Uncertainty and
Methods for Dealing with Uncertainty.

Gio Wiederhold
Claus-Rainer Rollinger
Josef Gemander
Rudi Studer

IBM Stuttgart, 15Dec1987.

In other words then, if a machine is expected to be infallible,
it cannot also be intelligent.

Alan Turing, 1946 [A. Hodges: *Alan Turing: The Enigma*, p.361; Simon and Schuster 1983]

Abstract

In this note we introduce a proposal to provide a mechanism to deal with uncertainty in a PROLOG environment. The intent is that such a mechanism will enable experimental and comparative research on applications which need to deal with uncertainty. Such research is needed since the variety of uncertainty algebras is large, and it appears that no single algebra will satisfy all applications. Not having a *right* solution is not a reason to avoid the issue. The power of dealing with concepts, that is generalization over many instances, is a capability needed for decision-making, where one must deal with cases not yet instantiated and representable in the database as facts.

In this note we first motivate the need for having the capability to compute with uncertainty, and then discuss criteria for representation and manipulation of uncertainty. We then show and comment on a number of uncertainty measures that have been used in artificial intelligence systems. The proposal is sketched in the final section. We emphasize that such a proposal is not an attempt at a solution; such a goal would not only be overly ambitious but also raise questions at a completly different level, namely is the solution proven to be correct and complete? Instead, the goal is to provide a system tool that will enable other researchers to use uncertainty methods in their work, and test and assess alternative formulations.

1. Generalities

Uncertainty is a common feature of our knowledge. We can distinguish many aspects where knowledge is uncertain. A general classification distinguishes object aspects that may be uncertain: structure, function, or behavior.

Within this classification many reasons and types of uncertainty can occur. Some candidates are:

1 We do not know a data value, it has not been provided. For example, we are not given the length of the road from the bridge to the restaurant. This simple instance is treated in attempts to develop a three-valued logic: `TRUE, FALSE, UNKNOWN`, i.e., [Zaniolo:1985].

2 We are uncertain about a data value, only an estimate is given or can be induced. This case is frequent, since attributes have often limited domains and simple reasoning can often eliminate large sets of alternatives. By applying corrolary information to the first example we can limit the road length to 10 km. Here a solution is to use *Set Nulls*, where the domain of values for an attribute is seen as a finite set and that set is partitioned into the possibly valid candidates and the definitely invalid candidates.
3 We have defined a *natural* classification and encounter a new candidate member. When we cross a ditch to we cross a bridge as well?
4 We are uncertain about the rules used to determine an object classification. For instance, is a three-wheeled enclosed vehicle with a one cylinder motor and a chain drive a motorcycle or a car?
5 We are uncertain about a mapping function, i.e., what does "the patient is seriously ill" mean. [Pinkal?]
6 We are uncertain about the ranking of terms: `likely` versus `possibly not` [Rollinger:1986]
7 We are uncertain about a cause and effect in specific instances, so we can only state a generality, i.e., `Driving on the wrong side of a road is` *likely* `to lead to an accident.`
8 We are uncertain about the completeness of a set: `Does not being married mean that Joe is single?`

As Umberto Ecco has William of Baskerville say in *The Name of a Rose*: "This list could go on and on, and there is nothing more wonderful than a list, but we have to go on ... ", so let us go on.

It appears essential that uncertainty be supported in AI systems, otherwise the power of generalization cannot be exploited. In the examples shown throughout this paper, as well as in published systems of reasonable power, there is a recurruring paradigm of tasks which permit decision-making to be carried out:

1 Individual objects, subject to decision-making are classified
2 Rules, applicable to that class in general are applied to determine alternatives
3 This process reduces the choices to be made and estimates attributes as benefits and risk
4 The result is a ranking of choices, presented for implementation to a decision maker.

Uncertainty exists in all phases of the process, but that is only natural, because the classifications used for assignment and rules are based not on homogenous sets, but on growing collections of varied individuals perceived as being similar by some finite assignemnt scheme.

However, the choice of representation and algebra of uncertainty is still uncertain, but has recently become a subtopic of much interest in AI. We will list some examples later in this note. A personal sense of mine is that there will not be single correct algebra, and that alternate application domains will require alternative methods.

One classification of uncertainty is a distinction of *factual* versus *conceptual* knowledge.

1.1 Uncertainty in Data Instances

When we focus on uncertainty in instances we can associate uncertainty with

1 belief about truth of a fact: I believe (with $p_b = 0.9$) that the restaurant is 10km from the bridge

or with

2 dispersion about the value: The restaurant is about 10km (10km$\pm$2km) from the bridge.

