

A Valuation-Based Language for Expert Systems

Prakash P. Shenoy

School of Business, University of Kansas, Lawrence, Kansas

ABSTRACT

A new language based on valuations is proposed as an alternative to rule-based languages for constructing knowledge-based systems. Valuation-based languages are superior to rule-based languages for maintaining consistency in the knowledge base, for caching inferences, for managing uncertainty, and for nonmonotonic reasoning. An abstract description of a valuation-based language is given. Two specific instances of valuation-based languages are described. The first is designed to represent categorical knowledge. The ability of such a language to maintain consistency and cache inferences is demonstrated with an example. The second is an evidential language—a valuation-based language in which valuations are belief functions. The ability of evidential languages to perform nonmonotonic reasoning and manage uncertainty is demonstrated with an example.

KEYWORDS. *valuation-based language, rule-based language, valuation system, knowledge-based system, rule-based system, consistency in knowledge bases, caching inferences, truth maintenance systems, evidential systems, nonmonotonic reasoning, management of uncertainty*

INTRODUCTION

This paper proposes a new language based on “valuations” as an alternative to rule-based languages for building knowledge-based systems. This language is inspired by the axiomatic framework for propagation of probabilities and belief functions (Shenoy and Shafer [1, 2]) and by its extension, which includes constraint propagation and discrete optimization (Shenoy and Shafer [3, 4]). Since the primary objects in the axiomatic framework are called valuations, we refer to this language as being valuation-based, and we call a formal structure created using this language a valuation system.

A popular language for building a knowledge-based system is a production or a rule-based language (Brownston et al. [5], Davis and King [6]). While these

Address correspondence to P. P. Shenoy, School of Business, University of Kansas, Summerfield Hall, Lawrence, Kansas 66045-2003

languages have many attractive features, they also have some serious shortcomings. In this paper we will focus on four major shortcomings of rule-based languages that are not shared by valuation-based languages. These four shortcomings are referred to as the problem of consistency, the problem of caching, the problem of nonmonotonic reasoning, and the problem of managing uncertainty.

A special case of a valuation system is an evidential network (Shenoy and Shafer [1, 2]). The use of evidential networks to manage uncertainty is well understood (see, e.g., Shafer et al. [7]). However, the use of valuation systems for representing categorical knowledge, maintaining consistency, and performing intelligent caching and nonmonotonic reasoning is not widely understood.

Valuation systems include as a special case belief networks and moral graphs. Belief networks have been proposed by Pearl [8, 9] and moral graphs by Lauritzen and Spiegelhalter [10] for managing uncertainty using probabilities (see also Heckerman and Horvitz [11]). The use of valuation systems in representing and propagating probabilities is described in Shenoy and Shafer [1, 2] and Shafer and Shenoy [12, 13].

Valuation languages can also be used to propagate constraints (Seidel [14], Dechter and Pearl [15], Shenoy and Shafer [3]) and to solve discrete optimization problems, both constrained and unconstrained (Bertele and Brioschi [16], Shenoy and Shafer [4]). Other problems that fit in the framework of valuation languages include solution to systems of equations (Rose [17]), propagation of Spohnian belief functions (Spohn [18], Hunter [19]), retrieval from acyclic database schemes (Malvestuto [20], Beerl et al [21]), and use of a Kalman filter (Dempster [22], Meinhold and Singpurwala [23]).

An outline of this paper is as follows. In the following section we discuss some problems with rule-based languages. In the third section we give an abstract description of a valuation-based language, and in the fourth section we describe a specific instance of a valuation language designed to represent categorical knowledge. We demonstrate, using an example, how such a language can be used to maintain consistency in a knowledge base and how inferences are cached. In the fifth section we describe an evidential language—a valuation language that uses belief functions as valuations. We also briefly describe a truth maintenance system and show the correspondence between concepts in truth maintenance systems and concepts in evidential systems. Next, using an example, we show how evidential languages can be used to reason nonmonotonically and manage uncertainty. We conclude with a summary and some general comments.

SOME PROBLEMS WITH RULE-BASED LANGUAGES

In this section, we look at some of the shortcomings of pure rule-based languages. In particular, we focus on the problems of consistency, caching,

nonmonotonic reasoning, and management of uncertainty. Since there is no universally accepted formal definition of a rule-based language, we will use the model of a pure production system given in Davis and King [6] as a representative rule-based language

Consistency

In large knowledge bases, consistency is an important issue. By consistency, we mean the absence of syntactic contradictions. An example of a syntactic contradiction is a premise $A = a$ and two rules: $A = a \rightarrow B = b$ and $A = a \rightarrow B = \sim b$. (The symbol $\rightarrow$ denotes the truth-functional conditional.)

Rule-based languages lack expressive power to check for contradictions. Accordingly, most commercial implementations of rule-based languages provide little or no support for checking for contradictions. However, this does not mean that such checking cannot be done outside the formal structure of rule-based languages. In recent years, there have been several studies on efficient methods for checking for contradictions in rule-based languages (see, e.g., Adams [24], Suwa et al. [25], Nguyen et al. [26], Pearl [27], Touretzky [28], and Ginsberg [29]). As we shall see, unlike rule-based languages, maintenance of consistency is an integral part of valuation-based languages

Caching

Regarding caching of inferences, typically, backward-chaining, goal-driven rule-based languages do not cache any inferences, whereas forward-chaining production systems cache all inferences in working memory. In either case, caching in rule-based languages is of little help to the knowledge engineer in understanding the implications of the knowledge in the knowledge base. As we shall see, valuation languages cache and display certain inferences, and this can be very useful in the knowledge engineering process.

Nonmonotonic Reasoning

The subject of nonmonotonic or default reasoning is an important area in artificial intelligence. We often use assumptions or defaults as facts until we observe something that contradicts the inferences we have derived. We then need to retract some assumptions or defaults to avoid the contradiction. A famous example is that of Tweety the bird. Most birds fly. We may initially use the rule *If X is a bird then X flies* as an assumption or a default. Upon learning that Tweety is a bird, we may infer that Tweety flies. However, we may subsequently learn that Tweety is a penguin and does not fly. At this stage, to keep our knowledge base contradiction-free, we need to retract the assumption that led to the contradiction.

The construction of efficient procedures to enable nonmonotonic or default

reasoning is the subject of considerable research in artificial intelligence (McCarthy [30], McCarthy and Hayes [31], McDermott and Doyle [32], Moore [33], Reiter [34]).

Uncertainty

Finally, it is now well known that pure rule-based languages are inadequate both to represent uncertain knowledge and to make inferences from such knowledge (Shafer [35], Heckerman and Horvitz [36]). For example, MYCIN used certainty factors and PROSPECTOR used a pair of likelihood ratios with each rule to represent uncertainty (Shortliffe and Buchanan [37], Duda et al [38]). However, these systems are brittle. They give the right answers in only the simplest of cases.

One solution to some of these problems is to couple a truth maintenance system to the knowledge base (Doyle [39], de Kleer [40], Reiter and de Kleer [41]). Truth maintenance systems were devised by logicians in artificial intelligence to reason with incomplete and uncertain information symbolically without using numerical calculi such as probability theory or belief functions. Truth maintenance systems are still in the developmental stage and are the subject of intense research in artificial intelligence.

Another solution has been to control the sequence of inferences so that the correct results are obtained. This approach has been studied, for example, by Laskey and Lehner [42] and by D'Ambrosio [43].

AN ABSTRACT DESCRIPTION OF A VALUATION-BASED LANGUAGE

This section gives an abstract description of a valuation-based language. The language consists of objects, and operators that operate on the objects. The objects are used to represent knowledge. The operators are used to make inferences from the knowledge. In rule-based languages, the objects are variables and rules and the operator is *modus ponens*. In valuation-based languages, the objects are called variables and valuations, and the operators are called combination, marginalization, and solution.

The level of abstractness at which this language is described here forces us to omit the computational details of how precisely the three operators are used to make inferences. This allows us to concentrate on the concepts. (For a more formal and less abstract exposition with theorems and proofs, we refer the reader to Shenoy and Shafer [1-4] and Shafer and Shenoy [13].) However, since abstract descriptions can be difficult to comprehend, we describe two specific valuation-based systems in the succeeding sections with concrete examples.

Variables and Configurations

We use the symbol $\mathcal{W}_X$ for the set of possible values of a variable X , and we call $\mathcal{W}_X$ the *frame for* X . We will be concerned with a finite set $\mathcal{X}$ of variables, and we will assume that all the variables in $\mathcal{X}$ have finite frames

Given a finite nonempty set h of variables, we let $\mathcal{W}_h$ denote the Cartesian product of $\mathcal{W}_X$ for X in h ; $\mathcal{W}_h = \times \{\mathcal{W}_X | X \in h\}$. We call $\mathcal{W}_h$ the *frame for* h . We will refer to elements of $\mathcal{W}_h$ as *configurations of* h .

PROJECTION OF CONFIGURATIONS *Projection* of configurations simply means dropping extra coordinates; if (w, x, y, z) is a configuration of $\{W, X, Y, Z\}$, for example, then the projection of (w, x, y, z) to $\{W, X\}$ is simply (w, x) , which is a configuration of $\{W, X\}$

If g and h are sets of variables, $h \subseteq g$, and x is a configuration of g , then we will let $x^{\downarrow h}$ denote the projection of x to h . The projection $x^{\downarrow h}$ is always a configuration of h . If $h = g$ and x is a configuration of g , then $x^{\downarrow h} = x$

Valuations

Given a set h of variables, there is a set $\mathcal{V}_h$. The elements of $\mathcal{V}_h$ are called *valuations of* h . We will let $\mathcal{V}$ denote the set of all valuations, that is, $\mathcal{V} = \bigcup \{\mathcal{V}_h | h \subseteq \mathcal{X}\}$. Valuations are primitives in our abstract description and as such require no definition. But as we shall see shortly, they are objects that can be combined, marginalized, and solved.

