1.4 *Three Approaches to Uncertainty*

AI researchers tackling these problems can be classified into three schools, which I will call: logicist, neo-calculist, and neo-probabilist. The logicist school attempts to deal with uncertainty using nonnumerical techniques. The neo-calculist school uses numerical representations of uncertainty but, believing that

probability calculus is inadequate for the task, invents entirely new calculi, such as the Dempster-Shafer calculus, fuzzy logic, certainty factors, and so on. Finally, the neo-probabilists remain within the traditional framework of probability theory, while attempting to equip the theory with computational facilities needed to perform AI tasks. This taxonomy, however, is rather superficial as it captures the notational rather than the semantical variations among the various approaches. A more fundamental taxonomy can be drawn along the dimensions I mentioned in the outline, namely, the extensional vs. the intensional approaches. For example, it is possible to use probabilities either extensionally (e.g., in PROSPECTOR [Duda et al., 1976]) or intensionally (e.g., in MUNIN [Andreassen et al., 1987]). Similarly, one can use the Dempster-Shafer notation either extensionally (as in [Ginsberg, 1984]) or intensionally (as in [Lowrance et al., 1986]).

1.5 *Extensional vs. Intensional Approaches*

1.5.1 ***The Role of Connectives*** Extensional systems, a typical representative of which is the certainty-factors calculus used in MYCIN [Shortliffe, 1976], treat uncertainty as a generalized truth value, i.e., the certainty of a formula is defined to be a unique function of the certainties of its subformulas. Thus, the connectives in the formula serve to select the appropriate weight-combining function. For example, the certainty of the conjunction $A \wedge B$ is given by some function (e.g., the minimum, or the product) of the certainty measures assigned to A and B individually. By contrast, in intensional systems, a typical representative of which is probability theory, certainty measures are assigned to sets of worlds and the connectives, too, combine sets of worlds by set theoretical operations. For example, the probability of $P(A \wedge B)$ is given by the weight assigned to the intersection of two sets of worlds, those in which A is true and those in which B is true, but cannot be determined from the individual probabilities $P(A)$ and $P(B)$.

1.5.2 ***What's in a rule?*** Rules, too, have different roles in these two systems. The rules in extensional systems provide licenses for certain symbolic activities. For example, the rule $A \rightarrow B(m)$ may mean: If you see A, then you have the license to update the certainty of B by a certain amount that is a function of the rule strength m. The rules are interpreted as a summary of past performance of the problem solver, describing the way an agent normally reacts to problem situations or to items of evidence. In intensional systems, the rules denote elastic constraints about the world. For example, in the Dempster-Shafer formalism the rule $A \rightarrow B(m)$ does not describe how an agent reacts to the finding of A, but asserts that the set of worlds in which A and $\neg B$ hold simultaneously is rather unlikely and hence should be excluded with probability m.

In the Bayesian formalism the rule $A \rightarrow B(m)$ is interpreted as a conditional probability statement $P(B \mid A) = m$ asserting that among all worlds satisfying A, those that also satisfy B constitute a majority of proportion m. Although there exists a vast difference between these two interpretations (as will be shown in Sections 3.2.2 and 4.1.1), they both represent summaries of factual or empirical information, rather than summaries of past decisions.

2 Extensional Systems: Merits, Deficiencies, and Remedies

2.1 Computational Merits

A good way to present the computational merits of extensional systems is to examine the way rules are handled in the certainty-factors formalism [Shortliffe, 1976] and contrast it with that dictated by probability theory. Figure 2 depicts the combination functions that apply to series and parallel rules, from which one can form a rule-network. The result is a modular procedure for determining the certainty factor of a conclusion, given the credibility of each rule, and the certainty factor of the premises (i.e., the roots of the network). To complete the calculus we also need to define combining functions for conjunctions and negation. However, ignoring mathematical details, the important point to notice is that the same combination function applies uniformly to all rules in the system, regardless of the topology of the network that surrounds them.

Computationally speaking, this uniformity mirrors the modularity of inference rules in classical logic. For example, the logical rule "If A then B" has the following procedural interpretation: "If you see A anywhere in the knowledge base, then, regardless of other things the knowledge base contains, and regardless of how A was derived, you have the license to assert B and add it to the database." This combination of *locality:* "regardless of other things," and *detachment:* "regardless of how it was derived," constitutes the principle of *modularity*. The numerical parameters that decorate the combination functions in Figure 2 do not alter this basic principle. The computational license provided by the rule $A \rightarrow B(m)$ reads: "If you see the certainty of A undergoing a change δ_A, then, regardless of other things the knowledge base contains, and regardless of how δ_A was triggered, you have an unqualified license to modify the current certainty of B by some amount, δ_B, that may depend on m, δ_A, and on the current certainty of B.[2]

2 The observation that the rules refer to changes, rather than absolute values, was made by [Horvitz and Heckerman, 1986].

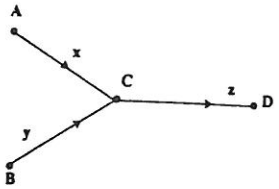
EMYCIN CERTAINTY MANAGEMENT

Rules:

- **If A then C (x)**
- **If B then C (y)**
- **If C then D (z)**

1. **Parallel Combination**

$$CF(C) = \begin{cases} x + y - xy & x, y > 0 \\ (x + y) / (1 - \min(x, y)) & x, y \text{ different sign} \\ x + y + xy & x, y < 0 \end{cases}$$



2. **Series Combination**

$$CF(D) = z \cdot \max(0, CF(C))$$

3. **Conjunction, negation ...**

Figure 2 Functions combining certainty factors in EMYCIN—an extensional system.

To appreciate the power of this interpretation, let us compare it with that given by an intensional formalism such as probability theory. Interpreting rules as conditional probability statements, $P(B \mid A) = p$, does not provide us with a license to do anything. Even if we are fortunate to find A true in the database, we still cannot assert a thing about B or $P(B)$, because the meaning of the statement is: If A is true, and A is the only thing that you know, then you can attach to B a probability p. As soon as we have other facts, K, in the database, the license to assert $P(B) = p$ is automatically revoked, and we need to look up $P(B \mid A, K)$ instead. Therefore, such a statement leaves one totally impotent, unable to initiate any computational activity, unless one can verify that all the other things in the knowledge base are irrelevant. It is for this reason that verification of irrelevancy is so crucial in intensional systems.

In truth, such verifications are also crucial in extensional systems, except that the computational convenience of the latter and their striking resemblance to logical derivations tempts people to neglect the importance of the former. We shall next demonstrate what semantic penalties are paid when relevance considerations are ignored.

2.2 *Semantic Deficiencies*

The price tag attached to the computational advantages of extensional systems is that they often yield incoherent updating, i.e., they are subject to surprises and counter-intuitive conclusions. These surface in several ways; the most notable are:

1. difficulties in retracting conclusions,
2. improper treatment of correlated sources of evidence, and
3. improper handling of bidirectional inferences.

We shall start with the latter.

2.2.1 *The Role of Bidirectional Inferences* The ability to use both predictive and diagnostic information is an important component of plausible reasoning, and improper handling of such information leads to rather strange results. A common pattern of normal discourse is that of *abductive* reasoning: If A implies B, then finding the truth of B makes A more credible [Polya, 1954]. This pattern involves reasoning both ways, from A to B, as well as from B to A. Moreover, it appears that people do not require two separate rules for performing these inferences; the first provides the license to invoke the second. Extensional systems, on the other hand, require that the second rule be stated explicitly and, what is more disturbing, that the first rule be removed. Otherwise, a cycle is created where any slight evidence in favor of A would be amplified via B and fed back to A, quickly turning into a stronger confirmation (of A and B), with no apparent basis. The prevailing practice in such systems (e.g., MYCIN) is to cutoff cycles of that sort, permitting only diagnostic reasoning but no predictive inferences.

Cutting off its predictive component, prevents the system from exhibiting another important pattern of plausible reasoning, one that we name "Explaining away": If A implies B, and C implies B, and B is true, then finding that C is true makes A *less* credible. In other words, finding a second explanation to an item of data, makes the first explanation less credible. Such interaction among multiple causes appears in many applications. When a physician discovers evidence in favor of one disease, this reduces the credibility of other diseases, although the patient may as well be suffering from two or more disorders simultaneously. A suspect who provides an alternative explanation for being at the scene of the crime appears less likely to be guilty, even though the explanation furnished does not preclude his having committed the crime.

To exhibit this sort of reasoning, a system must use bidirectional inferences—from evidence to hypothesis (or explanation), as well as from hypothesis to evidence. While it is sometimes possible to use brute force (e.g., enumerating all exceptions) and restore "explaining away" without the dangers of circular reasoning, we shall see that any system that succeeds in doing that must compromise the principles of modularity, i.e., locality and detachment. More precisely, every system that updates beliefs modularly at the natural rule level and that treats all rules equally, is bound to behave contrarily in prevailing patterns of plausible reasoning.

2.2.2 *The Limits of Modularity* The principle of locality attains its ultimate realization in the inference rules of monotonic logic. The rule "If P then Q" means that if P is found true, we can assert Q with no further analysis, even if the database contains some other knowledge K. In plausible reasoning, the luxury of ignoring the rest of the database can no longer be maintained. For example, suppose we have a rule

R_1 = *"If the ground is wet, then assume it rained (with certainty c_1)."*

Finding the ground wet does not permit us to raise the certainty of "rain" because the knowledge base might contain strange items such as K = "the sprinkler was on last night." These strange items, called *defeaters*, are sometimes easy to discover (as in the case of K' = "the neighbor's grass is dry," which directly opposes "rain"), but sometimes hide cleverly behind syntactical innocence. The neutral fact K = "sprinkler on" neither supports nor opposes "rain," yet K manages to undercut the rule R_1. This undercutting cannot be implemented in an extensional system; once R_1 is invoked, the increase in the certainty of "rain" will never be retracted, because, normally, no rule exists that directly connects "sprinkler on" to "rain." Forcing such a connection by proclaiming "sprinkler on" as an explicit exception to R_1, again defeats the spirit of modularity; it forces the rule-author to pack together items of information that are only remotely related to each other, and, moreover, it loads the rules with an unmanageably large number of exceptions.