These aspects interact, especially if some computation is performed on these uncertainties. They could be combined into a distribution function where a p_b value is associated with every possible measure, in a simple case 8km, 9km, 10km, 11km, and 12km. In some applications the sum of the p_b is not neccessarily 1.

When we obtain data for a database both aspects are typically involved, although implicitly:

1 apparent total lack of knowledge: `Height(Gertrud) =` Λ.

1.1 Given a domain specification of values for `Height` we can assume, in the first instance, that there is a uniform distribution over the reasonble domain for `Height:{20cm, ..., 250cm}`.

1.2 Given capability for inheritance from `People` we can use a default assumption and change the distribution, over the same domain, to one that matches people in general. Default values are less *believable* than read or computed values, and may have to be distinguished.

2 Lack of sufficiently detailed knowledge, common when processing text: "Gertrud is tall", leads perhaps to a normal distribution around `175 cm`.

Imprecise quantifiers, as `tall`, are used widely in human communications, and convey in one term both a measure and an uncertainty. Other quantifications are provided by predicates, as `most`, `few`, `about`, `often`, `nearly always`.

Uncertainty of data values is pronounced when information from multiple sources is being integrated. Not all sources will use the same metrics and granularity. One solution is to reduce the integrated information always to the greatest or coarsest grain common denominator: `Height {short, medium, tall}`. This solution causes an obvious loss of information when we receive in part more detailed information. Another solution is to introduce uncertainty explicitly.

Facilities to Handle Uncertainty in Data Instances

Database systems today do not recognize the existence of quantifiable uncertainty in instances, and provide only two alternatives:

Λ Null, the value is to be ignored throughout the computation

V The recorded value is taken to be correct.

Some recent work has considered uncertainty explicitly, and a recent thesis provides a formal foundation for manipulation incomplete knowledge for discrete-valued domains, including an algebra for dealing with set-nulls [Winslett, Marianne:1987].

Statistical analysis systems, often combined with specialized databases, do in general provide for unknown values. Again the degree of uncertainty is not quantified. These systems do, however, provide for appropriate statistical computations with unknown valuess. For instance, averages will be computed without counting unknows, and similarily, where statistical procedures can deal with unknowns, they will do so. Where they cannot, they may issue diagnostic messages.

1.2 Uncertainty in Abstractions

In knowledge-based reasoning we deal in abstract concepts of classes which pertain to many instances. We associate knowledge with classes and subclasses, and apply that knowledge to their instances.

Our knowledge can also be missing or uncertain. Incomplete knowledge is a further issue when dealing with abstractions. We often do not know the set of instances that an abstraction applies to, and perhaps not even the set of subclasses for an abstract concept. In that case we work within an *open world assumption*, and that assumption induces further uncertainty.

Abstract concepts, however, permit also manipulation of uncertainty, and perhaps a narrowing of the domain of possible values, with a commensurate increase in belief in the remaining values. As example we can use classification knowledge to improve the possible ranges of `Height(Gertrud)`. Given useful corrolary knowledge that, say, `Age(Gertrud)=25years` and `Nationality(Gertrud)=German`, we can, with some newly computed uncertainty, assume that `160<Height(Gertrud)<185` . While this is still a wide range, it is sufficient to let us further deduce that `Gertrud` can perform certain tasks. And those types of conclusions are, after all, the important objective of intelligent systems.

Manipulating high level concepts is an essential part of Artificial Intelligence. Say that we wish to determine if `Gertrud` is likely to be participating in any of a number of activities. Estimating for `Gertrud` as `activity="Basketball"` versus `"Needlepoint"` may use the `Height(Gertrud)`, as based on `Age(Gertrud)` and `Nationality(Gertrud)` as a criterion.

1.3 Uncertainty in Rules and Methods

When we consider attributes of high level abstractions, not all instances can be expected to satisfy all deductions we can make.

It is reasonable to infer that `Nationality(x)` $\rightarrow$ `Writes(x)`, but there are exceptions. Letting a few violations disallow this inference will weaken the power of reasonable methods.

Requiring all exceptions to be stated explicitly will place a heavy load on the representation, and it is unlikely that all cases will ever be covered.

2. Representation of Uncertainty

We have hinted already that we wish to represent uncertainty by numeric values. If we can assign values to terms as `unlikely`, `not likely`, `perhaps`, `maybe`, `likely`, `I'd think so`, etc. we have achieved a more compact representation than the source terms provided. Of course, the numeric weights of individuals may differ, but an individual lexicon could retrieve the original terms.