Intuitively, a valuation on h represents some knowledge about the variables in h

Examples of valuations on h are an *array*, a function $H: \mathcal{W}_h \rightarrow \mathbb{R}_+$ ($\mathbb{R}_+$ denotes the set of non-negative real numbers); a *superarray*, a function $H: 2^{\mathcal{W}_h} \rightarrow \mathbb{R}_+$ ($2^{\mathcal{W}_h}$ denotes the set of all subsets of $\mathcal{W}_h$); a *rule*, a function $H: \mathcal{W}_h \rightarrow \{\text{true}, \text{false}\}$, etc

PROPER VALUATIONS For each $h \subseteq \mathcal{X}$, there is a subset $\mathcal{P}_h$ of $\mathcal{V}_h$ whose elements will be called *proper valuations on* h . Let $\mathcal{P}$ denote $\bigcup \{\mathcal{P}_h | h \subseteq \mathcal{X}\}$, the set of all proper valuations.

Intuitively, a proper valuation represents knowledge that is consistent in itself. The notion of proper valuations is important as it enables us to define combinability of valuations, it allows us to define existence of solutions, and it allows us to constrain the definitions of combination and marginalization to meaningful operations.

Examples of proper valuations are a *potential*, a function $P: \mathcal{W}_h \rightarrow \mathbb{R}_+$ that is not identically zero for all configurations; a *superpotential*, a function $m: 2^{\mathcal{W}_h} \rightarrow \mathbb{R}_+$ that is not zero for all nonempty subsets of $\mathcal{W}_h$; a *satisfiable rule*, a function $R: \mathcal{W}_h \rightarrow \{\text{true}, \text{false}\}$ that is not identically false for all

configurations; etc. Potentials correspond to unnormalized probability distributions, and superpotentials correspond to unnormalized basic probability assignment functions.

Combination

We assume there is a mapping $\otimes: \mathcal{V} \times \mathcal{V} \rightarrow \mathcal{V}$, called *combination*, such that

1. If G and H are valuations on g , and h , respectively, then $G \otimes H$ is a valuation on $g \cup h$.
2. If either G or H is not a proper valuation, then $G \otimes H$ is not a proper valuation
3. If G and H are both proper valuations, then $G \otimes H$ may or may not be a proper valuation

If $G \otimes H$ is not a proper valuation, then we shall say that G and H are *not combinable*. If $G \otimes H$ is a proper valuation, then we shall say that G and H are *combinable* and that $G \otimes H$ is the *combination* of G and H .

Intuitively, as its name suggests, combination corresponds to aggregation of knowledge. If G and H are proper valuations on g and h representing knowledge about variables in g and h , respectively, then $G \otimes H$ represents the aggregated knowledge about variables in $g \cup h$.

For potentials, combination corresponds to pointwise multiplication, if G and H are potentials on g and h , respectively, then $(G \otimes H)(x) = G(x^{ig})H(x^{ih})$. For basic probability assignment functions, combination corresponds to Dempster's rule (Dempster [44, 45]). For rules, if G and H are rules on g and h , respectively, then $G \otimes H$ is a rule on $g \cup h$ such that $(G \otimes H)(x) = \text{true}$ iff $G(x^{ig}) = \text{true}$ and $H(x^{ih}) = \text{true}$.

Marginalization

We assume that for each $h \subseteq \mathcal{X}$, there is a mapping $\downarrow h: \bigcup \{\mathcal{V}_g | g \supseteq h\} \rightarrow \mathcal{V}_h$, called *marginalization to h* , such that

1. If G is a valuation on g and $h \subseteq g$, then G^{ih} is a valuation on h
2. If G is a proper valuation, then G^{ih} is a proper valuation.
3. If G is not a proper valuation, then G^{ih} is not a proper valuation

We will call G^{ih} the *marginal* of G for h .

Intuitively, marginalization corresponds to crystallization of knowledge. If G is a valuation on g representing some knowledge about variables in g , and $h \subseteq g$, then G^{ih} represents the knowledge about variables in h implied by G if we disregard variables in $g - h$.

In the case of potentials, marginalization from g to h is summation over the configurations of $g - h$. In the belief-function case, marginalization is explained in the section "An Evidential Language..." For rules, if G is a rule on g , then

G^{ih} is a rule on h such that $G^{ih}(x) = \text{true}$ iff there is a configuration y of $g - h$ such that $G(x, y) = \text{true}$.

Solution

We assume that for each $g \subseteq \mathcal{X}$, there is a mapping $\psi: \mathcal{V}_g \rightarrow 2^{\mathcal{W}_g}$ called *solution* such that

1. If G is a proper valuation on g , then $\psi(G)$ is a nonempty subset of $\mathcal{W}_g$.
2. If G is a valuation on g that is not proper, then $\psi(G) = \emptyset$.

The configurations in $\psi(G)$ are called *solutions for G* .

Intuitively, the solution operator maps knowledge from the space of valuations to the space of configurations. We encode knowledge as valuations so that we can aggregate and crystallize it. However, we need to decode the result. The solution operator simply serves as a decoding mechanism.

In the case of probabilities, solutions may correspond to configurations with the highest probability or simply configurations with positive probabilities. For belief functions, solutions may correspond to configurations with the highest plausibility or simply configurations with positive plausibilities. For rules, solutions may correspond to configurations whose value is true.

Propagation of Valuations.

A valuation-based language (VL) makes inferences by

1. Combining all proper valuations in the system (the resulting valuation, if proper, is called the *joint valuation*),
2. Computing the marginal of the joint valuation for each variable in the system,
3. Computing the set of all solutions for the marginals of the joint valuation for each variable; and
4. Computing the set of all solutions for the joint valuation

The above is only a conceptual description of the actions of a valuation-based language. It is not an algorithm. If there are n variables in the system, and each variable has two configurations in its frame, then there are 2^n configurations of all variables. Hence, it will not be feasible to compute the joint valuation when there are a large number of variables. The VL does not actually compute the joint valuation. It computes the marginals of the joint valuation without explicitly computing the joint valuation, and it does this using only local computations. An algorithm for computing exact marginals and solutions is described in detail in Shenoy and Shafer [1–4]. An algorithm for computing approximate marginals is described in Pearl [46], and an algorithm for computing an approximate solution to the joint valuation is described in Kirkpatrick et al. [47] and Geman and Geman [48].

Valuation System

A *valuation system* (VS) consists of a finite set of variables $\mathfrak{X}$, a finite frame $\mathfrak{W}_X$ for each variable X in $\mathfrak{X}$, and a finite collection of valuations $\{V_i\}_{i \in M}$ where each valuation V_i is on some subset of $\mathfrak{X}$.

A *valuation network* is a graph whose vertices represent either variables or valuations. If valuation V_i is on a subset h of vertices, then this is represented in the valuation network by including an edge between the vertex corresponding to V_i and all variable vertices X_j such that $X_j \in h$.

The valuation network serves as a graphical representation of a valuation system and can be used as a user interface. The valuation network is also used by the VL to propagate the valuations. The algorithm for computing exact marginals and solution requires that the valuation network be a tree. If the valuation network is not a tree, then this algorithm embeds it in a tree by clustering variables. Such a tree, called a Markov tree, is then used to compute the marginals and solutions (Shenoy and Shafer [1–4]). The simulation algorithms for computing marginals and solutions use the valuation network directly.

Capabilities of a Valuation-Based Language

A VL has the following capabilities:

1. A VS can be extended by adding new variables and adding new proper valuations.
2. A VS can also be reduced by removing variables and valuations.
3. Each time the VL receives a new proper valuation, it checks whether or not it is combinable with the proper valuations already present in the system.
4. If the new proper valuation is combinable with the valuations already present in the system, then the VL accepts the new valuation. If the new proper valuation is not combinable, then the VL rejects it and informs the user of its action.
5. Each time the VL accepts a proper valuation, it finds the marginal of the joint valuation (the valuation obtained by combining all proper valuations in the system) for each variable in the system. This is accomplished using local computations if an efficient Markov tree can be found for the valuation network (Shenoy and Shafer [1, 2]) or by stochastic simulation otherwise (Pearl [46]).
6. The VL also computes for each variable the set of all solutions for the marginal of the joint valuation for that variable. Once we have the marginal of the joint valuation for a variable, computing the set of all solutions is simply done by exhaustive enumeration of the frame for that variable.
7. If necessary, the VL can compute a configuration of all variables that is a

solution for the joint valuation. This can be done using an exact algorithm if an efficient Markov tree can be found for the valuation network (Shenoy and Shafer [3, 4]) or by stochastic relaxation and annealing (Kirkpatrick et al [47], Geman and Geman [48])

A VALUATION LANGUAGE FOR CATEGORICAL KNOWLEDGE

In this section, we describe an instance of a valuation-based language designed to represent categorical knowledge—the kind of knowledge traditionally represented by rules in rule-based systems. Next, we show by means of a small example how consistency is maintained in the knowledge base and how inferences are cached. Our exposition here is informal. A formal treatment (with theorems and proofs) of the valuation language described in this section is given in Shenoy and Shafer [3].