Violation of detachment can also be demonstrated in this example. In deductive logic, if K implies P and P implies Q, then finding K true permits us to deduce Q by simple chaining; a derived proposition (P) can trigger a rule with the same vigor as a directly observed proposition. However, chaining does not apply in plausible reasoning. The system may contain two innocent looking rules: "If wet-ground then rain" and "If sprinkler-on then wet-ground"; you find that the sprinkler is on and, obviously, you do not want to conclude that it rained. On the contrary, finding that the sprinkler is on only takes away support from "rain."

As another example, consider the relationships shown in Figure 3. Normally an alarm sound alerts us to the possibility of a burglary. If somebody calls you at the office and tells you that your alarm system is on, you would surely rush home, even though there could be other causes for the alarm. If you further hear a radio announcement that there was an earthquake nearby, and if the last false alarm you recall was triggered by an earthquake, then your certainty of a burglary would diminish. Again, this requires going both ways, from effect to cause (radio $\rightarrow$ earthquake), cause to effect (earthquake $\rightarrow$ alarm), and then back from effect to cause (alarm $\rightarrow$ burglary). However, notice what pattern of reasoning results from such a chain: We have a rule "If A (alarm) then B (burglary)," you listen to the radio, A becomes more credible,

and the conclusion B becomes less credible. Overall, we have: "If $A \rightarrow B$ and A becomes more credible, then B becomes less credible." This behavior is clearly contrary to everything we expect from local belief updating.

In conclusion, we see that the difficulties that plague classical logic do not stem from its nonnumeric, bi-value character. Equally troublesome difficulties emerge when truth and certainty are measured on a gray scale, whether by a point estimate, by interval bounds, or by linguistic quantifiers such as "likely" or "credible." There seems to be a basic struggle between procedural modularity and semantic coherence, independent of the notational system used.

2.2.3 *Correlated Evidence* Extensional systems, greedily exploiting the licenses provided by locality and detachment, respond only to the magnitudes of the weights but not to their origins. As a result they will produce the same conclusions regardless of whether the weights originate from identical or independent sources of information. An example due to Henrion [1986b] helps demonstrate the problems encountered by such local strategy. Figure 4 shows how multiple, independent sources of evidence would normally increase the confirmation of a hypothesis (e.g., "thousands dead"), yet, upon discovering the common origin of these sources, the confirmation should be reduced. Extensional systems are too local to recognize the common origin of the information, and will update the confirmation of the hypothesis as if supported by three independent sources.

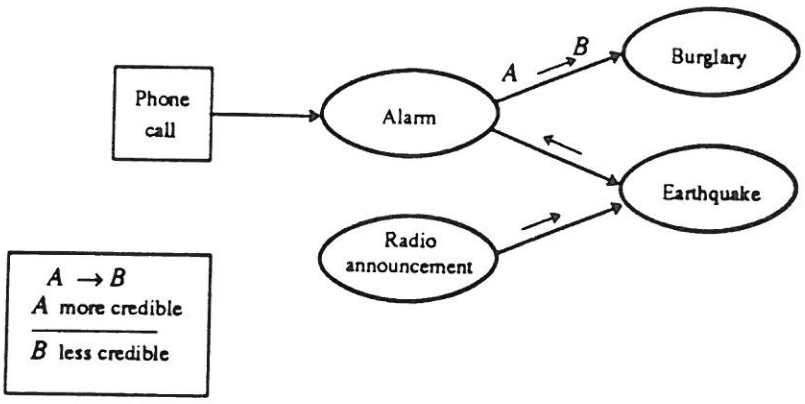


Figure 3 Making the antecedent of a rule more credible can cause the consequent to become less credible.

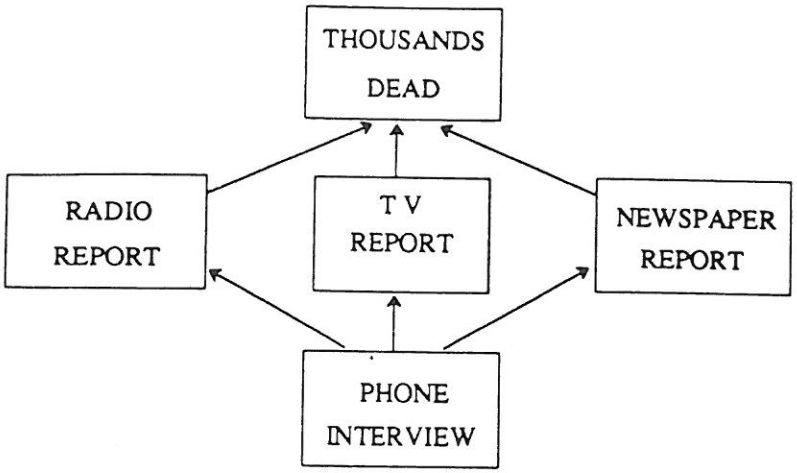


Figure 4 The Chernobyl disaster example (after Henrion) shows why rules cannot combine locally.

2.2.4 *Attempted Remedies and Their Limitations* The developers of extensional systems have proposed and implemented powerful techniques to remedy some of the semantic deficiencies discussed in the preceding subsections. Most have focused on the issue of correlated evidence and fall into two approaches:

1. **Bounds Propagation**—Since most correlations are unknown, certainty measures are combined under two extreme assumptions; one, that the components are highly positively correlated, the other that they are negatively correlated. This gives rise to upper and lower bounds on the combined certainty, which enter as inputs to subsequent computations and produce new bounds on the certainty of the conclusions. This approach has been implemented in INFERNO [Quinlan, 1983] and represents a local approximation to Nilsson's probabilistic logic [Nilsson, 1986].

2. **User-Specified Combination Functions**—Bonissone et al. [1987], in a system named RUM, has permitted the rule-author to specify the combination function that should apply to the rule's components. For example, if a, b, c stand for the weights assigned to propositions A, B, C respectively, in the rule

$$A \wedge B \rightarrow C$$

the user can specify which one of the following three combination functions should be used:

$$T_1(a, b) = \max(0, a + b - 1)$$
$$T_2(a, b) = ab$$
$$T_3(a, b) = \min(a, b)$$

These functions (called T norms) represent the probabilistic combinations obtained under three extreme cases of correlation between A and B: highly negative, zero, and highly positive.

Cohen et al. [1987b], have proposed a more refined scheme, where, for any pair of values, $P(A)$ and $P(B)$, the user is permitted to specify the value of the resulting probability, $P(C)$.

The difficulties with these correlation-handling remedies are several. First, the bounds produced by systems such as INFERNO are too wide. For example, if we are given $P(A) = p$ and $P(B \mid A) = q$ then the bounds we obtain for $P(B)$ are

$$pq \leq P(B) \leq 1 - p(1 - q)$$

that, for small p, approach the unit interval [0, 1]. Second, the user-specified approaches are plagued by the problem that pair-wise correlations are generally not sufficient to handle the intricate dependencies that may occur among rules; higher-order dependencies are often necessary [Bundy, 1985]. Finally, even if one succeeds in specifying higher-order dependencies, a much more fundamental limitation exists: dependencies are dynamic relationships, that are created and destroyed as new evidence obtains. For example, the dependence between a child's shoe size and reading ability is destroyed once we find out the child's age. A dependency between the propositions "it rained last night" and "the sprinkler was on" is created once we find out that the ground is wet. Thus, whatever correlations and/or combination functions are specified at the knowledge-building phase, these may quickly become obsolete once the program is put into use.

Heckerman [1986a, 1986b] delineated precisely the range of applicability of extensional systems of the MYCIN type. He proved that any system that updates certainty weights in a modular and consistent fashion can be given a probabilistic interpretation in which the certainty update of a proposition A is some function of the likelihood ratio

$$\lambda = \frac{P(Evidence \mid A)}{P(Evidence \mid \neg A)}.$$

In MYCIN, for example, the certainty update CF can be interpreted to stand for

$$CF = \frac{\lambda - 1}{\lambda + 1}$$

Once we have a probabilistic interpretation, it is easy to determine the set of structures within which the update procedure will be semantically valid. It turns out that a system of such rules will produce coherent updates if and only if the rules form a directed tree, i.e., no two rules may diverge from the same premise. This limitation explains why strange results were obtained in the burglary example of Figure 3. There the alarm event points to two possible explanations, "Burglary" and "Earthquake," giving rise to two evidential rules diverging from the premise "Alarm," in violation of the tree restriction.

Hajek [1985] and Hajek and Valdes [1987] have developed an algebraic theory that characterizes an even wider range of the extensional systems and combining functions, including, for example, those based on Dempster-Shafer intervals. The unifying properties common to all such systems is that they form an ordered Abelian group. Again, the knowledge base must form a tree in order that no evidence is double counted via alternative paths of reasoning.

2.3 *Conclusions*

Handling uncertainties is a rather tricky enterprise. It requires a very fine balance between our desire to use the computational permissiveness of extensional systems and our ability to refrain from committing semantical sins. It is like crossing a minefield with an untrained wild horse. You can make believe that your horse is smart and, being decorated with certainty weights, will keep you out of trouble. However, the danger is real, and highly skilled knowledge engineers are needed to prevent it from turning into a disaster. The other extreme is to try and work your way by foot with a semantically safe system, such as probability theory, but then you can hardly move—every step seems to require that you examine the entire field, afresh. We shall now examine means for making this movement brisker.

3 *Intensional Systems and Network Representations*

In intensional systems, the syntax consists of declarative statements and, hence, mirrors world knowledge fairly nicely. For example, conditional probability

statements are both empirically testable and conceptually meaningful parameters. Additionally, the problems of handling bidirectional inferences and correlated evidence do not arise; these are obtained as built-in features of one globally coherent model. However, since the syntax does not point to any useful procedures, we need to construct special mechanisms that convert the declarative input into query-answering routines.