More complex, and implied by the assignment of numeric values, is the ability to compute with them. But the issue of numeric encoding itself is independent, and should not be abandoned because of the implication of use and fear of misuse.

How many values are needed to describe the uncertainty associated with a datavalue, with a concept assignment, and with a method? Implementations differ a great deal, but most implementations settle on one representation. Often a single value is used, but some important examples use more than one.

2.0.1 Hierarchies of Uncertainty

We can view an uncertainty assignment, say u, as one, the median, or most probable value. See [Bhatnagar,R.K. and Kanal,L.N.:1986].

2.1 The Mathematics of Uncertainty

Now we come to the more problematical issue: computing with uncertainty. In inference we create chains, and may combine chains. We need to carry uncertainty through those chains, so that some algebra of uncertainty is needed. The most frequent operation is intersection ($\wedge$), used in chaining of conclusions when applying modus ponens. Other operations are union ($\vee$), when two reasoning chains merge, and complement ($\neg$).

Uncertainty repidly leads to a multi-valued representation. A probability distribution is characterized by its mean and its variance. A simple representation may only consider the mean, implicitly holding the variance constant. However, algebraic combinations of means affect the variance <>:

$$mean(A \vee B) = (mean(A) + mean(B))/2$$

$$var(A \vee B) = (var(A) + var(B))/(2\sqrt(2))?$$

A critical question in establishing an algebra is the independence assumption, essential if we manipulate uncertainties formally and independent of context. However, the independence assumption is often violated, as shown in assessment of Gertrud being with the sports team. Explicit rules on the intersection of concepts must then be provided, if generality is to be maintained. With a modest number of concepts the valid subset of the crossproduct should be managable, and be applicable to a very wide range of instances.

			Place			
People	Living	Bathing	Cooking	Working	Nursery	Entert.
baby	.2	.2	.1	.0	.5	.0
child	.3	.2	.2	.0	.2	.1
adult	.1	.1	.1	.4	.1	.2
senior	.3	.1	.2	.1	.0	.3

These tables increase exponentially as the number of alternative values and interacting functions increases, see already Samuel,Arthur L.: "Some:1959].

Some general principles are likely to hold.

Common operations to combine uncertainties should be symmetric. We expect uncertainty and its algebra to be convex, i.e., to remain in the range $\{0,1\}$ in any of its dimensions. Another desired behavior is that it be monotone, expressed as

$$Cert(A \wedge B) \leq Cert(A), \leq Cert(B) \tag{1}$$

$$Cert(A \vee B) \geq Cert(A), \geq Cert(B) \tag{2}$$

In reasoning the intersection ($\wedge$) applies when decisions involving uncertainty are sequential, and condition each other. Union ($\vee$) applies when multiple factors jointly, in parallel, support a result.

A formulation by [Choquet:1954] states more specifically

$$Cert(A \vee B) \geq Cert(A) + Cert(B) - Cert(A \wedge B)$$

These formulations assume that a single scalar value $V, 0 \leq V \leq 1$, is appropriate to represent uncertainty. Not all proposed or used methods follow that assumption.

Possible further references are: [Eddy,W.:1987], also [Levi,I.:1980.], [Cohen,P.R., Shafer,G., and Shenoy,P.P.:1987], [Goodman,I.R. and Nguyen,H.T.:1985], a monograph collecting much formal material.], a number of extended abstracts from the Conference Internationale sur le Traitement d'Information et la Gestion d'Incertitudes dans les Systèmes a Base de Connaissances, [Bouchon:1987], are also relevant.

3. A Survey of Approaches

We will now review the rules of combination used in some current applied systems. Other models are reviewed in [Smith,E.E. and Medin,D.L.:1981.] and [Horvitz,E.J., Heckerman,D.E. and Langlotz,C:1984].

3.1 Logics with Unknowns

As introduced above, the first step in dealing with uncertainty is to posit a new value, UNKNOWN, and develop a logic with that value. The set of alternatives remains small, and a complete logical system is easily constructed.