Suppose we are interested in representing categorical knowledge in a valuation system. Let us describe what valuations are and what the combination marginalization, and solution operations are for such systems

VALUATIONS A *valuation on h* is a function $H: \mathcal{W}_h \rightarrow \{t, f\}$, where t means true and f means false

Thus a rule that relates the values of variables in set h is represented as a valuation on h . For example, consider the rule *If $A = a$ then $B = b$* that relates two variables A and B whose frames are, respectively, $\mathcal{W}_A = \{a, \sim a\}$, and $\mathcal{W}_B = \{b, \sim b\}$. Then the rule can be represented by the valuation V on $\{A, B\}$ defined as follows: $V(a, b) = t$, $V(a, \sim b) = f$, $V(\sim a, b) = t$, $V(\sim a, \sim b) = t$.

Consider the valuation U on h such that $U(x) = t$ for all $x \in \mathcal{W}_h$. Obviously, such a valuation tells us nothing about the variables in h . We call such a valuation the *vacuous valuation on h* .

PROPER VALUATIONS Suppose H is a valuation on h . We shall say that H is a *proper valuation* if there exists a configuration x of h such that $H(x) = t$. Thus a proper valuation cannot be identically equal to f for all configurations.

COMBINATION Suppose G and H are valuations on g and h , respectively. The valuation $G \otimes H$ on $g \cup h$ is defined as follows:

$$(G \otimes H)(x) = \begin{cases} t & \text{if } G(x^{\downarrow g}) = t \text{ and } H(x^{\downarrow h}) = t \\ f & \text{otherwise} \end{cases}$$

for all $x \in \mathcal{W}_{g \cup h}$.

MARGINALIZATION Suppose G is a valuation on g , and suppose $h \subseteq g$. Then the *marginal of G for h* , G^{ih} , is defined as follows:

$$G^{ih}(x) = \begin{cases} t & \text{if there is a } y \in \mathcal{W}_{g-h} \text{ such that } G(x, y) = t \\ f & \text{otherwise} \end{cases}$$

for all $x \in \mathcal{W}_h$.

SOLUTION Suppose G is a valuation on g . The *solution for G* , denoted by $\psi(G)$, is a subset of $\mathcal{W}_g$ such that $y \in \psi(G)$ if and only if $G(y) = t$.

The combination, marginalization, and solution operations are used by the VL to make inferences from the knowledge

Suppose a knowledge base is built incrementally by adding valuations one at a time. Consistency in the knowledge base is maintained by the VL by checking whether the added valuation is proper and combinable with the proper valuations already present in the system. Thus combinability of valuations corresponds to consistency in the knowledge base (Shenoy and Shafer [3]).

As valuations are added to the knowledge base, the system propagates all valuations and computes the marginal of the joint valuation for each variable and the solutions for each of these marginals. More precisely, suppose $\{R_h | h \in \mathcal{H}\}$ is a collection of proper combinable valuations in the system. The valuation $\otimes \{R_h | h \in \mathcal{H}\}$ is called the *joint valuation*. The valuation system computes $(\otimes \{R_h | h \in \mathcal{H}\})^{i\{X_i\}}$ for each variable X_i and also computes $\psi((\otimes \{R_h | h \in \mathcal{H}\})^{i\{X_i\}})$. In doing so, the VS acts as a *cache*. At all times, the VS indicates the relevant inferences of the knowledge in the knowledge base.

AN EXAMPLE The following example is adapted from Etherington [49]. The knowledge base consists of four rules as follows

Rule 1. Gullible citizens are citizens.

Rule 2. Elected crooks are crooks.

Rule 3. Citizens dislike crooks.

Rule 4. Gullible citizens do not dislike elected crooks

First we observe that Fred is a gullible citizen. Next we observe that Dick is an elected crook. We would like to consult our knowledge base to see if Fred dislikes Dick or not.

One representation of this knowledge base is as follows. Let $C = c$, $G = g$, $K = k$, $E = e$, and $D = d$ be five variables and their respective configurations representing X is a citizen, X is a gullible citizen, Y is a crook, Y is an elected crook, and X dislikes Y , respectively. Suppose all five of these variables are binary variables.

Table 1. The Valuations Corresponding to the Four Rules

$\mathcal{W}_{\{C,G\}}$	R_1	$\mathcal{W}_{\{K,E\}}$	R_2
$c \quad g$	t	$k \quad e$	t
$c \quad \sim g$	t	$k \quad \sim e$	t
$\sim c \quad g$	f	$\sim k \quad e$	f
$\sim c \quad \sim g$	t	$\sim k \quad \sim e$	t
$\mathcal{W}_{\{C,K,D\}}$	R_3	$\mathcal{W}_{\{G,E,D\}}$	R_4
$c \quad k \quad d$	t	$g \quad e \quad d$	f
$c \quad k \quad \sim d$	f	$g \quad e \quad \sim d$	t
$c \quad \sim k \quad d$	t	$g \quad \sim e \quad d$	t
$c \quad \sim k \quad \sim d$	t	$g \quad \sim e \quad \sim d$	t
$\sim c \quad k \quad d$	t	$\sim g \quad e \quad d$	t
$\sim c \quad k \quad \sim d$	t	$\sim g \quad e \quad \sim d$	t
$\sim c \quad \sim k \quad d$	t	$\sim g \quad \sim e \quad d$	t
$\sim c \quad \sim k \quad \sim d$	t	$\sim g \quad \sim e \quad \sim d$	t

Rules 1, 2, 3, and 4 are represented by proper valuations on $\{C, G\}$, $\{K, E\}$, $\{C, K, D\}$, and $\{G, E, D\}$, respectively, as shown in Table 1

Suppose these variables and valuations are entered in the system. A network representation of the system is shown in Figure 1. In that figure, variables are represented by circles and valuations are represented by squares. For each variable, the set of all solutions for the marginal of the joint valuation for that variable is indicated inside the variable vertex. As can be seen from Figure 1, for each variable the marginal of the joint valuation for that variable is the vacuous valuation

Now, suppose we enter the observation that Fred is a gullible citizen. This is represented in the system as a proper valuation F_1 on $\{G\}$ as follows $F_1(g) = t$, $F_1(\sim g) = f$. The system accepts this proper valuation, and after propagation it displays the results as shown in Figure 2. Note that the system properly concludes that Fred is a citizen. However, the system also concludes that Y is not an elected crook! This is the first hint we have that something is wrong with our knowledge base. The system has concluded something about Y without being told it explicitly, and this is not an inference we expect from the knowledge base. The reason for the inference *Y is not an elected crook* is the contradictory nature of rules 3 and 4.

Finally, we enter the observation that Dick is an elected crook. This

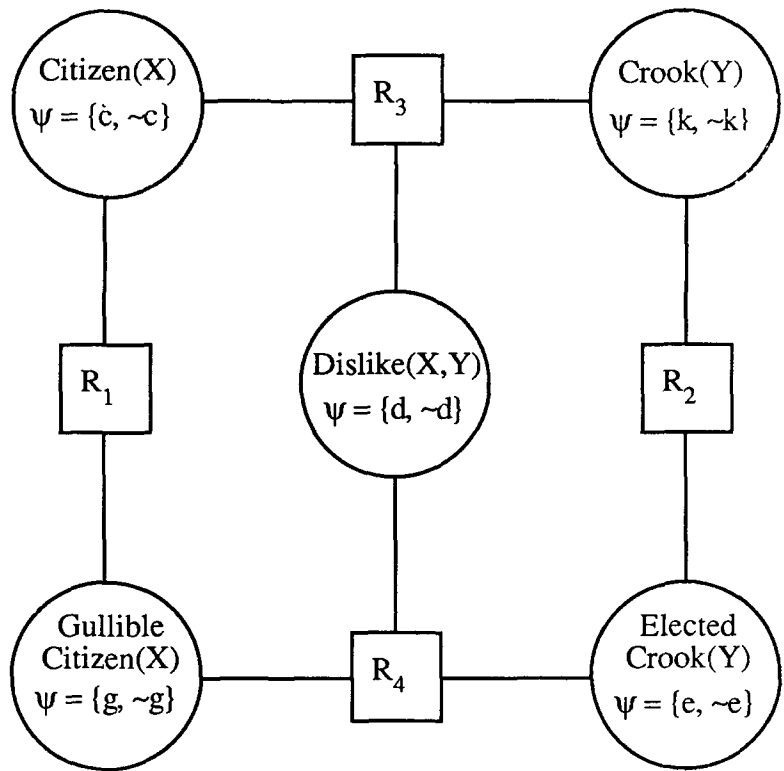


Figure 1. The valuation network with five variables and four rules

Table 2. The Valuation Corresponding to Rule 5

$\mathcal{W}_{\{G,K,D\}}$			R_5
g	k	d	t
g	k	$\sim d$	t
g	$\sim k$	d	t
g	$\sim k$	$\sim d$	t
$\sim g$	k	d	t
$\sim g$	k	$\sim d$	f
$\sim g$	$\sim k$	d	t
$\sim g$	$\sim k$	$\sim d$	t

