UNCERTAINTY
IN MATHEMATICAL AND LOGICAL MODELS

T. Hürlimann

Working Paper ???

March 1995

Talk at the APMOD 95 Conference, 3–5 April 1995,
at the Brunel University, West London

INSTITUT D'INFORMATIQUE, UNIVERSITE DE FRIBOURG

INSTITUT FÜR INFORMATIK DER UNIVERSITÄT FREIBURG

Institute of Informatics, University of Fribourg

site Regina Mundi, rue de Faucigny 2

CH-1700 Fribourg / Switzerland

tony.huerlimann@unifr.ch

phone: ++41 37 298 328 fax: ++41 37 299 726

Uncertainty in Mathematical and Logical Models

Tony Hürlimann, Dr. rer. pol.

Keywords: Uncertainty, modeling language, mathematical modeling.

Abstract: In many problem domains uncertain, incomplete or even inconsistent knowledge arises in a natural way in the computer-based model-building process. On the other hand, once a model has been build, its semantic must be univocal, complete, and consistent. How can this discrepancy be handled in the model-building process? There exists a wide range of different methods to specify and treat uncertainty. In this paper, several techniques are presented and especially how these methods could be integrated into the model-building process itself. The thesis in this paper is that we need modeling languages which contain elements that support the modeling of uncertain knowledge.

Stichworte: Modellieren von Unsicherheit, Modelliersprachen, Mathematische Modellierung.

Zusammenfassung: In vielen Problemen ist Unsicherheit, d.h. unvollständiges, unsicheres, ja selbst inkonsistentes Wissen ein integraler Bestandteil des Modellformulierungsprozesses. Ist andererseits ein Modell formuliert, so muss seine Interpretation eindeutig und präzis sein, um es einem automatischen Solverprogramm zur Lösungssuche zu übergeben. Wie kann diese Diskrepanz überbrückt werden? Es gibt viele Methoden und Techniken, um Unsicherheit in Modelle einzubauen und diese in geeigneter Weise zu interpretieren. Die These, die in diesem Artikel vertreten wird, ist, dass moderne, mathematische Modellierungssprachen verschiedene Arten von Unsicherheiten direkt in die Sprache einbauen sollten und können, um ein weites Anwendungsfeld von Problemen abdecken zu können.

"Both precision and certainty are false ideals. They are impossible to attain, and therefore dangerously misleading if they are uncritically accepted as guides. The quest for precision is analogous to the quest for certainty, and both should be abandoned. I do not suggest, of course, that an increase in the precision of, say, a prediction, or even a formulation, may not sometimes be highly desirable. What I do suggest is that it is always undesirable to make an effort to increase precision for its own sake - especially linguistic precision – since this usually leads to lack of clarity, and to a waste of time and effort on preliminaries which often turn to be useless, because they are bypassed by the real advance of the subject: one should never try to be more precise than the problem situation demands."
[Popper 1978].

"In general, complexity and precision bear an inverse relation to one another in the sense that, as the complexity of a problem increases, the possibility of analyzing it in precise terms diminishes. Thus 'fuzzy thinking' may not be deplorable, after all, if it makes possible the solution of problems which are much too complex for precise analysis."
[Zadeh 1968].

"Reasoning begins with the recognition of ignorance and uncertainty."
[Walley P.]

INTRODUCTION

This paper is about uncertainty, mathematical, and logical models. Let's first introduce the three concepts.

Our knowledge of the world is *partial* or *incomplete*, because complete information about an object would be too expensive or impossible to collect. Our linguistic notions we use are *vague* or *fuzzy*, because the objects we describe with our natural language have no clear boundary; but we do not consider this as a major shortcoming of our natural language, quite on the contrary – it gives us an efficient and flexible instrument to organize our mental world. Furthermore, our perceptions of the world are often *inconsistent* with our theories and models, because one observation gives us evidence for the truth of a statement, but another observation gives us evidence for the contrary. There may be many reasons for this phenomenon: the sources of the information may be unreliable, there may be imperfect or broken sensors that give rise to conflicting observations, there may be an unreliable informant, or an uncertain rule based on guesses due to, again, incomplete knowledge.

Incompleteness, vagueness, and inconsistency – which in this paper I shall subsume under the broad notion of *uncertainty* – are three main scope in human commonsense and plausible reasoning. Uncertainty is ubiquitous when formulating or revising a theory or a model that pictures a part of our world.

A *mathematical model* is a set of constraints: $G(x)=0 \quad with \quad x \in \Re^n \, (\mathrm{N}^n)$ normally together with a maximizing function $F(x)$. Solving the model means to assign a vector x_o to x such that the constraints are satisfied and the function $F(x)$ is maximized. If no such vector x_o is possible than the model is said to be infeasible, otherwise it is said to be feasible.