Even in this simple case a number of variations have been developed which differ in semantics. We consider three well-known systems: Kleene, Lukusiewicz, and Bochvar. They differ in their treatment of composition with UNKNOWN (u): Kleene and Lukusiewicz treat intersection and union identical, but differ in implication. The reason for the difference is that the meaning of UNKNOWN for Lukasiewicz is 'not having a truth value'. Bochvar's logic assign the value meaningless or paradoxical to UNKNOWN. Kleene's assignment seems more appropriate for databases where we deal indeed with incomplete knowledge.

intersection

Kleene, Lukasiewicz	$T \wedge u \rightarrow u, F \wedge u \rightarrow F, u \wedge u \rightarrow u$
Bochvar	$T \wedge u \rightarrow u, F \wedge u \rightarrow u, u \wedge u \rightarrow u$

union

Kleene, Lukasiewicz	$T \vee u \rightarrow u, F \vee u \rightarrow F, u \vee u \rightarrow u$
Bochvar	$T \vee u \rightarrow u, F \vee u \rightarrow u, u \vee u \rightarrow u$

implies

Kleene	$T \supset u \rightarrow u, F \supset u \rightarrow T, u \supset u \rightarrow u$

Lukasiewicz	$T \supset u \to u, F \supset u \to T, u \supset u \to T$
Bochvar	$T \supset u \to u, F \supset u \to u, u \supset u \to u$

3.2 A Four-valued Logic for Planning [Koo]

A planning model developed by [Koo:1987, draft in Lilog library] uses a four-valued logic:

$\{T$(rue), F(alse), Tu(rue with uncertainty), Fu(alse with uncertainty) $\}$

The values marked with Tu, Fu are created when alternative parallel plans are evaluated, so that uncertainty of choice exists. The new rules of combination are only given for union, due to the problem space — alternative plans can only be merged:

$$T \vee Tu \to T, T \vee Fu \to T, Tu \vee Tu \to Tu$$

$$F \vee Tu \to F, F \vee Fu \to F, Fu \vee Fu \to Fu$$

Inconsistent, since that implies inconsistent plans and not permitted due to program logic, are $T \vee F, Tu \vee Fu$. When planning is complete any remaining $Tu \to T, Fu \to F$. WE see here an attempt to distinguish recognize default values and distinguish their computation from set values.

Other logics can obviously be developed to suit given applications. The proposal should handle logics for any number of values, and, as its generalization, real-valued representations in the range from 0 to 1.

3.3 Probability

Probability is the traditional way to deal with uncertainty in a range of 0 to 1. The underlying assumption is one of independence, so that two factors A, B do not intersect. The basic combination functions for A, B are:

intersection	$A \wedge B = A \times B$
union	$A \vee B = A + B$
complement	$\neg A = 1 - A$

These rules satisfy Eqns. 1 and 2.

As shown in the introduction, these rules can profitably be extended with second order rules on the variance or standard deviation, if that value is given as well. A small variance is assiciated with a high degree of knowledge, and a large variance with a low degree of knowledge. Further, third order, rules, may be feasible as well, but have not been explored.

The main problem with these rules in applying them to knowledge-based reasoning, however, is the underlying independence assumption. Attributes of one instance are not independent in the sense that coin-tosses are.

3.4 Uncertainty in MYCIN [Shortliffe]

In this original expert system uncertainty is represented also by a single factor, combining both the concepts of intensity and belief.

The combination functions for A, B are:

intersection $A \wedge B = A \times B$

$$\text{union} \quad A \vee B = A + B - (A \times B)$$

These satisfy the basic constraints, and have been shown to have a Bayesian Interpretation. The reduction of the value of the union operator can be seen as an attempt to avoid the effect of the expected dependence of A and B.

The cases where either A or B, or both, are negative are also defined

$$\text{union} \quad A \vee B = (A + B)/(1 - \min(|A|, |B|)), |AB < 0$$

$$\text{union} \quad A \vee B = A + B + (A \times B), |A, B < 0$$

but these definitions seem awkward. I prefer to treat the sign of values as a seperate component, as in RX.

Negation was originally avoided, negative terms were instatiated as needed. See [Langlotz,C., Shortliffe,E., And Fagan,L.:1986]. The uncertainty factor used can be explained as a probability, given some assumptions [Heckerman,David:1983.]

3.5 Prior and Posterior Probabilities [Bayes]

Bayesian theory provided a probability theory where new information is applied to prior probabilities to yield new, posterior probabilities. The computation applies best to the intersection concept, traditional probability is appropriate for union. Prior knowledge means that measurements of the frequency of occurence of one the factors must be available. The formulation is asymetric. Which side is chosen depends on the available prior information.

$$\text{intersection} \quad A \wedge B \approx A|B = \sum_i (a|b_i)(b_i|B)$$

where the b_i are the prior components.