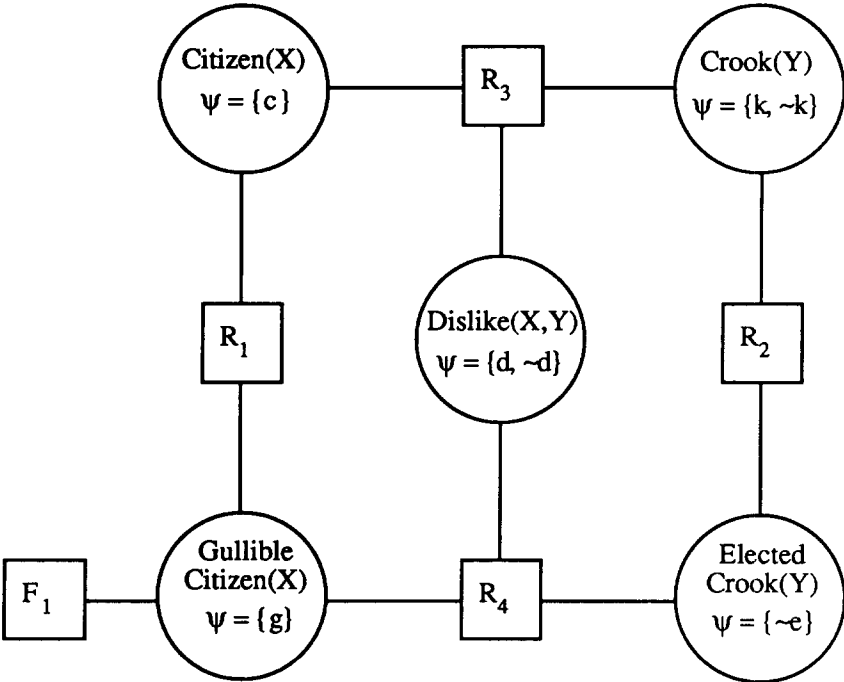


Figure 2. The valuation network after valuation F_1 is included

observation is represented as a proper valuation F_2 on $\{E\}$ as follows: $F_2(e) = t$, $F_2(\sim e) = f$. This time the system refuses to accept the valuation because the system detects that the joint valuation $R_1 \otimes R_2 \otimes R_3 \otimes R_4 \otimes F_1 \otimes F_2$ is not a proper valuation. This signals that the knowledge in the system is inconsistent.

Suppose we remove rule 3 from the system and substitute instead rule 5 as follows:

Rule 5 Nongullible citizens dislike crooks.

Rule 5 is represented in the system as the valuation R_5 on $\{G, K, D\}$ as shown in Table 2. The valuation system accepts valuation R_5 with the results shown in Figure 3. Note that the system now concludes nothing about Y .

Finally we enter valuation F_2 in the system. This time the system accepts the valuation with the results shown in Figure 4. Thus we conclude that Fred does not dislike Dick.

We have not described the exact process by which the valuation language arrives at the results displayed in Figures 1–4. A computationally efficient procedure in sparse networks that uses only local computation is described in Shenoy and Shafer [3].

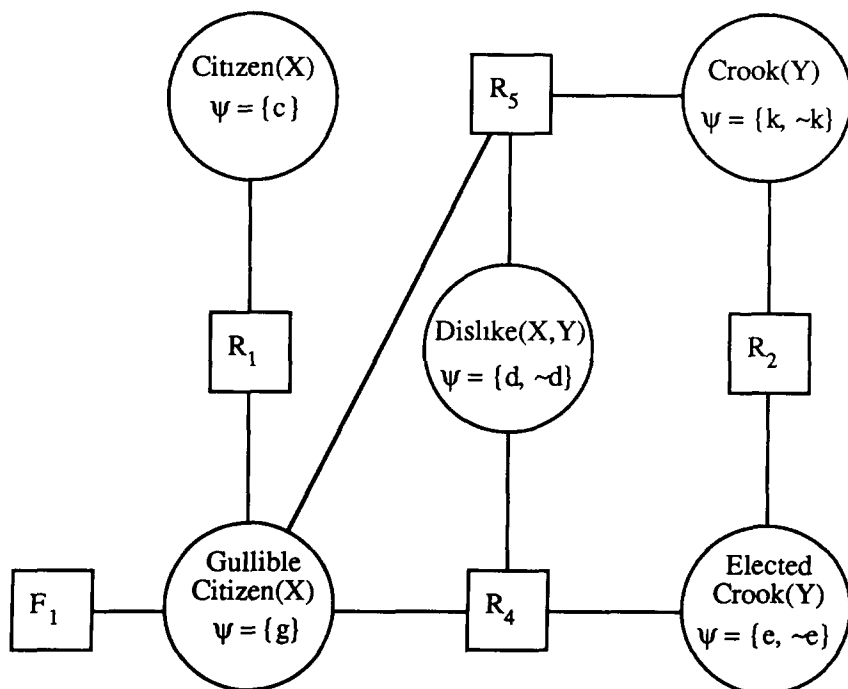


Figure 3. The valuation network after valuation R_3 is removed and valuation R_5 is added

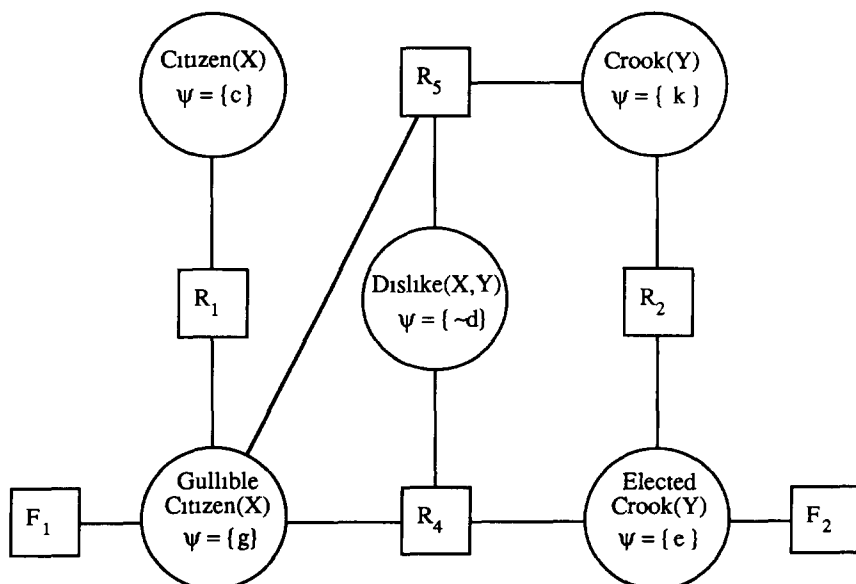


Figure 4. The valuation network after valuation F_2 is included

AN EVIDENTIAL LANGUAGE FOR UNCERTAIN AND NONMONOTONIC REASONING

In this section, we describe another valuation language called an evidential language. The valuations in this language are belief functions.

Propagation of belief functions has been studied by Shafer and Logan [50], Shenoy and Shafer [1, 2, 51], Shenoy et al. [52], Kong [53, 54], Dempster and Kong [55], Shafer et al. [56], Mellouli [57], Shafer and Shenoy [13], Dempster [22], and Almond [58]. Zarley [59] describes an implementation of an evidential system on a Symbolics workstation (see also Zarley et al. [60]). Yen-Teh Hsia has implemented an evidential system called AUDITOR'S ASSISTANT on a Macintosh microcomputer. Shafer et al. [7] describe an application of AUDITOR'S ASSISTANT for assisting in audit decisions.

The use of probabilities or belief functions to perform nonmonotonic reasoning is not new. Such an approach has been suggested, for example, by Baldwin [61], Ginsberg [62], and Rich [63]. The essence of these approaches is to relax the binary constraint of Boolean logic and allow truth values to be measured by a number between 0 and 1. Our approach is different. We do not tack on probabilities or belief functions to logic. Instead, we show that pure belief-function reasoning is inherently nonmonotonic. A similar approach is taken by Grosz [64], who discusses how probabilistic reasoning is nonmonotonic.

In this section, we will first briefly describe evidential systems. Next, we sketch the basic definitions in a truth maintenance system and describe the correspondence between concepts in an truth maintenance system and concepts in an evidential system. Finally, we study a small example in nonmonotonic reasoning and demonstrate how evidential systems handle such problems. This example also serves to illustrate the management of uncertainty in evidential systems.

An Evidential System

In evidential systems (ES), proper valuations correspond to superpotentials, which are unnormalized basic probability assignment functions. First we will briefly describe the basics of the theory of belief functions (Shafer [65]). Next, we define superpotentials and combination, marginalization, and solution for superpotentials.

Suppose $\mathcal{W}_h$ is the frame for a subset h of variables. A *basic probability assignment function (bpa function)* for h is a non-negative, real-valued function m on the set of all subsets of $\mathcal{W}_h$ such that

1. $m(\emptyset) = 0$
2. $\sum\{m(\alpha) \mid \alpha \subseteq \mathcal{W}_h\} = 1$

Intuitively, $m(\alpha)$ represents the degree of belief assigned exactly to α (the proposition that the true configuration of h is in the set α) and to nothing smaller.

A bpa function is the belief function equivalent of a probability mass assignment function in probability theory. Whereas a probability mass function is restricted to assigning probability masses only to singleton configurations of variables, a bpa function is allowed to assign probability masses to sets of configurations without assigning any mass to the individual configurations contained in the sets.

For example, if we have absolutely no knowledge about the true value of a variable, we can represent this situation by a bpa function as follows:

$$m(\mathcal{W}_h) = 1, \quad m(\alpha) = 0 \text{ for all other } \alpha \in 2^{\mathcal{W}_h}$$

Such a function is called a *vacuous bpa function*. Note that in Bayesian theory the only way to express total ignorance is to assign a mass of $1/n$ to each value, where n is the total number of possible values. Thus, in Bayesian theory, we are unable to distinguish between equally likely configurations and total ignorance. The theory of belief functions offers richer semantics.