A solution, or at least part of a solution, is offered by techniques based on *belief networks*. The idea is to make intensional systems operational by making relevance relationships explicit, thus curing the impotence of declarative statements such as $P(B \mid A) = p$. As we mentioned earlier, the reason one cannot act on the basis of such declarations is that one must first make sure that other things contained in the knowledge base are irrelevant to B, hence can be ignored. The trick is, therefore, to encode knowledge in such a way that the ignorable be recognizable or, better yet, that the nonignorable be quickly identified and readily accessible. Belief networks encode relevancies as neighboring nodes in a graph, thus ensuring that by consulting the neighborhood you have taken everything into account and gain a license to act; what you don't see locally won't matter any way. In summary, what network representations offer is a dynamically updated list of all currently valid permissions to ignore, and permissions to ignore amount to permissions to act.

Network representations are not foreign to AI systems. Most reasoning systems encode relevancies using intricate systems of pointers, i.e., networks of indices that group facts into structures, such as frames, causal chains, and inheritance hierarchies. These structures, while shunned by pure logicians, have proven to be indispensable in practice, because they make the information required to perform an inference task reside "in the vicinity" of the propositions involved in the task. Moreover, many patterns of human reasoning can be explained only by people's tendency to seriously conform to the pathways laid out by such networks.

The special feature of the networks discussed in this survey is that they have clear semantics. In other words, they are not auxiliary devices, contrived to make reasoning more efficient but, rather, are an integral part of the semantics of the knowledge base and, to a certain degree, can even be derived from the knowledge base.

I will first discuss the nature of these networks in two uncertainty formalisms: probability theory, where they are called *Bayesian networks, causal nets*, or *influence diagrams*, and the Dempster-Shafer theory, where they are referred to as *galleries* [Lowrance et al., 1986], *qualitative Markov networks* [Shafer et al., 1987], or *constraint networks* [Montanari, 1974]. In Section 4.1 I will briefly discuss the theory of *graphoids*, which provides an axiomatic characterization of the notion of relevance and its relation to network representations.

3.1 *Evidential Reasoning with Bayesian Networks*

3.1.1 *Network Construction and the Role of Causality* Defined formally, Bayesian networks are directed acyclic graphs in which each node represents a random variable, or uncertain quantity, that can take on two or more possible values. The arcs signify the existence of direct influences between the linked variables, and the strengths of these influences are quantified by forward conditional probabilities. Informally, the structure of a Bayesian network can be determined by a simple procedure: we assign a vertex to each variable in the domain and draw arrows toward each vertex X_i from a select set S_i of vertices perceived to be "direct causes" of X_i. The strength of these direct influences is then quantified by a link matrix $P(x_i \mid s_i)$, that represents (judgmental estimates of) the conditional probabilities of the event $X_i = x_i$, given any value combination s_i of the parent set S_i. The ensemble of these local estimates specifies a complete and consistent global model (i.e., a joint distribution function), on the basis of which all probabilistic queries can be answered. The overall joint distribution function on the variables $X_1, \ldots, X_n$, is given by the product:

$$P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid s_i)$$

So, for example, the joint distribution corresponding to the network of Figure 5 is given by:

$$P(h, e, r, s, d, w, g) = P(h)P(e)P(r \mid e)P(s \mid e, h)P(d \mid s)P(w \mid s)P(g \mid s)$$

where lowercase symbols stand for any particular value (i.e., true or false) of their corresponding variables.

To pacify the mathematicians among us, note that, conversely, the structure of the network can be determined by the joint distribution function, if such is ever available. Once we agree on a total order (e.g., temporal precedence) for the variables involved, the set of parents S_i of variable X_i is chosen from its predecessors by the criterion that

$$P(x_i \mid s_i) = P(x_i \mid x_1, \ldots, x_{i-1})$$

In other words, knowing the parents renders all other predecessors of X_i irrelevant relative to our belief in X_i. In principle, any choice S_i satisfying this criterion will define an adequate network, but, of course, choosing minimal sets of parents will be more efficient, and ordering the variable chronologically would probably result in sparser networks than otherwise.

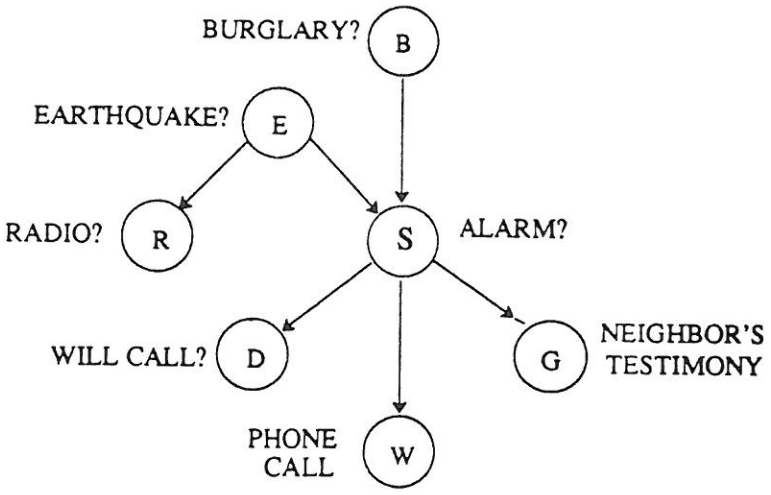


Figure 5 The Bayesian network associated with the burglary alarm story.

Figure 5 depicts the burglary alarm story of Figure 3, with two added variables D and G. D describes the event that your daughter, having been surprised by the alarm, will try to reach you at the office. G stands for the testimony of another neighbor relative to the alarm sound S. The transition from Figure 3 to Figure 5 demonstrates the incremental nature of the process of constructing the knowledge base. Adding the facts about D only requires that one identifies the possible causes of D (in our case, S) and estimates two parameters:

$P(D \mid S)$ = How likely is it that your daughter will try to call, given that she hears the alarm sound, and

$P(D \mid \neg S)$ = How likely is it for her to call, assuming there is no alarm.

The addition of the link $S \rightarrow G$ requires similar parameters, except that, if the testimony G is available (even if it is nonpropositional, say, a lengthy conversation [Pearl, 1987b]), it can be summarized by a single parameter; the likelihood ratio:

$$\lambda = \frac{P(G \mid S)}{P(G \mid \neg S)}$$

The advantage of a network representation is that it allows people to directly express the fundamental qualitative relationship of "direct depend-

ency"; the network then displays a consistent set of many additional direct and indirect dependencies and preserves them as a stable part of the model, independent of the numerical estimates. For example, Figure 5 displays the fact that the radio report *(R)* would not change the prospects of the daughter's phone call *(D)*, once we verify the actual state of the alarm system *(S)*. This fact is conveyed via the network topology—showing S intercepting the path between R and D—despite the fact that it was not considered explicitly during the construction of the network. It can be inferred visually from the linkages used to put the network together and, moreover, will remain part of the model regardless of the numerical estimates that are assigned to the links.

The directionality of the arrows is essential for displaying nontransitive dependencies, e.g., S depends on both E and H and, yet, E and H are independent; they become dependent only if S or any of its descendants is known. Had the arcs been stripped of their arrows, some of these relationships would be misrepresented. This role of identifying what information is or is not relevant in any given state of knowledge is an important feature of causal schemata. In this role, causality serves as a lubricant that modularizes experience. By displaying a high number of legitimate irrelevancies in the domain, causal schemata minimize the number of relationships that need to be considered while the model is constructed. Thus, causality also operationalizes our experience, because modularity authorizes a high number of licenses to perform local inferences. The currently prevailing practice in rule-based expert systems, of encoding knowledge by evidential rules (i.e., if effect then cause), is deficient in this respect. It normally fails to account for intercausal dependencies (e.g., an earthquake explaining away the alarm sound), and if one ventures to encode these interactions by direct rules, legitimate independencies are no longer represented, such as between earthquakes and burglaries (see [Shachter and Heckerman, 1988]).

There is a long and rich tradition of Bayesian belief networks, starting in 1921 with a geneticist named Wright. He developed a method called *path analysis* [Wright, 1934], that later on, became an established representation of causal models in economics [Wold, 1964], sociology [Kenny, 1979; Blalock, 1971] and psychology [Duncan, 1975]. *Influence diagrams* represent another component in this tradition [Howard and Matheson, 1981; Shachter, 1988]. These were developed for decision analysis and contain both event nodes and action nodes. *Recursive models* is the name given to such networks by statisticians seeking meaningful and effective decompositions of contingency tables [Lauritzen, 1982; Wermuth and Lauritzen, 1983; Kiiveri et al., 1984].

The next subsection illustrates the role of networks as a representation capable of converting declarative knowledge to answer-producing procedures. The illustration will focus on Bayesian networks, but similar techniques have been developed for constraint networks in the Dempster-Shafer formalism [Shafer et al., 1987; Kong, 1986].

3.1.2 *Belief Propagation by Message Passing* Since a fully specified Bayesian network constitutes a complete probabilistic model of all variables in the domain, it contains the information necessary to answer all probabilistic queries about these variables. Such queries include, for example, "what are the chances of a burglary, given that the radio announced an earthquake and the daughter did not call?" or "what is the most likely explanation for your daughter's not having called?" Additionally, due to the relevance information conveyed by their links, belief networks can also be used as inference engines, i.e., the nodes can be regarded as processors and the links as communication channels that provide the (storage locations of the) inputs and outputs as well as the timing information necessary for sequencing the computational steps. In other words, many of the computations can be conducted by a local and parallel message-passing process, with minimum external supervision, similar to the derivational steps taken by extensional systems.

The advantages of this distributed, message-passing paradigm is that it provides a natural mechanism for exploiting the independencies embodied in sparsely constrained systems and translating them, by subtask decomposition, into substantial reduction in complexity. Additionally, distributed propagation is inherently "transparent"; namely, the intermediate steps, by virtue of their reflecting interactions only among semantically related variables, are conceptually meaningful. This facilitates the use of natural, object-oriented programming tools and helps establish confidence in the final result.