See [Pearl, Judea:1984] argue against this specific fuzzyness, but I find their counterexample flawed. What is missing from their, and many other models, is recognition of subclasses and then dealing with instances for those subclasses. (Their "pet fish" x is not $x \in (pet \vee fish)$ but $x \in pet - fish \longleftarrow\!\!\!\!\!\in fish$)

[See Zadeh,Lotfi:1983, 1985]; also [Turner:1984] and [Cohen,B. and Murphy,G.l.:1984] discuss fuzzy logic.

A somewhat alternate interpretation of fuzzy reasoning is propounded by [Gaines,P.R.:1978, pp.154–169.]. Here the concept is belief in propositions. Identical computational rules evolve.

Assessing this formulation we see that here independence of belief in A versus B is assumed, up to the point where $A = B$, and then the roles are reversed. This formulation appears to be not acceptable by those who consider that the effect of belief in on factor, say in A, should affect the reulting value throughout the result range.

3.6 Categorical Inferences [Szolovits]

In order to avoid unwarranted computational rules [Szolovits,P., Pauker,S.G.:1978, pp.115–144] insert *categorical* rules into the derivation procedure. Three levels are applied to intermediate uncertainty values:

1 is-sufficient-for

2 is-neccessary-for

3 excludes

An algebra on these values is then developed.

3.7 Endorsements [Cohen]

An endorsement is a symbolic label attached to a concept node. The intent of endorsements is to avoid numeric computation, and deal with propagation in a symbolic way. Endorsements are propagated, and are attached to results of computation. Heuristics may be used to convert combinations of symbols to new symbol instances, so that the explosion of symbols is reduced. The user of the result is presented the remaining set of symbols, but will lack information on where they were applied.

[Cohen,Paul R.:1985.]

3.8 Dempster-Shafer (DS) Theory [Dempster, Shafer]

Uncertainty is presented as a triple: $\{Bel, Pl, m\}$. Bel represents belief, a maximum term, Pl represents plausibility, Pl a minimum term, and m mass, a factor computed to satisfy constraints needed to keep $Pl \leq 1$.

Initial states do not have to satisfy probability or Bayesian constraints, i.e., it is acceptable that $\sum Bl_i \not\models 1$. There is an implication here of 'open-world reasoning'. An example from [Shafer,G:1967] and [Shafer,Glenn:1976.] It is argued that DS does not satisfy the general constraints for a consistent theory. See [Black,Paul K.:1987.] Also [Laskey,Kathryn B.:1987] and [Zadeh,Lotfi:1984.]

3.9 Possibility Theory [Prade]

Uncertainty is visualized here in two dimensions: Certainty versus imprecision. Imprecision is a function on the range

```
|          ---------
|         /         \       certainty e ↑
|        /           \
|-------              ------
|______________________________
.      a b          c d     range →
       imprecision
```

The representation is a quintuple$\{a, b, c, d, e\}$: 2×2 rangepoints, and the certainty. Multiple quintuples can be used for complex configurations. Any remaining uncertainty is spread over the remainder of the range domain.

The $\wedge$ combination function is in principle the geometric intersection.

For more information (not read) see [Dubois,D., Prade,H.:1985F] or [Dubois,D., Prade,H.:1985E].

4. A System to Manage Uncertainty

The survey shows that there is not today any generlly acceptable approach to manage uncertainty. We would furthermore argue that there should not be such an approach, because the appropriate method is linked to the type of application model chosen.

If the representation and the rules of combination are application dependent then we may decide to ignore the issue in support systems for AI and leave the management of uncertainty to the user. As an alternative a user may select some AI system shell which provides an acceptable method.

While the approach of leaving the worry to the user seems attractive there are certainly arguments in favor of support of uncertainty within programmable deductive systems.

1 the communities of system writers, that can provide systems with an adequate computational facilty, and application developers, that may want to use such a facilty are disjoint. We are likely to either see nice implementations of some notions that remain unused beyond demonstrations, or awkward systems linked to some application.

2 any specific group is likely to only implement one choice, so that comparison and evaluation will not be feasible.

4.1 A Proposal in the Domain of PROLOG

We propose that as an extension to a typed PROLOG, as, for instance, [Protos ref] new declarations be provided. These should offer the capability to

1 declare variables as being endowed with uncertainty

2 insert a computational procedures to set or evaluate uncertainty parameters during PROLOG computations.

Some suggestions follow, but will need to be worked out in the context of a research study.