Associated with a bpa function are two related functions called belief and plausibility. A *belief function* is a function $\text{Bel}: 2^{\mathcal{W}_h} \rightarrow [0, 1]$ such that

$$\text{Bel}(\alpha) = \sum \{m(\beta) \mid \beta \subseteq \alpha\}$$

Whereas $m(\alpha)$ represented the belief assigned exactly to α , $\text{Bel}(\alpha)$ represents the total belief assigned to α . Note that $\text{Bel}(\emptyset) = 0$ and $\text{Bel}(\mathcal{W}_h) = 1$ for any belief function. For the vacuous bpa function m , the corresponding belief function Bel is given by

$$\text{Bel}(\mathcal{W}_h) = 1, \quad \text{Bel}(\alpha) = 0 \text{ for all other } \alpha \in 2^{\mathcal{W}_h}$$

A *plausibility function* is a function $\text{Pl}: 2^{\mathcal{W}_h} \rightarrow [0, 1]$ such that

$$\text{Pl}(\alpha) = \sum \{m(\beta) \mid \beta \cap \alpha \neq \emptyset\}$$

$\text{Pl}(\alpha)$ represents the total degree of belief that could be assigned to α . Note that $\text{Pl}(\alpha) = 1 - \text{Bel}(\sim \alpha)$, where $\sim \alpha$ represents the complement of α in $\mathcal{W}_h$; $\sim \alpha = \mathcal{W}_h - \alpha$. Also note that $\text{Pl}(\alpha) \geq \text{Bel}(\alpha)$. For the vacuous bpa function, the corresponding plausibility function is

$$\text{Pl}(\emptyset) = 0, \quad \text{Pl}(\alpha) = 1 \text{ for all other } \alpha \in 2^{\mathcal{W}_h}.$$

If a bpa function m is also a probability mass function (i.e., all the probability masses are assigned only to singleton subsets), then

$$\text{Bel}(\alpha) = \text{Pl}(\alpha) = \sum \{m(\{x\}) \mid x \in \alpha\} = \text{probability of proposition } \alpha$$

SUPERPOTENTIALS Suppose h is a subset of variables. A *superpotential for h* is a non-negative, real-valued function on the set of all subsets of $\mathcal{W}_h$ such that the values of nonempty subsets are not all zero. Given a superpotential H on h ,

we can construct a bpa function H' for h from H as follows:

$$H'(\emptyset) = 0, \quad H'(\alpha) = H(\alpha) / \Sigma \{ H(\beta) \mid \beta \subseteq \mathcal{W}_h, \beta \neq \emptyset \}$$

Thus superpotentials can be thought of as unnormalized bpa functions. Superpotentials correspond to the notion of proper valuations in the general framework.

PROJECTION AND EXTENSION OF SUBSETS Before we can define combination and marginalization for superpotentials, we need the concepts of projection and extension for subsets of configurations.

If g and h are sets of variables, $h \subseteq g$, and q is a nonempty subset of $\mathcal{W}_g$, then the *projection of q to h* , denoted by q^{th} , is the subset of $\mathcal{W}_h$ given by $q^{th} = \{x^{th} \mid x \in q\}$.

For example, if α is subset of $\mathcal{W}_{\{W, X, Y, Z\}}$, then the marginal of α to $\{X, Y\}$ consists of the elements of $\mathcal{W}_{\{X, Y\}}$ that can be obtained by projecting elements of α to $\mathcal{W}_{\{X, Y\}}$.

By extension of a subset of a frame to a subset of a larger frame, we mean a cylinder set extension. If g and h are sets of variables, $h \subseteq g$, $h \neq g$, and k is a subset of $\mathcal{W}_h$, then the *extension of k to g* is $k \times \mathcal{W}_{g-h}$. If k is a subset of $\mathcal{W}_h$, then the extension of k to h is defined to be k . We will let k^{tg} denote the extension of k to g ,

For example, if α is a subset of $\mathcal{W}_{\{W, X\}}$, then the vacuous extension of α to $\{W, X, Y, Z\}$ is $\alpha \times \mathcal{W}_{\{Y, Z\}}$.

COMBINATION For superpotentials, combination is called Dempster's rule (Dempster [44, 45]) Consider two superpotentials G and H on g and h , respectively. If

$$\Sigma \{ G(\alpha) H(\beta) \mid (\alpha^{t(g \cup h)} \cap \beta^{t(g \cup h)}) \neq \emptyset \} \neq 0 \quad (1)$$

then their *combination*, denoted by $G \oplus H$, is the superpotential on $g \cup h$ given by

$$(G \oplus H)(c) = \Sigma \{ G(\alpha) H(\beta) \mid (\alpha^{t(g \cup h)} \cap \beta^{t(g \cup h)}) = c \} \quad (2)$$

for all $c \subseteq \mathcal{W}_{g \cup h}$. If $\Sigma \{ G(\alpha) H(\beta) \mid (\alpha^{t(g \cup h)} \cap \beta^{t(g \cup h)}) \neq \emptyset \} = 0$, then we say that G and H are *not combinable*.

Intuitively, if the bodies of evidence on which G and H are based are independent, then $G \oplus H$ is supposed to represent the result of pooling these two bodies of evidence. Note that condition (1) ensures that $G \oplus H$ defined in (2) is a superpotential. If condition (1) does not hold, this means that the two bodies of evidence corresponding to G and H contradict each other completely and it is not possible to combine such evidence

MARGINALIZATION Suppose G is a superpotential for g , and suppose $h \subseteq g$.

Then the *marginal of G for h* is the superpotential G^{ih} for h defined as follows:

$$G^{ih}(\alpha) = \Sigma \{ G(\beta) \mid \beta \subseteq \mathcal{W}_g \text{ such that } \beta^{ih} = \alpha \}$$

for all subsets α of $\mathcal{W}_h$.

SOLUTION There are several definitions of solution possible for evidential systems. For nonmonotonic reasoning, we will define a solution for m to be a configuration whose plausibility is positive. Formally, suppose m is a bpa on h . Suppose Pl is the plausibility function on h corresponding to m . Then we say that $x \in \mathcal{W}_h$ is a *solution for m* if $Pl(\{x\}) > 0$.

A Truth Maintenance System

Assume a propositional language consisting of propositional symbols, the logical connectives $\wedge$, $\vee$, $\sim$, $\rightarrow$, $\leftrightarrow$, formulas, and the usual standard entailment relation $\Rightarrow$. If S is a set of formulas and w is a formula, then $S \Rightarrow w$ if every assignment of truth values to the propositional symbols of the language that makes each formula of S true also makes w true.

A *literal* is a propositional symbol or the negation of a propositional symbol. A *clause* is a finite disjunction of literals with no literals repeated whose truth value is true. A *premise* is a literal whose truth value is true. A *categorical justification* is a conditional whose truth value is true. Note that a categorical justification can be represented as a clause. For example, the conditional $A = a \rightarrow B = b$ can be represented as a clause as follows: $\sim(A = a) \vee (B = b)$.

An *assumption* is a literal whose truth value is assumed to be true in the absence of a contradiction. A *noncategorical justification* is a conditional whose truth value is assumed to be true in the absence of a contradiction. A *nogood* is a clause whose truth value is false.

A *knowledge base* is a collection of justifications (rules), premises (observations), and assumptions (uncertain judgments). Justifications may be categorical or noncategorical. Categorical justifications may describe logical relations between propositional symbols. Non-categorical justifications may describe facts that are usually but not always true.

The functions of a truth maintenance system (TMS) are as follows:

1. The use of noncategorical justifications and assumptions or defaults is permitted.
2. In the absence of a contradiction, noncategorical justifications and assumptions are assumed to be true.
3. If there is a contradiction in the knowledge base, then some noncategorical justifications or assumptions or both need to be retracted so that consistency is restored. When an assumption or a noncategorical justification is retracted, all inferences made using these assumptions and noncategorical justifications must also be retracted.

4. All inferences that are consistent with the knowledge in the knowledge base should be displayed to the user so that the user is aware of the implications of the knowledge.

USING AN EVIDENTIAL LANGUAGE AS A TMS We will now outline a correspondence between the concepts in a TMS and concepts in an evidential system (ES).

A literal in a TMS is represented in an ES by a variable and one of its values. Thus $X = x$ is an ES representation of the literal x where x belongs to $\mathcal{W}_X$, the set of possible values of variable X . For example, suppose the proposition TWEETY IS A BIRD is represented in a TMS as a literal. In the ES, this could be represented by a variable BIRD with two possible values *yes* and *no*. Then the literal TWEETY IS A BIRD corresponds to $\text{BIRD} = \text{yes}$ in an ES.

A premise is a literal whose truth value is true. In an ES, a premise is represented as a categorical belief function. For example, the premise $X = x$ is represented by a belief function on $\mathcal{W}_X$ given by $m(\{x\}) = 1$.

An assumption in a TMS is a literal whose truth value is set to true in the absence of a contradiction in the knowledge base. In the ES, an assumption $X = x$ is represented by a noncategorical belief function Bel (with basic probability assignment m) on $\mathcal{W}_X$ such that

$$m(\{x\}) = p \quad \text{and} \quad m(\mathcal{W}_X) = 1 - p$$

where $0 < p < 1$. The actual value of p will depend on the particular assumption. p can be interpreted to be the prior degree of belief in the assumption.