Distributed schemes for belief *updating* and belief *revision* are described in [Pearl, 1986, 1987a]. Belief updating aims at assigning each variable a posterior probability that correctly accounts for the evidence at hand. The aim of belief revision is to identify a composite set of propositions (one from each variable) that "best" explains the evidence at hand, i.e., attains the highest posterior probability. These involve the updating and transmittal of two types of messages:

> λ—the strength of evidential support that a variable obtains from its descendants, and
>
> π—the strength of causal support that a variable obtains from its nondescendants.

This separation into causal and evidential components permits the execution of bidirectional inferences without the dangers of circular reasoning (see Section 2.2.1).

Figure 6 shows six successive stages of belief propagation through a simple binary tree, assuming that all activities are triggered by changes in the parameters of neighboring processors. Initially (Figure 6a), the tree is in equilibrium, representing the state of belief due to all prior information. As soon as two nodes are activated by new information (Figure 6b), white tokens (representing λ) are placed on their links, directed toward their parents. Activated by

these tokens, the parents compute their degree of belief, and manufacture the appropriate number of tokens for their neighbors (Figure 6c): white tokens for their parents and black tokens (representing π) for the children. (The links through which the absorbed tokens have entered do not receive new tokens, thus reflecting the feature that a π-message is not affected by a λ-message crossing the same link). The root node now receives two white tokens, one from each of its descendants. That triggers the production of two black tokens for top-down delivery (Figure 6d). The process continues in this fashion until, after six cycles, all tokens are absorbed, and the network reaches a new equilibrium, where each variable is assigned a probability measure reflecting the new information.

The updating scheme possesses the following properties:

1. New information diffuses through the network in a single pass, i.e., equilibrium is reached in time proportional to the diameter of the network.
2. The primitive processors are simple, repetitive, and they require no working memory except that used in matrix multiplication.
3. The local computations and the final belief distribution are entirely independent of the control mechanism that activates the individual operations. They can be activated by either data-driven or goal-driven (e.g., requests for evidence) control strategies, by a clock, or at random.

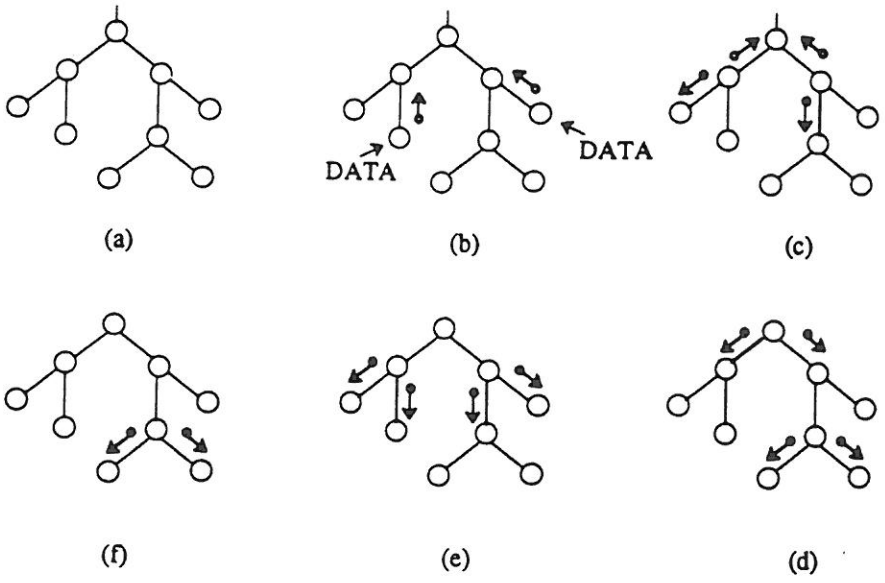


Figure 6 The impact of new data propagates through a tree by a message-passing process.

As soon as a node posts a token for its parent, it is ready to receive new data, and when this occurs, a new token is posted on the link, replacing the old one. In this fashion the network can track a changing environment and provide coherent interpretation of signals emanating simultaneously from multiple sources. Having an efficient mechanism of updating and/or revising beliefs also facilitates various control functions such as, for example, selecting the next best test in diagnosis. This can be done by the method of "hypothesizing"; we imagine what impact the outcome of various tests would have on some target hypothesis, and select the test with the highest impact.

The objective of updating beliefs coherently by purely local computations can be fully realized if the network is singly-connected, i.e., if there is only one undirected path between any pair of nodes. These include trees, where each node has a single parent, as well as networks with multi-parent nodes, representing events with several causal factors, as in Figure 5.

Here the π message transmitted from "Earthquake" to "Alarm" interacts with the λ message that "Alarm" receives from "Phone call" to produce a reduction of the evidential support (λ) the "Alarm" lends to "Burglary." This distinction between causal (π) and evidential (λ) supports identifies the origin of beliefs and permits the system to treat multiple causes differently than multiple symptoms; the former compete with each other, the latter support each other. It is due to this distinction that the system obtains coherent updating via modular computations, dispensing with the need to specify direct inhibitory connections from one cause to another [Pearl, 1988b].

The profile of π and λ messages that load the network at any given time also provides the information needed for generating explanations, similar to the justification network in truth-maintenance systems. Tracing the most influential π and λ messages back to their origins yields a skeletal subgraph from which verbal explanations can be structured, clearly reflecting the distinction between causal and evidential supports.

3.1.3 *Coping with Loops* When loops are present, as in Figure 3, the network is no longer singly-connected, and local propagation schemes invariably run into trouble. Several methods have been developed that extend the propagation method to networks containing loops while still maintaining global coherence relative to probability theory. The most notable are conditioning, clustering, and stochastic simulation.

Before describing each of these methods, one should not overlook a simple but important approximation method called "ignore the loops," namely, propagate the π and λ messages according to the equations developed for a singly-connected network. If loops are present, this strategy will cause the messages to circulate indefinitely until their magnitude becomes insignificantly small (this will always be the case because the conditional probabilities on the links

tend to attenuate the messages). If the loops are long, ignoring them is not going to introduce a significant error because the degree of inter-message correlation, created by multiple paths, diminishes with the lengths of such paths. At any rate, the results obtained after relaxation should be closer to the theoretical results than those obtained by extensional updating strategies, because the latter totally ignore the distinction between causal and evidential supports, while the former account for it in an approximate way.

The method of conditioning involves identifying a set of variables (called cycle cutset) that, if known with certainty, would render the network singly-connected, instantiate these variables to some values, conduct the propagation on the rest of the network, repeat for all possible instantiations, then combine the results by taking their weighted average. In Figure 3, for example, we would run two propagation exercises, one under the assumption "Thousands dead" = true, the other under "Thousands dead" = false. The evidential supports obtained under these two assumptions would then be combined to yield the overall, unconditioned results.

The effectiveness of conditioning depends heavily on the topological properties of the network. In general, the number of instantiations required is 2^c, where c is the size of the cycle cutset chosen for conditioning. Since each propagation phase takes only time linear with the number of variables in the system (n), the overall complexity is exponential with the size of the cycle cutset that we can identify. If the network is sparse, topological considerations can be used to find a small cycle cutset and render the interpretation task tractable.

A second method of sidestepping the loop problem is that of stochastic simulation [Henrion, 1986a]. It amounts to generating a random population of scenarios agreeing with the evidence, then answering queries on the basis of this population. This is accomplished distributedly by having each processor inspect the current state of its neighbors, compute the belief distribution of its host variable, then randomly select one value from the computed distribution, to be inspected by its neighbors in their turn [Pearl, 1987c]. Probabilities are calculated by counting the frequency at which a proposition obtains the value *true*. The advantages of this method are that it is purely distributed, and that the rate of convergence does not depend on the topology of the network. Unfortunately, the rate of convergence deteriorates when the links convey logical constraints, i.e., extreme probabilities [Chin and Cooper, 1987].

The third technique, and currently the most promising, is that of *clustering*. It involves forming local groups of variables in such a way that the topology of the resulting network (treating each group as a single compound node), is singly-connected. For example, grouping the three intermediate nodes in Figure 3 into one compound variable will result in a three-node causal chain. Once a clustered configuration is found, the propagation method described in the preceding subsection is applicable with a processor assigned to each cluster. The complexity of this scheme is exponential with the size of the

largest cluster found, because the processor assigned to manage that cluster must handle that many value combinations (e.g., eight in Figure 3).

A popular method of selecting clusters is to form *join trees*, i.e., trees made up of overlapping clusters in such a way that all links are contained within the clusters. The network of Figure 3, for example, will be decomposed into two overlapping clusters, one comprising the top four nodes, the other the bottom four nodes. The merit of join tree representations have been recognized by statisticians for over 25 years [e.g., Vorobev, 1962; Goodman, 1970; Haberman, 1974]. Their applications to databases are discussed in [Beeri et al., 1983 and Malvestuto, 1986] and they also have been suggested for Bayes inferences [Lemmer, 1983] and constraint processing [Pechter and Pearl, 1987b]. A systematic method of finding such clusters and a thorough analysis of the updating scheme are described in [Lauritzen and Spiegelhalter, 1988]. The method involves triangularizing the network [Tarjan and Yannakakis, 1984], identifying the maximal cliques of the triangularized (or chordal) graph, organizing the cliques in a tree structure, and assigning a processor to each clique. Beliefs can then propagate by the message-passing mechanism described in Section 3.1.2.

The attractive feature of clustering schemes is that, once the clusters are formed and their tree organization established, the resulting structure offers an effective database that can be amortized over many evidential reasoning tasks. A large variety of queries could be answered swiftly by unsupervised, local and parallel processes. Therefore, if one takes seriously the paradigm that unsupervised parallelism is one capability that human learning aspires to achieve [Pearl, 1986], then it is quite reasonable to speculate that the clusters found for join tree representations form the nuclei around which higher cognitive concepts normally evolve.