We are not proposing here to also associate uncertainty with the rules themselves. We expect currently that such rules can be written with a variable which specifies their uncertainty. Of course the computation, where the rules operate on uncertain values, will be carried out according to the uncertainty model chosen.

This proposal should be seen as a meta-project, intended to provide a facility to support application-oriented research in the representation and usage of uncertainty.

4.2 The Parameters

To support a variety of approaches it must be possible to endow PROLOG variables with several (0 to perhaps 10) cells, `A, B, C, ...` to hold uncertainty values. A function to initialize the values will be needed, and undoubtedly also a default mechanism for new variables. In most cases the parameter values are likely to be constrined to the interval (0.0, 1.0). An alternative is to use a list of linguistic symbols, following the concept of endorsements, where one parameter is either a subset-list or a one-of list.

At this point we forsee that two distinguished parameters will be required in all algebras.

`A` The first one continues to hold the basic value `TRUE` versus `FALSE`, and that parameter will control unification. In addition to setting this distinguished parameter through the normal PROLOG processes, we will also permit setting of this parameter through a function invokable by the programmer.

B A flag indicating `ACTUAL` versus `DEFAULT` will permit any algebra to decide if other parameter values should be overridden or not.

The semantics of other parameters should depend on the algebra as chosen for the application. If multiple algebras are active, then one more parameter, say `C`, could be used as a switch to indicate the semantics of the remaining ones.

4.3 The Computational Functions

Such a system must now provide for computations on the uncertainty vector. Two types of computational functions are needed. Some functions will be built-in, to be specified directly by application programs, and others will be invoked implicitly during execution of the PROLOG interpreter. For all of these an interface, which permits convenient attachment and perhaps an ability to switch computational routines will be needed.

4.3.1 Built-in Functions

To complement the implicit functions we must provide some built-in functions, where the user can specify the conditions for invocation. The most interesting function, we will name it `SET-TRUTH`, permits setting of the truth value of the first distinguished variable to True or False. This setting controls unification. The value provided by `SET-TRUTH` will be computed according to an evaluation function which operates on the uncertainty parameters and assesses a threshold for True or False.

Controlling unification in this manner determines both scope and cost of computation.

- If the threshold for True is set low, many variables, even of low certainty, can participate in the computation, but the computation will be costly. If aluernatives are to be presented, few will be left out.
- If the threshold for True is set high, few variables, even of moderatly high certainty, will participate in the computation, so that the computation will be cheap. The chance of not evaluating some alternative will be great.

The actual evaluation function is provided by the user, best as part of the interface for the automated functions described in the next section. Some control for these and other invokable function may be provided by a context stack. Finctions to push and pop contexts onto this stack are also needed.

Furthermore we will need functions for output, input, and explicit numeric computation on uncertainty parameters. These must be able to to extract the parameter values and place them into first-class PROLOG variables for explicit comparison, computation, and output. A similar set is needed for input, setting values into these parameters.

4.3.2 Implicit Functions

During processing of PROLOG clauses uncertainty computations are to be implicitly invoked. The function packages to carry out these tasks use an interface to the PROLOG interpreter, so that these packages can essentially be provided by the user. It is appropriate that a few choices, as shown in the previous sections, will be available, since the test procedure will need that.

These packages will be invoked during unification, and will provide the needed results. They must have access to

1 the vector of parameters of the variables participating in the unification step

2 an indication of the type of operation

 2-1 Intersection, due to a PROLOG unification operation.

 2-2 Union, due to a PROLOG operation which sets a result from a number of disjointed clauses within a PROLOG statement.

 2-3 Negation, due to a PROLOG failure.

 2-4 The user-callable function `SET-TRUTH` as described above.

 2-5 Other user-callable function, as those listed above.

3 Context

For context we visualize a context stack, which may be set, perhaps explicitly during PROLOG computations.

5. Conclusion

We propose here a general architecture. In order to validate such an architecture it will be neccessary to test some applications with algebras that are suitable for them.

Within such an architecture new algebras can be tested and compared without a major new effort. Candidate alternatives were shown above. Generalizations may include lattices which exploit type or sort information.

Once the architecture is validated, a new phase can be initiated. By providing representation and algebraic alternatives within a consistent context it will become possible to study formally the appropriateness of such alternatives, and induce eventually some general rules linking scientific AI methods to a *best* representation and computational paradigm.

Without such a tool we must continue to deal with unfalsifiable hypotheses for method evaluation. For applications we would have to continue to use intuition and hunches when selecting computational alternatives.