A justification is a conditional,

$$x_1 \wedge x_2 \wedge \cdots \wedge x_n \rightarrow y$$

where $x_1, x_2, \dots, x_n, y$ are literals. In an ES, a categorical justification $x_1 \wedge x_2 \wedge \cdots \wedge x_n \rightarrow y$ is represented as a categorical belief function on the frame $\mathcal{W}_h$, where $h = \{X_1, X_2, \dots, X_n, Y\}$. For example, consider two variables X and Y with frames $\mathcal{W}_X = \{x, \sim x\}$ and $\mathcal{W}_Y = \{y, \sim y\}$. Then the categorical justification $x \rightarrow y$ is represented in the ES as a categorical belief function on $\mathcal{W}_{\{X,Y\}}$ given by

$$m(\{(x, y), (\sim x, y), (\sim x, \sim y)\}) = 1$$

Noncategorical justifications are represented in the ES as noncategorical belief functions. There are several ways in which this can be done. The most appropriate way will depend on the nature of the particular justification.

The first type of belief-function representation of a noncategorical justification is called *exceptional*. The exceptional representation of a noncategorical justification is implied by McCarthy's [66] formulation. For example, consider the noncategorical justification MOST BIRDS FLY. This can be represented in a

TMS by a categorical justification and an assumption as follows:

$$\begin{aligned} \text{BIRD} = \text{yes} \wedge \text{EXCEPTIONAL_BIRD} = \text{no} &\rightarrow \text{FLY} = \text{yes} \\ \text{Assume } \text{EXCEPTIONAL_BIRD} &= \text{no} \end{aligned}$$

Here $\text{EXCEPTIONAL_BIRD} = \text{no}$ is a literal that captures all the conditions under which birds fly. Let $B = b$, $E = \sim e$, and $F = f$ denote the ES representation of the literals $\text{BIRD} = \text{yes}$, $\text{EXCEPTIONAL_BIRD} = \text{no}$, and $\text{FLY} = \text{yes}$. Then the justification MOST BIRDS FLY can be represented in an ES by two independent basic probability assignment functions, m_1 on $\mathcal{W}_{\{B,E,F\}}$ and m_2 on $\mathcal{W}_E$ as follows:

$$m_1(\mathcal{W}_{\{B,E,F\}} - \{(b, \sim e, \sim f)\}) = 1$$

$$m_2(\{\sim e\}) = p, \quad m_2(\mathcal{W}_E) = 1 - p$$

where $0 < p < 1$. Note that $m_1 \oplus m_2$ is a basic probability assignment on $\mathcal{W}_{\{B,E,F\}}$ given by

$$m_1 \oplus m_2(\{(b, \sim e, f), (\sim b, \sim e, f), (\sim b, \sim e, \sim f)\}) = p$$

$$m_1 \oplus m_2(\mathcal{W}_{\{B,E,F\}} - \{(b, \sim e, \sim f)\}) = 1 - p$$

$m_1 \oplus m_2$ is then the exceptional representation in an ES of the justification MOST BIRDS FLY

The second type of belief-function representation of a noncategorical justification is called *associational*. Consider again the justification MOST BIRDS FLY. We can interpret this to mean that birds are associated with flying with a certain degree of belief. This association may just go one way, that is, we may not necessarily associate all flying objects with birds. Interpreted in this way, we can represent this justification by a basic probability assignment function m_3 on $\mathcal{W}_{\{B,F\}}$ as follows

$$m_3(\{(b, f), (\sim b, f), (\sim b, \sim f)\}) = p, \quad m_3(\mathcal{W}_{\{B,F\}}) = 1 - p$$

where $0 < p < 1$.

Obviously, exceptional representations of noncategorical justifications have greater expressive power than associational representations. In the bird example, if the basic probability assignment function $m_1 \oplus m_2$ is marginalized by deleting the E variable, then we obtain precisely the associational representation m_3 , that is, $(m_1 \oplus m_2)^{\downarrow\{B,F\}} = m_3$. However, this expressive power comes at a computational cost since more variables are required in the exceptional representation than in the associational representation.

Consider a knowledge base represented by a collection of bpa functions $\{m_i | i = 1, \dots, n\}$ representing premises, rules, and assumptions. Suppose m_j is an assumption $X = x$. We shall say that the assumption m_j is *retracted by the knowledge base* $\{m_i | i = 1, \dots, n\}$ if $\text{Pl}^{\downarrow\{X\}}(\{x\}) = 0$ where $\text{Pl}^{\downarrow\{X\}}$ is the plausibility function corresponding to $(\oplus \{m_i | i = 1, \dots, n\})^{\downarrow\{X\}}$. We shall say

that the assumption m_i is confirmed by the knowledge base $\{m_i | i = 1, \dots, n\}$ if $m^{+ \{X\}}(\{x\}) = 1$ where $m = (\oplus \{m_i | i = 1, \dots, n\})$.

An Example

Consider the following knowledge base:

Rule 1. Most Republicans (at least 80%) are not pacifists.

Rule 2. Most Quakers (at least 90%) are pacifists

First we observe that Nixon is a Republican. Then we observe that Nixon is also a Quaker. We would like to consult our knowledge base to find out whether Nixon is a pacifist or not. Next we will add the premise that Nixon is not a pacifist and see how the evidential system reconciles this premise with rule 2.

One representation of this knowledge base is as follows. Let $R = r$, $Q = q$, $P = p$ be three variables and their respective configurations representing the propositions X is a Republican, X is a Quaker, and X is a pacifist, respectively. Furthermore, let $ER = er$ and $EQ = eq$ be two more variables and their respective configurations representing the propositions X is an exceptional Republican and X is an exceptional Quaker, respectively

We will represent rule 1 with categorical rule 1 and assumption 1 as follows.

CATEGORICAL RULE 1 If X is a Republican and X is not an exceptional Republican, then X is not a pacifist

ASSUMPTION 1. X is not an exceptional Republican

The bpa function representation of categorical rule 1 is as follows

$$m_1(\mathcal{W}_{\{R, ER, P\}} - \{(r, \sim er, p)\}) = 1$$

The bpa function representation of assumption 1 is as follows:

$$m_2(\{\sim er\}) = 0.8, \quad m_2(\mathcal{W}_{ER}) = 0.2$$

We will represent rule 2 with categorical rule 2 and assumption 2 as follows:

CATEGORICAL RULE 2. If X is a Quaker and X is not an exceptional Quaker, then X is a pacifist

ASSUMPTION 2 X is not an exceptional Quaker.

The bpa function representation of categorical rule 2 is as follows.

$$m_3(\mathcal{W}_{\{Q, EQ, P\}} - \{(q, \sim eq, \sim p)\}) = 1$$

The bpa function representation of assumption 2 is as follows:

$$m_4(\{\sim eq\}) = 0.9, \quad m_4(\mathcal{W}_{EQ}) = 0.1$$

If we enter these four bpa functions in the evidential system, the resulting

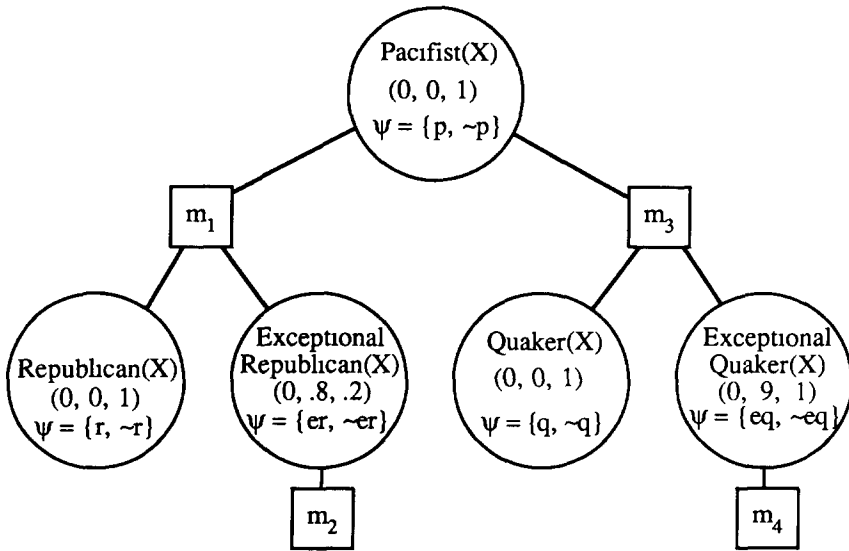


Figure 5. The evidential network with two categorical rules and two assumptions

evidential network is as shown in Figure 5. As before, variable vertices are shown as circles and valuation vertices are shown as squares. In addition to displaying the set of all solutions for the marginal of the joint valuation, the marginal bpa function is also displayed. If $\{x, \sim x\}$ is the frame for variable X , then the marginal of the joint valuation for X is shown as a vector $(m^{\{X\}}(\{x\}), m^{\{X\}}(\{\sim x\}), m^{\{X\}}(\{x, \sim x\}))$, where $m = \bigoplus \{m_i | i = 1, \dots, n\}$.

Suppose we now enter the premise that Nixon is a Republican. This is represented as a bpa function as follows

$$m_5(\{r\}) = 1$$

The evidential system accepts this bpa function with the results as shown in Figure 6. Note that the belief in the proposition that Nixon is not a pacifist has increased from 0 to 0.8 and the belief in the proposition Nixon is not a Quaker has increased from 0 to 0.72.