It is important to note that the difficulties associated with the presence of loops are not unique to probabilistic formulations but are inherent to any problem where globally defined solutions are produced by local computations, be it probabilistic, deterministic, logical, numerical, or hybrids thereof. Identical computational issues arise in Dempster-Shafer's formalism [Kong, 1986], constraint-satisfaction problems [Dechter and Pearl, 1987a], truth-maintenance systems [Doyle, 1979], diagnostic reasoning [Geffner and Pearl, 1987a], relational databases [Beeri et al., 1983], matrix inversion [Tarjan, 1976], and network reliability [Arnborg et al., 1987]. The importance of network representation, though, is that it uncovers the core of these difficulties, and provides a unifying abstraction that encourages the exchange of solution strategies across domains.

3.2 *Dempster-Shafer Theory and Constraint Networks*

Pure Bayesian theory requires the specification of a complete probabilistic model before reasoning can commence, namely, determining for each variable X the conditional probabilities that govern the values of X, given their causal

factors. When a full specification is not available, Bayes practitioners have devised approximate methods of completing the model. For example, if we are given the strength of each individual cause but not the combined impact of several causes, we assume that they combine disjunctively, and that all exceptions are independent [Peng and Reggia, 1986; Pearl, 1987a].

An alternative method of handling partially specified models is provided by the Dempster-Shafer (D-S) theory [Shafer, 1976]. Rather than completing the model, the D-S theory sidesteps the missing specifications, and is resigned instead to less ambitious inference tasks: computing probabilities of provability rather than probabilities of truths. The partially specified model is idealized by qualitative relationships of compatibility constraints, and these qualitative relationships are then used as a logic for assembling proofs of various propositions. Items of evidence are modeled as probabilistic modifications of the available constraints, and the support they lend to a given hypothesis H is defined as the probability that a proof of H can be assembled.

The current popularity of the D-S theory stems both from its readiness to admit partial models and its compatibility with the classical, proof-based style of logical inference. As such, the approach matches the syntax of deductive databases and logic-programming languages but may inherit many of the problems associated with monotonic logic, some of which will be discussed in Section 4.1.1.

3.2.1 *Belief Functions as Probabilities of Provability* I will introduce the D-S theory from a rather unconventional perspective, one that I hope will be more meaningful to AI researchers, especially those versed in constraint processing, truth-maintenance systems and logical programming. Our starting point will be a static network of logical constraints that represents generic knowledge about the world. Each constraint is a declarative statement on a group of variables specifying what is and what is not permitted to hold in the domain. For example the rule $A \rightarrow B$ forbids the simultaneous assignment of *true* to A and *false* to B. A collection of such constraints yields a (possibly empty) set of *extensions* or *solutions*, i.e., assignments of values to all variables that simultaneously satisfy all constraints.

In addition to this static network, we also have items of evidence that provide direct but partial support to a select set of propositions in the system. Each such item of evidence is modeled as a randomly fluctuating constraint, that, for a certain fraction of the time m, imposes the value *true* on the propositions supported by that item. The larger the m the stronger the support. To compute the overall support that several items of evidence impart to a given proposition, say A, we subject the static network to the corresponding set of externally imposed, randomly fluctuating constraints, assume that they act independently of each other, and ask for the probability (or fraction of the time) that A can be proven true. This probability defines the belief function $Bel(A)$, and similarly, a

plausibility function $Pl(A) = 1 - Bel(\neg A)$ is defined by the probability that A is not proven false.

This scheme is illustrated metaphorically in Figure 7. It shows a static network of variables $X, Y, Z, V, \ldots$ (the nodes) interacting via local constraints (the arcs), subject to the influence of two switches that impose additional time varying constraints on various regions of the network. The switches represent two independent items of evidence, each characterized by the fraction of time spent in each position.

To illustrate the analysis of belief functions, let us assume that the static network represents the familiar graph-coloring problem: Each node may take on one of three possible colors, 1, 2, or 3, but no two adjacent nodes may take on identical colors. The position of the switches represents additional constraints e.g., C_{XY}: either X or Y must contain the color 1, or C_Z: Z cannot be assigned the color 2, and so on. The relative time that a switch spends enforcing each of the constraints is indicated by the weight measures $m_1(C_X)$, $m_1(C_{XY})$, $m_2(C_Z)$, and so on. Our objective is to compute $Bel(A)$ and $Pl(A)$, where A stands for the proposition $V = 1$, namely, variable V is assigned the color 1.

Figure 8 represents typical sets of solutions to the coloring problem under different combinations of the switches (the actual values are fictitious).

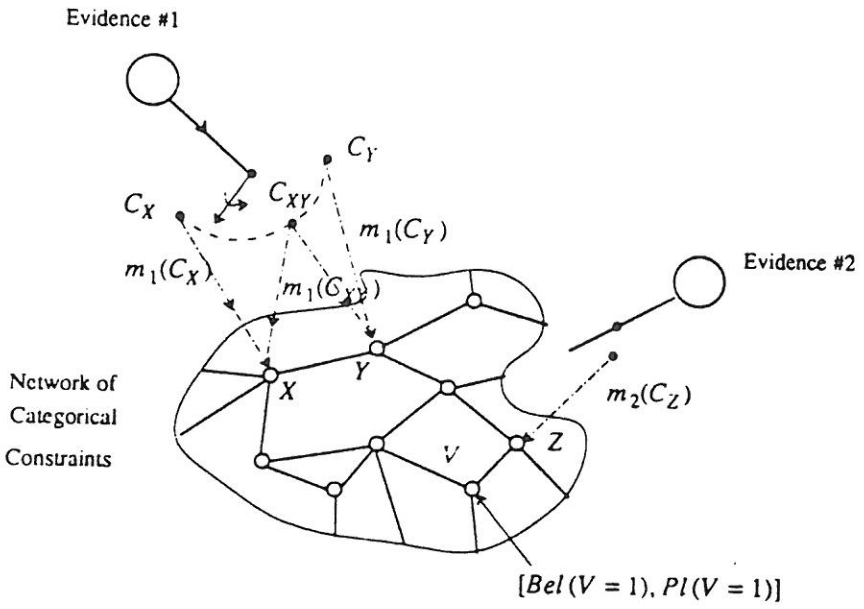
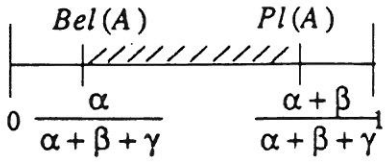


Figure 7 Multiple evidence modeled as random switches, imposing additional constraints on a static network of compatibility relations.

	$VXY \cdots$	
Type-1 positions Time = α	$\begin{bmatrix} 1\,2\,3 \cdots \\ 1\,1\,2 \\ 1\,3\,2 \end{bmatrix}$	$V = 1$ in all solutions
Type-2 positions Time = β	$\begin{bmatrix} 1\,2\,1 \cdots \\ 2\,3\,1 \\ 2\,2\,3 \end{bmatrix}$ $\begin{bmatrix} 3\,2\,1 \cdots \\ 1\,2\,1 \end{bmatrix}$	$V = 1$ and $V \neq 1$ are compatible with each position
Type-3 positions Time = γ	$\begin{bmatrix} 2\,1\,3 \cdots \\ 2\,3\,1 \\ 3\,3\,3 \end{bmatrix}$	$V \neq 1$ in all solutions
Type-4 positions Time = δ	$\begin{bmatrix} \text{Nil} \end{bmatrix}$	no solution

(a)



(b)

Figure 8 (a) Four types of constraints in the graph coloring problem and (b) the resulting belief interval for the proposition *A*: $V = 1$.

Each row represents one extension (or solution) where the entries indicate the value assigned to the variables (columns). The first set of solutions is characterized by having the value 1 assigned to V in each and every row. If the system spends a fraction α of the time in such combinations of switches, we say that $P(e \models A) = \alpha$, namely, the proposition A:"$V = 1$" can be proven true with probability α, given the evidence e. A type-2 position is characterized by the column of V containing 1s as well as alternative values, e.g., 2 or 3. Each such

position (or position combination) is compatible with both A and $\neg A$. Similarly, a type-3 position permits only extensions that exclude $V = 1$, while a type-4 position represents conflict situations; there exists no extension consistent with all the constraints. *Bel(A)* and *Pl(A)* are computed from the time spent in each type of constraint combination:

$$Bel(A) = \frac{\alpha}{\alpha + \beta + \gamma}$$

$$Pl(A) = 1 - Bel(V \neq 1) = 1 - \frac{\gamma}{\alpha + \beta + \gamma} = \frac{\alpha + \beta}{\alpha + \beta + \gamma}$$

These are illustrated as a belief interval in Figure 7(b).

The assumption of evidence independence, coupled with the normalization rule above, leads to an evidence pooling procedure known as *Dempster's Rule of Combination*. For any combination of the evidential constraints, we need to examine the set of extensions permitted by that combination and decide whether the proposition A is entailed by the set, i.e., if every extension contains A and none contain $\neg A$. The total time that a system spends under constraint combinations that compel A, divided by the total time spent in no-conflict combinations, yields *Bel(A)*.

The preceding analysis can be rather complex. The graph-coloring problem, even with only three colors, is known to be NP complete. Moreover, if each item of evidence is modeled by a 2-position switch, and if we have n such switches, then a brute force analysis of *Bel(A)* would require solving 2^n graph-coloring problems. Listing the solutions obtained under every switch combination and identifying those combinations yielding $e \models A$ seems hopeless. Fortunately, two factors help alleviate these difficulties: the sequential nature of Dempster's rule and the ability to exploit certain topological properties of sparse constraint networks. The former permits us to combine evidence incrementally if we store the set of distinct solution sets produced in the past. The latter revolves around the idea of decomposing the network into a tree of clusters, where solutions can be obtained in linear time [Dechter and Pearl, 1987b]. The use of tree decomposition techniques for belief function computations are reported in [Shafer et al., 1987] and [Kong, 1986].