References

[Bhatnagar:1986] Bhatnagar,R.K. and Kanal,L.N.: "Handling Uncertain Information: A Review of Numeric and Non-numeric Methods"; *Uncertainty in Artificial Intelligence*, Kanal and Lemmer(eds.), N-H, 1986.

[Black:1987] Black,Paul K.: "Is Shafer General Bayes"; *Uncertainty in Artificial Intelligence*, Proc.of an AAAI workshop, 10Jul1987.

[Bouchon:1987] Bouchon,Bernadette and Yager,Ronald R.(eds) (@ CNRS Paris and Iona College, New Rochelle NY) *Conference Internationale sur le Traitement d'Information et la Gestion d'Incertitudes dans les Systèmes a Base de Connaissances*; IPMU, Paris France, Jun-Jul.1987. = Extended Abstracts of an Uncertainty Conference, most are in English. A.o. papers (copied at LILOG) by Lowrance,J.D.; Wesley,L.P. (both at SRI Int); Mellouli,K., Shafer,G., Shenoy,P.P.; Gallant,S.I., Frydenberg,M.; Belvedere,M. et al (ISInf.Bari); Prade,H., Testemale,C.; Ras,Z.W., Zemankova,M.; Oblow,E.M.; Smets,M., Magrez,P.; Clouâire,R.M.; Sabatier,P.C.; Steve,G.

[Choquet:1954] Choquet,G.: "Theory of Capacities"; *Annales de l'institut Fourier*, Grenoble, Vol.5, 1954, pp.131–295. = conditions for an algebra of uncertainty.

[Cohen:1984] Cohen,B. and Murphy,G.L.: "Models of Concepts"; *Cognitive Science* , vol.8, 1984, pp.27–58. = argues about flaws in fuzzy logic and proposes alternatives. Some of arguments are based on weak linguistic examples.

[Cohen:1985] Cohen,Paul R.: *Heuristic Reasoning about Uncertainty: An Artificial Intelligence Approach*; Pitman, 1985.

[Cohen:1987] Cohen,P.R., Shafer,G, and Shenoy,P.P.: "Modifiable Combining Functions"; *Uncertainty in Artificial Intelligence*, Proc.of an AAAI workshop, Jul.1987. = uncertainty functions.

[Dempster:1967] Dempster,A.P.: "Upper and Lower Probalilities Induced by a Multi-Valued Mapping" ; *Annals of Mathematical Statistics* , vol.38, pp.325–339, 1967.

[Dubois:1985F] Dubois,D. and Prade,H.: *Theorié de Possibilities*; Masson, Paris 1985. = Possibility theory, 'soon to appear in English'. 5-valued representation.

[Dubois:1985E] Dubois,D. and Prade,H.: "The Generalized *modus ponens* under sub/min composition, a theoretical study" ; *Approximate Reasoning in Expert Systems*, Gupta et al (eds), N-H, 1985, pp.217–232.

[Eddy:1987] Eddy,W.F.: "Approximating Belief Functions in Rule-Based Systems" ; *Proc. First International Confernce on Statistical Computing*, 1987? = uncertainty functions.

[Gaines:1978] Gaines,P.R.: "Fuzzy and Probability Uncertainty Logics"; *Information and Control*, Vol.38, 1978, pp.154–169. = Here the concept is belief in propositions.

[Goodman:1985] Goodman,I.R. and Nguyen,H.T.: *Uncertainty Models for Knowledge-based Systems*; North-Holland, 1985; in LILOG library. = A monograph collecting much formal material.

[Gupta:1985] Gupta,M.M., Kandel,A., Bandler,W., and Kiszka,J.B.(eds): *Approximate Reasoning in Expert Systems*; N-H, 1985, 834pp.; in LILOG libary. = theory and application papers oriented towards fuzzy reasoning.

[Heckerman:1986] Heckerman,David: "Probabalistic Interpretations for MYCIN's Certainty Factors"; Kanal and Lemmer, 1986.

[Horvitz:1986] Horvitz,E.J., Heckerman,D.E., and Langlotz,C.: "A Framework for Comparing Alternate Formalisms for Plausible Reasoning" ; *AAAI-86*, pp.210–214. = the paper has some confusing statements.

[Kanal:1986] Kanal,L.N. and Lemmer,G.F.(eds): *Uncertainty in Artificial Intelligence*; Elseviers N-H, 1986.