Suppose we now enter the premise that Nixon is a Quaker. This is represented by a bpa function as follows

$$m_6(\{q\}) = 1$$

The evidential system accepts this bpa function with the results shown in Figure 7. Note that as per the ES, Nixon could either be a pacifist or not. The plausibility of Nixon being a pacifist (0.71) is higher than the plausibility that Nixon is not a pacifist (0.35). This is because Quakers have higher belief (0.90) of being pacifists than Republicans have of not being pacifists (0.80).

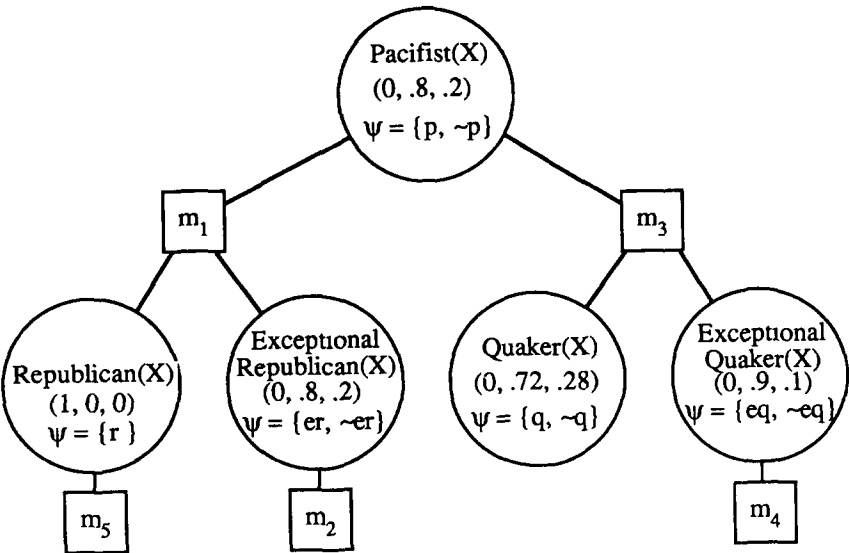


Figure 6. The evidential network with the premise that Nixon is a Republican

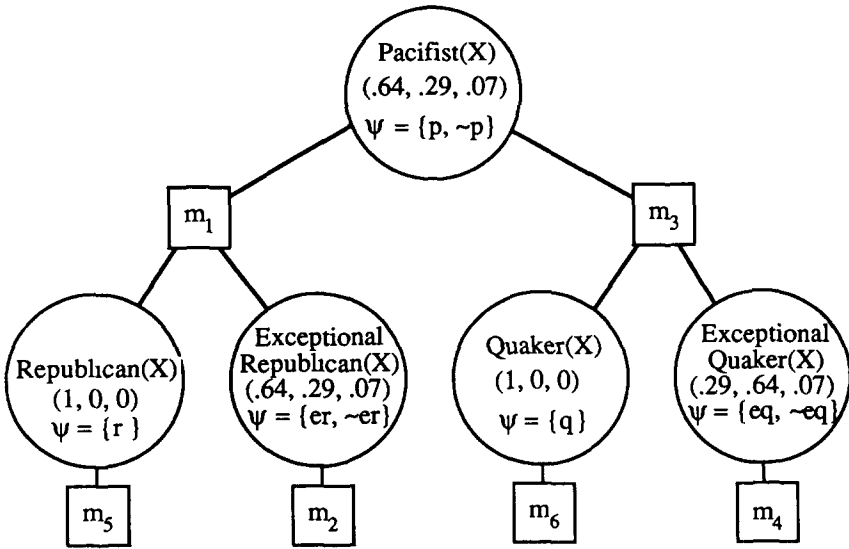


Figure 7. The evidential network with the premise that Nixon is a Quaker

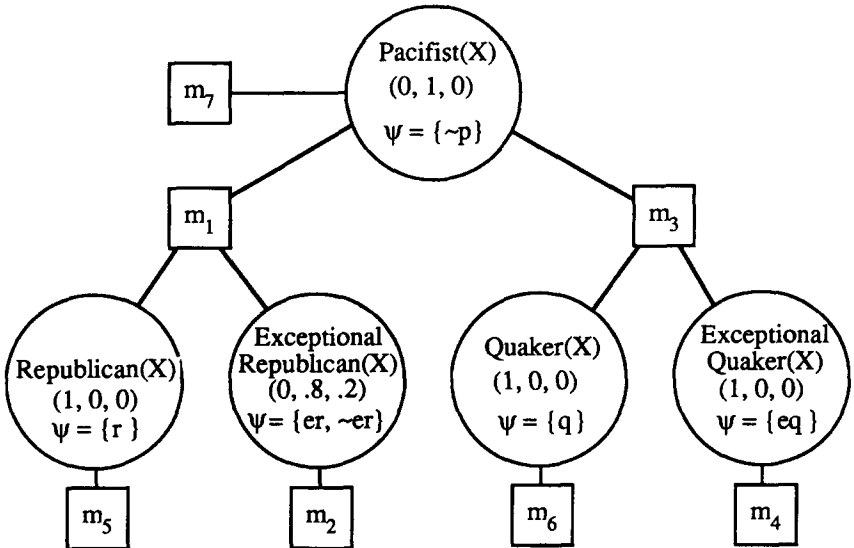


Figure 8. The evidential network with the premise that Nixon is not a pacifist

Now suppose we enter the premise that Nixon is not a pacifist. This is represented by a bpa function as follows:

$$m_7(\{\sim p\}) = 1$$

The ES accepts this bpa function, and the results are displayed in Figure 8. Note that the assumption that Nixon is not an exceptional Quaker has been retracted by the evidence!

SUMMARY AND CONCLUSIONS

The main objective of this article is to introduce a new language for building knowledge-based systems as an alternative to rule-based-languages. Whereas rule-based languages use rules as a knowledge representation device and modus ponens as an operation for making inferences, our language uses proper valuations as a knowledge representation device and three operations—combination, marginalization, and solution—for making inferences. Combination corresponds to aggregation of knowledge, marginalization corresponds to crystallization of knowledge, and solution is a decoding mechanism that maps knowledge from the space of valuations to the space of configurations. Conceptually, the language combines all valuations, finds the marginal of the joint valuation for each variable, and then finds the solution for each marginal.

Like rule-based languages, our valuation-based language retains the modular-

ity feature Each valuation represents a distinct modular chunk of knowledge. If the combination operator is commutative and associative, then, like rule-based languages, valuation-based languages are nonprocedural. These desirable features of rule-based languages are retained

Unlike rule-based languages, our valuation-based language automatically maintains consistency in the knowledge base, caches and displays relevant inferences, reasons nonmonotonically, and permits coherent management of uncertainty

A natural question is, what is the computational power of valuation-based languages? Anderson [67] has formally shown that it is possible to imagine coding any given Turing machine using a pure production system. We suspect that valuation-based languages have the same computational power, but we do not have a proof.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation under grant IRI-8610293 and a Research Opportunities in Auditing grant 87-135 from the Peat Marwick Foundation. The foundation on which this paper rests is the result of research done jointly with Glenn Shafer over the last three years. I owe a lot to Glenn. I would also like to acknowledge the influence of the AI group in the Business School Finally, this paper has benefitted from useful comments from Bruce D'Ambrosio and three anonymous referees.

This paper is a revision of Shenoy [68]

References

- 1 Shenoy, P. P., and Shafer, G., An axiomatic framework for Bayesian and belief-function propagation, Proc 4th Workshop on Uncertainty in AI, Minneapolis, Minn., 307-314, 1988
- 2 Shenoy, P. P., and Shafer, G., Axioms for probability and belief-function propagation, Working Paper No. 209, School of Business, University of Kansas, Lawrence, Kan., 1988
- 3 Shenoy, P. P., and Shafer, G., Constraint propagation, Working Paper No. 208, School of Business, University of Kansas, Lawrence, Kan., 1988
- 4 Shenoy, P. P., and Shafer, G., Axioms for discrete optimization using local computation, School of Business Working Paper No. 207, University of Kansas, Lawrence, Kan., 1988
- 5 Brownston, L. S., Farrell, R. G., and, Kant, E., and Martin, N., *Programming Expert Systems in OPS5: An Introduction to Rule-Based Programming*, Addison-Wesley, Reading, Mass., 1985

- 6 Davis, R , and King, J J , The origin of rule-based systems in AI, in *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project* (B G Buchanan and E H Shortliffe, Eds), Addison-Wesley, Reading, Mass., 20–52, 1984
- 7 Shafer, G , Shenoy, P P , and Srivastava, R P , AUDITOR'S ASSISTANT a knowledge engineering tool for audit decisions, Working Paper No 197, School of Business, University of Kansas, Lawrence, Kan , 1988
- 8 Pearl, J , Fusion, propagation and structuring in belief networks, *AI* 29, 241–288, 1986
- 9 Pearl, J , *Networks of Belief: Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, Palo Alto, Cal , 1988
- 10 Lauritzen, S L , and Spiegelhalter, D J , Local computations with probabilities on graphical structures and their application to expert systems (with discussion), *J Roy. Stat Soc. Ser B* 50(2), 157–224, 1988
- 11 Heckerman, D E , and Horvitz, E J , On the expressiveness of rule-based systems for reasoning with uncertainty, Proc 6th National Conference on AI (AAAI-87), Seattle, Wash , 1, 121–126, 1987
- 12 Shafer, G , and Shenoy, P P , Probability propagation, Working Paper No 200, School of Business, University of Kansas, Lawrence, Kan , 1988 To appear in Proc 2nd International Workshop on AI and Statistics, Fort Lauderdale, Fla , 1989
- 13 Shafer, G , and Shenoy, P P , Local computation in hypertrees, Working Paper No 201, School of Business, University of Kansas, Lawrence, Kan , 1988
- 14 Seidel, R , A new method for solving constraint satisfaction problems, Proc 7th International Joint Conference on AI (IJCAI-81), Vancouver, B C , Canada, 1, 338–342, 1981
- 15 Dechter, R , and Pearl, J , Tree-clustering schemes for constraint processing, Proc 7th National Conference on AI (AAAI-88), St Paul, Minn , 1, 150–154, 1988
- 16 Bertele, U , and Brioschi, F , *Nonserial Dynamic Programming*, Academic, New York, 1972.
- 17 Rose, D J , A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations, in *Graph Theory and Computing* (R C Read, Ed), Academic, New York, 183–217, 1973
- 18 Spohn, W , Ordinal conditional functions a dynamic theory of epistemic states, in *Causation in Decision, Belief Change, and Statistics*, Vol 2, (W L Harper and B Skyrms, Eds), D Reidel, Dordrecht, Holland, 105–134, 1988
- 19 Hunter, D , Parallel belief revision, Proc 4th Workshop on Uncertainty in AI, Minneapolis, Minn., 170–176, 1988
20. Malvestuto, F M , Decomposing complex contingency tables to reduce storage requirements, Proc 1986 Conference on Computational Statistics, 66–71, 1986