3.2.2 *Comparing Bayes and Dempster-Shafer Formalisms* We see that the D-S theory differs from probability theory in several aspects. First, it accepts an incomplete probabilistic model where some parameters (e.g., the prior or conditional probabilities) are missing. Second, the probabilistic information that is available, like the strength of evidence, is not interpreted as likelihood ratios but rather as random epiphenomena that impose truth values

on various propositions for a certain fraction of the time. This model permits a proposition and its negation to be simultaneously compatible (with the evidence) for a certain portion of the time, and this may permit the sum of their beliefs to be smaller than unity. Finally, due to the incompleteness of the model, the D-S theory does not pretend to provide full answers to probabilistic queries, but rather, is resigned to providing partial answers. It estimates how close the evidence is to forcing the truth of the hypothesis, instead of estimating how close the hypothesis is to being true.

Phrased another way, the D-S theory computes the probability that some set of hypotheses suggested by the evidence would materialize from which the truth of A can be derived out of logical necessity. Thus, instead of the conditional probability $P(A \mid e)$, the D-S theory computes the probability of the logical entailment $e \models A$. The entailment $e \models A$ is not a proposition in the ordinary sense, but a meta-level relationship between e and A, requiring a logical, object-level theory by which proofs from e to A can be assembled. In the D-S scheme the object-level theory consists of categorical *compatibility* constraints, for example, that it is incompatible for an alarm system to turn off unless either a burglary or an earthquake occurred (see Figure 5). It is remarkable that, while the calculation of $P(A \mid e)$, and even the probability of the material conditional $P(e \supset A)$, require complete probabilistic models, $P(e \models A)$ does not.

At this point, it is worthwhile reflecting on the significance of the interval $Pl(A) - Bel(A)$ in the D-S formalism. This interval is often interpreted to portray the degree of ignorance we have about probabilities, namely, the amount of information needed in order to construct a complete probabilistic model. Such intervals would have been a useful supplement to Bayes methods, which always provide point probabilities and so might give one a false sense of security in the model.

Unfortunately, the D-S intervals have little to do with ignorance, nor do they represent *bounds* on the probabilities that would ensue once ignorance is removed. For example, the disappearance of the interval $Pl(A) - Bel(A)$ often vanishes when the model is far from being complete. The equality $Bel(A) = Pl(A)$ simply means that, based on the categorical abstraction captured by the compatibility constraints, the available evidence could not simultaneously be compatible with A and its negation $\neg A$. It is curious to note that applying the same interpretation to noncategorical models yields an interval that *never* vanishes because, barring extreme probabilities, a body of evidence is always compatible with both a proposition and its negation. For example, if in the model of Figure 5 we assume that all rules have exceptions (e.g., there is a nonzero chance of a false alarm, a nonzero chance of a prank phone call, and so on), then all propositions will be assigned zero belief and unit plausibility, because none can actually be *proven* true. Thus, the choice of a categorical abstraction is a crucial one.

3.2.3 *Relations to Truth Maintenance Systems and Incidence Calculus*

The readiness of the D-S formalism to accept knowledge in the form of logical constraints, rather than conditional probabilities, renders it close to uncertainty management technique developed in the logicist camp of AI, most notably truth-maintenance systems (TMS) [Doyle, 1979] and incidence calculus [Bundy, 1985]. These two approaches can be regarded as cousins to the Dempster-Shafer theory because, like the latter, they are based on *provability* as the basic relationship connecting evidence with a conclusion.

Truth-maintenance systems also use logical rules as their elementary units of knowledge, and, similar to our treatment in Section 3.2.1, conclusions are drawn by piecing together rules to form proofs. Likewise, rules may have exceptions that may cause the expected conclusion of the proof to clash with observed facts or with other deductions. However, whereas the exceptions and/or assumptions in the D-S theory were summarized numerically, using the evidence weight m, the TMS approach maintains an explicit list of the main assumptions and exceptions that are involved in each rule.

In the ATMS approach [de Kleer, 1986] one further maintains for each conclusion c, a list $L(c)$ of nonredundant sets of assumptions called *environments*, each of which is sufficient to support a proof of c. Thus $L(c)$ is a Boolean expression whose truth signifies the existence of a proof for c. If we are given probabilities on the assumptions that appear in $L(c)$ and if we further assume that they are independent, then we can obtain $Bel(c)$ by simply computing the probability of $L(c)$:

$$Bel(c) = P[L(c)]$$

Moreover, the computation can be done symbolically, which might be more efficient than the computations method shown in Section 3.2.1. Thus, the ATMS can be used as a symbolic engine for computing the belief functions sought by the D-S theory. Steps in this direction have been taken by D'Ambrosio [1987].

Incidence calculus [Bundy, 1985] suggests a method of computing belief functions by logical sampling, similar in spirit to the method of stochastic simulation [Henrion, 1986a; Pearl, 1987c]. A probabilistic model is used to generate random samples of truth values (bit strings) for a select set of propositions representing uncertain facts. These values are presented as assumptions, or axioms, to a theorem prover. Different sets of assumptions give rise to different theorems and $Bel(c)$ is given by that fraction of the time that c can be proven. This scheme is a physical embodiment of the random switch model described in Figure 7. The random position of each switch is replaced by a random bit string assigned to the propositions impacted by the evidence.

The advantage of this scheme is that the theorem prover can be general purpose (e.g., First Order Logic), not limited to propositional constraint net-

works. Moreover, the scheme is not limited to simulating independent switches; dependencies can be simulated by having the bit strings generated by a complete probabilistic model (e.g., a causal network) in which these dependencies are encoded.

4 Lessons and Open Issues

4.1 Relations to Nonmonotonic Logic

4.1.1 Softened Logic vs. Hardened Probabilities The ills of monotonic logic have often been attributed to its coarse and sharp, bi-valued character. Indeed, when one tries to figure out why logic would not predict the obvious fact that penguin birds do not fly, the first thing that one tends to blame is the sharp, uncompromising stance of the rule "birds fly" toward exceptions. It is natural, therefore, to assume that once we soften the constraints of Boolean logic and allow truth values to be measured on a gray scale, these problems will disappear. There have been several attempts along this line. Rich [1983] has proposed a likelihood-based interpretation of default rules, managed by certainty-factors calculus. Ginsberg [1984], and Baldwin [1987] have, likewise, pursued similar aspirations using the Dempster-Shafer notion of belief functions. While these attempts produce valuable results, revealing, for instance, how sensitive a conclusion is to the uncertainty of its premises, the fundamental problem of monotonicity remains unresolved. For example, regardless of the certainty calculus used, these analyses always yield an increase in the belief that penguins can fly, if one adds the superfluous information that penguins are birds and birds normally fly. Identical problems surface in the use of incidence calculus and softened versions of truth-maintenance systems [D'Ambrosio, 1987].

Evidently, it is not enough to add a soft probabilistic veneer on top of a system that is basically structured after hard monotonic logic. The problem with monotonic logic lies not in the hardness of its truth values, but rather in its inability to process context-dependent information. Logic does not have a device equivalent to the conditional probability statement "$P(B \mid A)$ is high," whose main function is to identify the context A where the proposition B can be believed, and to make sure that only legitimate changes in that context (e.g., going from A = penguins to A' = bird-penguins or A'' = white penguins) will be permitted without significant changes in the belief of B.

Lacking an appropriate logical device for conditionalization, the natural tendency is to interpret the English sentence "If A then B" as a softened version of the material implication constraint $A \supset B$. A useful consequence of such softening is allaying the fears of outright contradictions. For example,

while the classical interpretation of the three rules: "penguins do not fly," "penguins are birds" and "birds fly," yields an unforgivable contradiction, the uncertainties attached to these rules now render them manageable. Still, they are managed in the wrong way, because the material implication interpretation of if-then type rules is so fundamentally wrong that its maladies cannot be rectified by simply allowing exceptions in the form of shaded truth values. The source of the problem lies in the property of transitivity, $(a \rightarrow b\ ,\ b \rightarrow c) \Rightarrow a \rightarrow c$, that is inherent to the material-implication interpretation.

There are occasions where rule transitivity must be totally suppressed, not merely weakened, or else strange results will surface. One such occasion occurs in property inheritance, where subclass specificity should override superclass properties. Another occurs in causal reasoning where predictions should not trigger explanations, (e.g., "sprinkler-on" predicts "wet-ground," "wet-ground" suggests "rain," yet "sprinkler-on" should not suggest "rain"). In such cases, softening the rules only weakens the flow of inference through the rule chain but does not bring it to a dead halt, as it should.

Apparently, what is needed is a new interpretation of "if-then" statements, one that does not destroy the context-sensitive character of probabilistic conditionalization. McCarthy [1986] remarks that circumscription indeed provides such an interpretation. In his words:

> Since circumscription doesn't provide numerical probabilities, its probabilistic interpretation involves probabilities that are either infinitesimal, within an infinitesimal of one, or intermediate—without any discrimination among the intermediate values. The circumscriptions give conditional probabilities. Thus we may treat the probability that a bird can't fly as an infinitesimal. However, if the rare event occurs that the bird is a penguin, then the conditional probability that it can fly is infinitesimal, but we may hear of some rare condition that would allow it to fly after all.

Rather than contrive new logics and hope that they match the capabilities of probability theory, an alternative approach would be to start with probability theory and, if we can't get the numbers or find their use inconvenient, we can extract qualitative approximations as idealized abstractions of the latter, while preserving its context-dependent properties. In this way, nonmonotonic logics should crystallize that are guaranteed to capture the context-dependent features of natural defaults [Pearl, 1988a].