[Koo:1987] Koo, Char-Lin: *A Distributed Model for Performance systems: Synchronizing Plans Among Intelligent agents via Communication* Stanford PhD thesis; Civil Engineering, 1987, in LILOG library. = A planning model developed uses a four-valued logic.

[Kruse:1987] Kruse,R.,Meyer,K.D.: *Statistics with Vague Data*; Riedel, Dordrecht, 1987.

[Langlotz:219] Langlotz,C., Shortliffe,E., and Fagan,L.: "Using Decision Theory to Justify Heuristics"; *Proc. AAAI-86*, pp.215–219. = certainty factors, Bayes rule, and probability.

[Laskey:1987] Laskey,Kathryn B.: "Belief in Belief Theory, an Examination of Shafer's Canonical Examples"; *Uncertainty in Artificial Intelligence*, Proc.of an AAAI workshop, 10Jul1987.

[Levi:1980] Levi,I.: *The Enterprise of Knowledge*; MIT press, 1980. = uncertainty.

[Levitt:1987] Levitt,T.S. and Cheeseman,P.C.(eds): *Uncertainty in Artificial Intelligence*; *AAAI*, Martin Marietta, ADS, 1987. = Proceedings of Jul.1987 AAAI workshop, Seattle.

[Orci:1983] Orci,Istran: *Contributions to Automatic Programming Theory*; PhD thesis, Tech.H.S. Stockholm, 1983,rcvd. Oct.1987. = Broad, includes uncertainty, knowledg-based approaches, induction. Extensive references, including foundations as Polya, etc.

[Pearl:1986] Pearl,Judea: "Fusion, Propagation, and Structuring in Bayesian Networks"; *Artificial Intelligence*, Vol.29 No.3, pp.241–288.

[Rollinger:1986] Rollinger,Claus-Rainer: *Die Repräsentation Naturlishsprachlich Formulierten Wissens-Behandlung der Aspeckte Unsicherheit und Satzverknupfung*; KIT-Report 26, Tech.Univ.Berlin, Fachbereich Informatik, 1986. = Uncertainty in language. Issues as 'likely' versus 'possibly not'.

[Russinoff:1987] Russinoff,D.: "PROTEUS: A Framed-based Non-monotonic Inference System"; MCC Technical Report ACA-AI-302-87, 1987.

[Samuel:1959] Samuel,Arthur L.: "Some Studies in Machine Learning Using the Game of Checkers"; *IBM JRD*, vol.3, pp.210–229, 1959. = uncertainty of combinations represented as 2D decision tables.

[Scott:1976] Scott,Dana: "Does Many-valued Logic Have Any Use?"; *Philosophy of Logic*, Korner(ed.), U.C. Press, 1976.

[Shafer:1976] Shafer,Glenn: *A Mathematical Theory of Evidence*; Princeton Univ.Press, 1976.

[Shafer:1986] Shafer,G.: "Probability Judgement in Artificial Intelligence"; Kanal and Lemmer, 1986. = non-disjointness? Fred being truthful (p=.8) or careless (p=.2).

[Smith:1981] Smith,E.E. and Medin,D.L.: *Categories and Concepts*; Harvard University Press, 1981. = Model of uncertainty.

[Turner:1984] Turner,Raymond: *Logics for Artificial Intelligence*; Ellis Norwood and John Wiley, Chichester, 1984, 121pp.; in LILOG library. = Non-standard logics: Modal logic; Dynamic logic; 3-valued logics; Intuitionstic logics; non-monotonic inference; temporal logic; Fuzzy logic. An easy to read overview.

[Winslett:1987] Winslett,Marianne: *Updating Databases with Incomplete Information*; PhD thesis, Stanford CSD, 1987, in LILOG library; *Proc. AAAI* , 1987; *Proc.Database Theory Conference*, Rome 1987. = an algebra for dealing with set-nulls

[Zadeh:1984] Zadeh,Lotfi A.: "Review Of Shafer's Mathematical Theory Of Evidence"; *AI Magazine*, vol.5, pp.81–83, 1984.

[Zadeh:1983] Zadeh,Lotfi A.: "The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems"; *Fuzzy Set and Systems*, North-Holland Elseviers, 1983, pp.199–217; *Approximate Reasoning in Expert System* , North-Holland Elseviers 1985, Gupta, Kandel, Bandler, Kiszka(eds.), pp.3–31.

[Zaniolo:1985] Zaniolo,Carlo: "On Null Values"; *ACM PODS 1*, 1985. = a three-valued logic: TRUE, FALSE, UNKNOWN