- 21 Beerl, C , Fagin, R , Maier, D , and Yannakakis, M , On the desirability of acyclic database schemes, *J. ACM*, 30(3), 479–513, 1983
- 22 Dempster, A. P , Construction and local computation aspects of network belief functions, Research Report S-125, Department of Statistics, Harvard University, Cambridge, Mass , 1988
- 23 Meinhold, R J , and Singpurwalla, N D , Understanding the Kalman filter, *Am. Stat* , 37, 241–288, 1982
- 24 Adams, E , Probability and the logic of conditionals, in *Aspects of Inductive Logic* (J Hintikka and P Suppes, Eds), North-Holland, New York, 1986
- 25 Suwa, M Scott, A C , and Shortliffe, E H , An approach to verifying completeness and consistency in a rule-based expert system, *AI Mag.* 3(3), 16–21, 1982
- 26 Nguyen, T A , Perkins, W A , Laffey, T J , and Pecora, D , Checking an expert system's knowledge base for consistency and completeness, Proc 9th International Joint Conference on AI (IJCAI-85), Los Angeles, Cal , 1, 375–378, 1985
- 27 Pearl, J , Deciding consistency in inheritance networks, Tech Report No 870053 (R96), Cognitive Systems Laboratory, University of California at Los Angeles, Cal , 1987
- 28 Touretzky, D S . *The Mathematics of Inheritance Systems*, Morgan Kaufmann, Los Altos, Cal , 1986
- 29 Ginsberg, A , Knowledge-base reduction a new approach to checking knowledge bases for inconsistency and redundancy, Proc 7th National Conference on AI (AAAI-88), St Paul, Minn , 1, 585–589, 1988
30. McCarthy, J , Circumscription—a form of non-monotonic reasoning, *AI* 13, 27–39, 1980
- 31 McCarthy, J , and Hayes, P J , Some philosophical problems from the standpoint of artificial intelligence, in *Machine Intelligence*, Vol. 4 (B Meltzer and D Michie, Eds), Edinburgh University Press, 463–502, 1969
- 32 McDermott, D , and Doyle, J , Non-monotonic logic I, *AI* 13, 41–72, 1980
- 33 Moore, R C , Semantical consideration on nonmonotonic logic, *AI* 25, 75–94, 1985
- 34 Reiter, R , A logic for default reasoning, *AI* 13, 81–132, 1980
- 35 Shafer, G , Probability judgment in artificial intelligence and expert systems, *Stat. Sci.* 2(1), 3–44, 1987
- 36 Heckerman, D E , and Horvitz, E J , The myth of modularity in rule-based systems for reasoning with uncertainty, in *Uncertainty in Artificial Intelligence*, Vol 2 (J F Lemmer and L N Kanal, Eds), North-Holland, New York, 23–34, 1988

- 37 Shortliffe, E., and Buchanan, B. G., A model of inexact reasoning in medicine, *Math. Biosci.* 23, 351–379, 1975
- 38 Duda, R., Hart, P., and Nilsson, N., Subjective Bayesian methods for rule-based inference systems, in *Readings in Artificial Intelligence* (B. L. Webber and N. J. Nilsson, Eds.), Tioga, Palo Alto, Cal., 192–200, 1981
- 39 Doyle, J., A truth maintenance system, *AI* 12(3), 231–272, 1979
- 40 de Kleer, J., An assumption-based TMS, *AI* 28, 127–162, 1986
- 41 Reiter, R., and de Kleer, J., Foundations of assumption-based truth maintenance systems: preliminary report, Proc. 6th National Conference on AI (AAAI-87), Seattle, Wash., 1, 183–188, 1987
- 42 Laskey, K. B., and Lehner, P. E., Belief maintenance: an integrated approach to uncertainty management, Proc. 7th National Conference on AI (AAAI-88), St. Paul, Minn., 1, 210–214, 1988
- 43 D'Ambrosio, B., A hybrid approach to reasoning under uncertainty, *Int. J. Approximate Reasoning* 2(1), 29–46, 1988
- 44 Dempster, A. P., New methods for reasoning toward posterior distributions based on sample data, *Ann. Math. Stat.* 37, 355–374, 1966
- 45 Dempster, A. P., Upper and lower probabilities induced by a multivalued mapping, *Ann. Math. Stat.* 38, 325–339, 1967
- 46 Pearl, J., Evidential reasoning using stochastic simulation of causal models, *AI* 32, 245–257, 1987
- 47 Kirkpatrick, S., Gelatt, C. D., Jr., and Vecchi, M. P., Optimization by simulated annealing, *Science* 220, 671–680, 1983
- 48 Geman, S., and Geman, D., Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images, *IEEE Trans. PAMI* 6, 721–741, 1984
- 49 Etherington, D. W., More on inheritance hierarchies with exceptions: default theories and inferential distance, Proc. 6th National Conference on AI (AAAI-87), Seattle, Wash., 1, 352–357, 1987
- 50 Shafer, G., and Logan, R., Implementing Dempster's rule for hierarchical evidence, *AI* 33, 271–298, 1987
- 51 Shenoy, P. P., and Shafer, G., Propagating belief functions using local computations, *IEEE Expert* 1(3), 43–52, 1986.
- 52 Shenoy, P. P., Shafer, G., and Mellouli, K. (1986), Propagation of belief functions: a distributed approach, Proc. 2nd Workshop on Uncertainty in AI, Philadelphia, Penn., 249–260, 1986. Also in *Uncertainty in Artificial Intelligence*, Vol. 2 (J. F. Lemmer and L. N. Kanal, Eds.), North-Holland, New York, 325–335, 1988.
- 53 Kong, A., Multivariate belief functions and graphical models, Ph.D. Thesis, Department of Statistics, Harvard University, Cambridge, Mass., 1986

- 54 Kong, A , A belief function generalization of Gibbs ensembles, Tech Report No 239, Department of Statistics, University of Chicago, Chicago, Ill , 1988
- 55 Dempster, A P , and Kong, A , Uncertain evidence and artificial analysis, Research Report S-108, Department of Statistics, Harvard University, Cambridge, Mass , 1986
- 56 Shafer, G , Shenoy, P P , and Mellouli, K , Propagating belief functions in qualitative Markov trees, *Int. J. Approximate Reasoning* 1(4), 349-400, 1987
- 57 Mellouli, K , On the propagation of beliefs in networks using the Dempster-Shafer theory of evidence, Ph D Thesis, School of Business, University of Kansas, Lawrence, Kan , 1987
- 58 Almond, R , Fusion and propagation in graphical belief models, Research Report S-121, Department of Statistics, Harvard University, Cambridge, Mass , 1988
- 59 Zarley, D K., An evidential reasoning system, Working Paper No. 206, School of Business, University of Kansas, Lawrence, Kan , 1988
- 60 Zarley, D K , Hsia, Y T , and Shafer, G , Evidential reasoning using DELIEF, Proc 7th National Conference on AI (AAAI-88), Minneapolis, Minn , 1, 205-209, 1988
- 61 Baldwin, J F , Evidential support logic programming, *Fuzzy Sets Syst.* 24, 1-26, 1987
- 62 Ginsberg, M. L , Non-monotonic reasoning using Dempster's rule, Proc. 4th National Conference on AI (AAAI-84), Austin, Tex , 126-129, 1984
- 63 Rich, E., Default reasoning as likelihood reasoning, Proc. 3rd National Conference on AI (AAAI-83), Washington, D C , 348-351, 1983
- 64 Grosof, B N., Non-monotonicity in probabilistic reasoning, in *Uncertainty in Artificial Intelligence* Vol 2 (J F Lemmer and L N Kanal, Eds), North-Holland, New York, 237-249, 1988
- 65 Shafer, G , *A Mathematical Theory of Evidence*, Princeton Univ Press, Princeton, N J , 1976
- 66 McCarthy, J , Applications of circumscription to formalizing common-sense knowledge, *AI* 28(1), 89-116, 1986
- 67 Anderson, J , *Language, Memory and Thought*, Erlbaum, Hillsdale, N J , 1976
- 68 Shenoy, P P , Valuation systems a language for knowledge-based systems, Working Paper No 203, School of Business, University of Kansas, Lawrence, Kan , 1988