4.1.2 *The Logic of "Almost True"* This program had in fact been initiated over twenty years ago by the philosopher Ernest Adams [1966] who developed a logic of conditionals based on probabilistic semantics. The sentence "If A then B" is interpreted to mean that the conditional probability of B given A is very close to 1, short of actually being 1. An adaptation of Adams'

logic to default schema of the form *Bird (x)* $\rightarrow$ *Fly (x)*, where x is a variable, is reported in [Geffner and Pearl, 1987b]. The resulting logic is nonmonotonic relative to learning new facts, in accordance with McCarthy's desiderata. For example, learning that Tweety is a bird would yield the conclusion that Tweety can fly; subsequently learning that Tweety is also a penguin would yield the opposite conclusion: Tweety can't fly. Further, learning that Tweety is white will not alter this belief, because white is a typical color for penguins. However, and this is where it falls short of expectations, learning that Tweety is clever would cause Adams' logic to retract all previously held beliefs about Tweety's flying and answer: "I don't know." The logic is so conservative that it never jumps to conclusions that some new rule schema might invalidate (e.g., that clever penguins can fly). In other words, the logic does not capture the usual convention that, unless we are told otherwise, properties are presumed to be *irrelevant* to each other.[3]

Attempts to enrich Adams' logic with *relevance*-based features are described in [Pearl, 1987d], [Geffner and Pearl, 1987b], and [Geffner, 1988]. The idea is to follow a default strategy similar to that of belief networks (Section 3.1); dependencies exist only if they are mentioned explicitly or if they logically follow from other explicit dependencies. However, whereas the stratified method of constructing belief networks ensures that all relevant dependencies are already encoded in the network, this can no longer be assumed when knowledge is presented in the form of isolated default rules and logical constraints. A new logic is needed to tell us when one relevancy follows from others. This issue is further discussed in the Section 4.2.

4.1.3 *The Issue of Consistency* There is another dimension along which probabilistic analysis can assist current research in nonmonotonic logics. The latter do not provide any criterion for testing whether a database comprising default rules is internally consistent. The prevailing attitude is that once we tolerate exceptions we might as well tolerate anything [Brachman, 1985]. However, there is a sharp qualitative difference between exceptions and outright contradictions. For example, the statement "red penguins can fly" can be accepted as a description of a world in which redness defines an abnormal type of penguins. However, the statements "typically birds fly" and "typically birds do not fly" stand in outright contradiction to each other; since there is no world in which the two can hold simultaneously, they will invariably lead to strange, inconsistent conclusions. While such obvious contradictions can easily be removed from the database (e.g., [Touretzky, 1986]), more subtle ones might escape detection, e.g., "birds fly," "birds are feathered animals," "feathered animals are birds," and "feathered animals do not fly."

3 Grosof [1986] discusses this convention in terms of a principle of maximizing conditional independencies, similar in spirit to the maximum entropy principle [Cheeseman, 1983].

Adams' logic provides a criterion for detecting such inconsistencies, in the form of three axioms that should never be violated. In inheritance hierarchies this criterion yields a simple graphical test [Pearl, 1987e] that is a generalization of Touretzky's: A network N is consistent iff for every pair of conflicting rules $p_1 \rightarrow q$ and $p_2 \rightarrow \neg q$, p_1 and p_2 are distinct and there is no cycle of rules that embraces both p_1 and p_2. For more intricate structures of default rules the test becomes more involved.

4.2 *Graphoids and the Formalization of Relevance*

A central requirement in several topics of this survey has been to articulate the conditions under which one item of information is considered relevant to another, given what we already know, and to encode knowledge in structures that vividly display these conditions as the knowledge undergoes changes. Different formalisms give rise to different definitions of relevance. For example, in probability theory, relevance is identified with dependence; in constraint-based formalisms (and in relational databases) relevance is associated with induced constraints—two variables are said to be relevant to each other if we can restrict the range of values permitted for one by constraining the other.

The essence of relevance can be identified with a structure common to all these formalisms. It consists of four axioms that convey the simple idea that when we learn an irrelevant fact, the relevance relationships of all other propositions remain unaltered; any information that was irrelevant remains irrelevant and that which was relevant remains relevant. Structures that conform to these axioms are called *graphoids* [Pearl and Paz, 1987]. Interestingly, both undirected graphs and directed acyclic graphs conform to the graphoids axioms (hence the name) if we associate the sentence "variable x is irrelevant to variable y once we know z" with the graphical condition "every path from x to y is intercepted by the set of nodes corresponding to z." (A special definition of "intercept" is required for directed graphs.)

With this perspective in mind, graphs, networks, and diagrams can be viewed as inference engines devised for efficiently representing and manipulating relevance relationships: The topology of the network is assembled from a list of local relevance statements (e.g., direct dependencies), this input list entails (using the graphoids axioms) a host of additional statements, and the function of the graph is to ensure that a substantial portion of the latter can be read off by simple graphical criteria. Such a mapping will enable one to determine, at any state of knowledge z, which information is relevant to the task at hand and which can be ignored. Permissions to ignore, as we saw in Section 3.1, are the fuel that gives intensional systems the power to act.

An important result from the theory of graphoids states that Bayesian networks constitute a sound and complete inference mechanism relative to probabilistic dependencies, i.e., it identifies, in polynomial time, each and every

conditional-independence relationship that logically follows from those used in the construction of the network [Pearl and Verma, 1987; Geiger and Pearl, 1988]. Similar results hold for other types of relevance relationships, e.g., partial correlations and constraint-based dependencies. However, the essential requirement for soundness and completeness is that the network be constructed *causally*, i.e., that we specify, recursively, the relationship of each variable to its predecessors in some total order. (Once the network is constructed, the original order can be forgotten; only the partial order displayed in the network matters).

One can speculate whether it is this soundness-completeness feature that renders causal schemata so important in knowledge organization. More generally, the precise relationship between causality as a representation of irrelevancies and causality as a commitment to a particular inference strategy (e.g., chronological ignorance [Shoham, 1986]) is yet to be fully investigated. A different notion of relevance has been explored by Subramanian and Genesereth [1987], based on logical derivability. The latter takes propositions, rather than variables, as the atomic entities in the relevance relationships, and, again, the connection to graphoid structures is not fully understood.

References

Adams, E., 1966. Probability and the Logic of Conditionals. In *Aspects of Inductive Logic*, J. Hintikka and P. Suppes, ed. North-Holland, Amsterdam.

Andreassen, S., Woldbye, M., Falck, B., and Andersen, S. K., 1987. MUNIN—A Causal Probabilistic Network for Interpretation of Electromyographic Findings. In *Proceedings of the Tenth International Joint Conference on AI*, Milan, Italy. pp. 366–372.

Arnborg, S., Corneil, D. G. and Proskurowski, A., 1987. Complexity of Finding Embeddings in a K-Tree. *SIAM Journal on Algebraic and Discrete Methods* **8**(2):277–284.

Baldwin, J. F., 1987. Evidential Support Logic Programming. *Fuzzy Sets and Systems* **24**:1–26.

Beeri, C., Fagin, R., Maier, D., and Yannakakis, M., 1983. On the Desirability of Acyclic Database Schemes. *Journal of ACM* **30**:479–513.

Ben-Bassat, M., Carlson, R. W., Puri, V. K., Lipnick, E., Portigal, L. D., and Weil, M. H., 1980. Pattern-based Interactive Diagnosis of Multiple Disorders: The MEDAS System. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-2**(2):148–160.

Blalock, H. M., 1971. *Causal Models in the Social Sciences*. London, Macmillan.

Bonissone, P. P., Gans, S. S., and Decker, K. S., 1987. RUM: A Layered Architecture for Reasoning with Uncertainty. In *Proceedings of the Tenth International Joint Conference of Artificial Intelligence*. Milan, Italy. pp. 891–898.

Brachman, R. J., 1985. I Lied About the Trees, or, Defaults and Definitions in Knowledge Representation. *AI Magazine* **6**(3):80–93.

Bundy, A., 1985. Incidence Calculus: A Mechanism for Probabilistic Reasoning. *Journal of Automated Reasoning* **1**:263–283.

Chandrasakaran, B., and Mittal, S., 1983. Conceptual Representation of Medical Knowledge for Diagnosis by Computer: MDX and Related Systems. *Advances in Computers* **22**:217–293.

Cheeseman, P., 1983. A Method of Computing Generalized Bayesian Probability Values for Expert Systems. In *Proceedings of the Sixth International Joint Conference on AI*, Karlsruhe, W. Germany. pp. 198–202.

Chin, H. L. and Cooper, G. F., 1987. Stochastic Simulation of Bayesian Belief Networks. In *Proceedings of the Uncertainty in AI Workshop*, Seattle, Washington. pp. 106–113.

Clancey, W. J., 1985. Heuristic Classification. *Artificial Intelligence* **27**(3):289–350.

Cohen, P. R., 1985. *Heuristic Reasoning about Uncertainty: An Artificial Intelligence Approach*. Pitman, Boston.

Cohen, P., Day, D., Delisio, J., Greenberg, M., Kjeldsen, R., Suthers, D., and Berman, P., 1987a. Management of Uncertainty in Medicine. *International Journal of Approximate Reasoning* **1**(1):103–116.

Cohen, P. R., Shafer, G., and Shenoy P. P., 1987b. Modifiable Combining Functions. In *Proceedings of the Uncertainty in AI Workshop*. Seattle, Washington. pp. 10–21.

D'Ambrosio, B., 1987. Truth Maintenance with Numeric Certainty Estimates. In *Proceedings of the 3rd Conference on AI Applications*. Orlando, Florida, 244–249.

de Kleer, J., 1986. An Assumption-Based Truth Maintenance System. *Artificial Intelligence* **29**:241–288.

Dechter, R., and Pearl, J., 1987a. Network-Based Heuristics for Constraint-Satisfaction Problems. *Artificial Intelligence* **34**(1).

Dechter, R., and Pearl, J., 1987b. *Tree-Clustering Schemes for Constraint-Processing*. UCLA Cognitive Systems Laboratory Technical Report 870054 (R-92). Also in *Proceedings of AAAI-88*. Minneapolis, Minnesota.

Doyle, J., 1979. A Truth Maintenance System. *Artificial Intelligence* **12**(3).

Duda, R. O., Hart, P. E., and Nilsson, N. J., 1976. Subjective Bayesian Methods for Rule-Based Inference Systems. In *Proceedings of the National Computer Conference*. AFIPS. **45**:1075–1082.

Duncan, O. D., 1975. *Introduction to Structural Equation Models*. New York, Academic Press.

Geffner, H., and Pearl, J., 1987a. An Improved Constraint-Propagation Algorithm for Diagnosis. In *Proceedings of the Tenth International Joint Conference on AI*. Milan, Italy. pp. 1105–1111.

Geffner, H., and Pearl, J., 1987b. *A Sound Framework for Reasoning with Defaults*. UCLA Cognitive Systems Laboratory Technical Report 870058 (R-94).

Geffner, H., 1988. On the logic of defaults. In *Proceedings of AAAI-88*. Minneapolis, Minn.

Geiger, D. and Pearl, J., 1988. On the logic of influence diagrams. In *Proceedings of AAAI-88 Workshop on Uncertainty in AI*. Minneapolis, Minn.

Ginsberg, M. L., 1984. Nonmonotonic Reasoning Using Dempster's Rule. In *Proceedings, 3rd National Conference on AI*. AAAI-84. Austin, Texas. pp. 126–129.

Goodman., 1970. The Multivariate Analysis of Qualitative Data: Interaction among Multiple Classifications. *Journal of the American Statistics Association* **65**:226–256.

Grosof, B. N., 1986. Nonmonotonicity in Probabilistic Reasoning. In *Proceedings of AAAI Workshop on Uncertainty in AI*. Philadelphia, Pennsylvania. pp. 91–98.

Haberman, S. J., 1974. *The General Log-Linear Model*. Ph.D. thesis, Department of Statistics, University of Chicago.

Hajek, P., 1985. Combining Functions for Certainty Degrees in Consulting Systems. *International Journal Man-Machine Studies*. **22**:59–65.

Hajek, P., and Valdes, J. J., 1987. *Algebraic Foundations of Uncertainty Processing in Rule-Based Expert Systems*. Ceskoslovenka Akademie Ved, Matematicky Ustav.

Heckerman, D., 1986a. A Probabilistic Interpretation for MYCIN's Certainty Factors. In *Uncertainty in Artificial Intelligence*. North-Holland, Amsterdam.

Heckerman, D., 1986b. *A Rational Measure of Confirmation. Medical Computer Science Group*. Technical Report, Memo-KSL-86-25. Stanford University.

Henrion, M., 1986a. *Propagation of Uncertainty by Logic Sampling in Bayes Networks*. Technical Report, Department of Engineering and Public Policy, Carnegie-Mellon.

Henrion, M., 1986b. Should We Use Probability in Uncertain Inference Systems? In *Proceedings, Cognitive Science Society Meeting*. Amherst. pp. 320–330.

Horvitz., E. J. and Heckerman, D. E., 1986. The Inconsistent Use of Measures of Certainty in Artificial Intelligence Research. In *Uncertainty in Artificial Intelligence*. Kanal, L., Lemmer J., ed. North-Holland, Amsterdam. pp. 137–151.

Howard, R. A., and Matheson, J. E., 1981. Influence Diagrams. In *Principles and Applications of Decision Analysis*. Menlo Park, California: Strategic Decisions Group.

Kanal, L. N., and Lemmer, J. F., ed., 1986. *Uncertainty in Artificial Intelligence*. North-Holland, Amsterdam.

Kenny, D. A., 1979. *Correlation and Causality*. John Wiley and Sons

Kiiveri, H., Speed, T. P., and Carlin, J. B., 1984. Recursive Causal Models. *Journal of Australian Math Society* **36**:30–52.

Kong, A., 1986. *Multivariate Belief Functions and Graphical Models*. Ph.D. Thesis, Department of Statistics, Harvard University.

Lauritzen, S. L., 1982. *Lectures on Contingency Tables*. Second edition. University of Aalborg Press, Aalborg, Denmark.

Lauritzen, S. L., and Spiegelhalter, D. J., 1988. Local Computations with Probabilities on Graphical Structures and their Applications to Expert Systems. To appear in *Journal of the Royal Statistics Society Bulletin*. **50**.

Lemmer, J., 1983. Generalised Bayesian Updating of Incompletely Specified Distributions. *Large Scale Systems* **5**:51–68.

Lowrance, J. D., Garvey, T. D., and Strat, T. M., 1986. A Framework for Evidential-Reasoning Systems. In *Proceedings of the Fifth National Conference on AI*. AAAI-86, Philadelphia, Pennsylvania,. pp. 896–901.

Malvestuto, F. M., 1986. Decomposing Complex Contingency Tables to Reduce Storage Requirements. In *International Workshop on Scientific and Statistical Database Management*. R. Cubitt et al., ed. Luxembourg. pp. 66–71.

McCarthy, J., 1986. Applications of Circumscription to Formalizing Common-Sense Knowledge. *Artificial Intelligence* **28**(1):89–116.

Miller, R. A., Poole, H. E., and Myers, J. P., 1982. INTERNIST-1, An Experimental Computer-Based Diagnostic Consultant for General Internal Medicine. *New England Journal of Medicine* **307**(8):468–470.

Montanari, U., 1974. Networks of Constraints, Fundamental Properties and Applications to Picture Processing. *Information Science* 7:95–132.

Nilsson, N., 1986. Probabilistic Logic. *Artificial Intelligence*. **28**(1):71–87.

Quinlan, J. R., 1983. Inferno: A Cautious Approach to Uncertain Inference. *The Computer Journal* **26**:255–269.

Pearl, J., 1986. Fusion, Propagation and Structuring in Belief Networks. *Artificial Intelligence* **29**(3):241–288.

Pearl, J., 1987a. Distributed Revision of Composite Beliefs. *Artificial Intelligence* **33**(2):173–215.

Pearl, J., 1987b. Bayes Decision Methods. *Encyclopedia of AI*. Wiley Interscience, New York. pp. 48–56.

Pearl, J., 1987c. Evidential Reasoning Using Stochastic Simulation of Causal Models. *Artificial Intelligence* **32**(2):245–258.

Pearl, J., 1987d. *Probabilistic Semantics for Inheritance Hierarchies with Exceptions*. UCLA Cognitive Systems Laboratory Technical Report 870052 (R-93). Also in [Pearl, 1988a].

Pearl, J., 1987e. *Deciding Consistency in Inheritance Networks*. UCLA Cognitive Systems Laboratory Technical Report 870053 (R-96).

Pearl, J., 1988a. *Networks of Belief: Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann Publishers, San Mateo, California.

Pearl, J., 1988b. Embracing Causality in Formal Reasoning. *Artificial Intelligence* **35**(2):259–271.

Pearl, J., and Paz, A., 1987. Graphoids: A Graph-Based Logic for Reasoning about Relevance Relations. In *Advances in Artificial Intelligence-II*. B. Du Boulay et al., ed. North-Holland, Amsterdam.

Pearl, J., and Verma, T., 1987. The Logic of Representing Dependencies by Directed Graphs. In *Proceedings of the AAAI Conference*. Seattle, Washington. pp. 374–379.

Peng, Y., and Reggia, J., 1986. Plausibility of Diagnostic Hypotheses. In *Proceedings of the Fifth National Conference on AI*. AAAI-86. pp. 140–145.

Perez, A., and Jirousek, R., 1985. Constructing an Intensional Expert Systems (INES). In *Medical Decision Making*. Elsevier Scientific Publishers. pp. 307–315.

Polya, G., 1954. *Patterns of Plausible Inference*. Princeton University Press.

Prade, H., 1983. A Synthetic View of Approximate Reasoning Techniques. In *Proceedings of the Eighth International Joint Conference of Artificial Intelligence*. Karlsruhe, West Germany. pp. 130–136.

Rich, E., 1983. Default Reasoning as Likelihood Reasoning. In *Proceedings of the International Joint Conference of Artificial Intelligence*. pp. 348–351.

Shachter, R. D. and Heckerman, D. V., 1987. A Backward View for Assessment. *AI Magazine* **8**(8):55–62.

Shachter, R. D., 1988. Probabilistic Inference and Influence Diagrams. To appear in *Operations Research*.

Shafer, G., 1976. *Mathematical Theory of Evidence*. Princeton University Press.

Shafer, G., Shenoy, P. P., and Mellouli, K., 1987. Propagating Belief Functions in Qualitative Markov Trees, working paper no. 190. To appear in *International Journal of Approximate Reasoning*.

Shoham, Y., 1986. Chronological Ignorance: Time, Nonmonotonicity, Necessity and Causal Theories. In *Proceedings of AAAI-86*. Philadelphia, pp. 389–393.

Shortliffe, E. H., 1976. *Computer-Based Medical Consultation:* MYCIN. Elsevier Scientific Publishers.

Stephanou, H., and Sage, A., 1987. Perspectives on Imperfect Information Processing. *IEEE Transactions on Systems, Man, and Cybernetics* **SMC-17**(5):780–798.

Subramanian, D., and Genesereth, M., 1987. The Relevance of Irrelevance. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*. Milan, Italy. pp. 416–422.

Tarjan, R. E., 1976. Graph Theory and Gaussian Elimination. In *Sparse Matrix Computations*. D. J. Rose, ed. Academic Press, New York. pp. 3–22.

Tarjan, R. E., and Yannakakis, M., 1984. Simple Linear-Time Algorithms to Test Chordality of Graphs, Test Acyclicity of Hypergraphs, and Selectively Reduce Acyclic Hypergraphs. *SIAM Journal on Computing* **13**:566–579.

Thompson, T. R., 1985. Parallel Formulation of Evidential Reasoning Theories. In *Proceedings of the Eighth International Joint Conference of Artificial Intelligence*. Los Angeles, California. pp. 321–327.

Touretzky, D. S., 1986. *The Mathematics of Inheritance Systems*. Morgan Kaufmann Publishers, San Mateo, California.

Vorobev, N. N., 1962. Consistent Families of Measures and Their Extensions. *Theory of Probability and Applications*. **7**:147–163.

Wermuth, N., and Lauritzen, S. L., 1983. Graphical and Recursive Models for Contingency Tables. *Biometrika* **70**:537–552.

Wold, H., 1964. *Econometric Model Building*. North-Holland, Amsterdam.

Wright, S., 1921. Correlation and Causation. *Journal Agricultural Research* **20**:557–585.

Wright, S., 1934. The Method of Path Coefficients. *Ann. Math. Statist.* **5**:161–215.