A *logical model* is a set S of well formed formula (wff) of first order logic. The programming language PROLOGII, for instance, can be used to code a subset of all logical models. Solving a logical model means to check whether there exists an interpretation or, in other words, whether the model is satisfiable. For the propositional logic this is the well known satisfiability problem (SAT). If no interpretation exist the model is called inconsistent (or unsatisfiable). Instead of a logical model we will also use the term *knowledge base*.
Mathematical and logical models have the important feature in common that their both can be used to represent knowledge in a declarative manner. A mathematical and logical model can even be merged into a single combined model. This is obvious, since any logical model can be expressed as a mathematical model. There exists actually a fast growing research community around the *constraint satisfaction programming (CLP) paradigm* that merges the two paradigms (for an introduction see [Van Hentenryck, 1989]). The CLP paradigm is a powerful method to represent and handle knowledge.

What has uncertainty got to do with mathematical and logical models? Isn't mathematical knowledge not antipodal to uncertain knowledge? Of course, a mathematical formula has always a precise interpretation. However, what is often not so precise is whether the formulas reflect the real problem to be modeled. Everybody who builds large mathematical models knows the most probable answer from the (computer-based) solver at the very beginning of the model building process: "the model is infeasible!" Infeasible? From the point of view of logic this response is disastrous because an infeasible model has no solution and is completely useless because anything follows from it. But what is the reaction of the model builder towards such a situation? Does he or she throw the model away? Not at all! Infeasibility or inconsistency does not necessarily mean that the model does not reflect our knowledge about the problem at hand. Our knowledge might be – and frequently is – inconsistent and our goals are incoherent with each other.
The model revision process is another operation when a model can become inconsistent: adding a new constraint to an otherwise well behaved and feasible model, for instance, is almost never a trivial task. But there are still other circumstances in which uncertainty pops up in the modeling process: certain constraints do hold only with a given probability, numerical data are not known exactly, only their probabilistic distribution is known, or even not that. Hence, inconsistent and imprecise models arise in a natural way in the mathematical model building process itself.

The goal of this paper is to show methods to support the model building process using computer-based modeling tools. To be usable in a large range of applications, such a toolbox should integrate

(1) A specification language in which a large range of models can be formulated,

(2) A large number of methods to solve the models,

(3) Methods to handle uncertainty,

(4) Interface and graphical tools to manipulate the models and the data by an "end-user".

This paper is about how (3) can be integrated with (1).

The paper is divided into 6 sections. The first section presents briefly the decision making paradigm. The next section gives a short survey of approaches on how to model uncertainty. Some approaches in mathematical modeling are summarized in the third section. Section four and five treat numerical and symbolic approaches in logical models. The last section exposes how fuzzy systems could be integrated into modeling languages.

DECISION MAKING

Models are built to clarify the situation, to gain insight and understanding of the problem at hand. Models are the bases of our mental representation of a real problem. Insight and models give us a more solid base of making better decisions. The better the model the better the decision for attaining a certain goal! Models may take many shapes: they may be just a drawing, a sketch of points and lines, or they may be a sophisticated graph or a formal description. In a broad sense, we can define model as an collection of abstractions to represent knowledge.

The best we can get is a complete, precise representation of the problem in a mathematical or logical formulation. But this is rarely possible for complex problems. Either the data are not available or the situation cannot be formulated in a precise manner. But whether we have a precise model or not, we are nevertheless forced to make a decision.

Human decision makers do, in general, very well when making decisions under vague, incomplete or even inconsistent knowledge. They do not need a mathematical representation and rarely do calculation to get a reasonable result. They estimate a situation using their "feelings" and reason by "intuition". How is human reasoning distinguished from automated, computer-based, or logic-based reasoning? Why is human reasoning so powerful?

In contrast to human-based reasoning, computer-based decision making is (still) rigid and inflexible. Of course, this do not mean that computer-based decision making – as it is today – is not useful or does not have some practical application; quite on the contrary, never in the history before have we been able to solve, say, big LPs which usefulness cannot be denied. We can solve practical problems today which would be inconceivable without the rigidity of logic and the speed of computers.

Still, we may ask the question why human commonsense based reasoning is so powerful and why logic-based reasoning sometimes behaves so poorly in practical decision making and in solving problems. Is it because computers capacity in speed and memory is still very limited compared with the human brain or is it because commonsense-based reasoning is not based on the classical logic paradigm? Experiences from the last 50 years show that faster computers and more efficient algorithms – based on logic – has certainly greatly enlarged the class of problems – in the sense of complexity not in the sense of computability – we can practically handle and eventually solve on a computer. This is a remarkable fact, since the most sophisticated present computer can be reduced to a simple Turing machine, invented by A. Turing 60 years ago even before the first computer was manufactured. But it is probably also true that improved computers and better algorithms in the future will still *not* solve most problems a human brain can "solve". This is, I think, not because the human brain seemingly could solve uncomputable functions (in the sense of Turing) as some researcher have suggested (Penrose, Eccles), but because the human brain does not necessarily follow a deductive inference when finding new results or when updating its "knowledge base"; it follows what some have called an "inductive inference", whatever this means. An important difference between computer-based and commonsense-based reasoning is certainly, that human can reason under incomplete and inconsistent knowledge. We have subsumes this under the broad concept of uncertainty.

Uncertainty is not a new paradigm of an academic research, neither is it a very original one. It has been studied extensively by decision analysts, statisticians, logicians, philosophers, (cognitive) psychologists, gamblers, insurance companies, and – of course – by practitioners in artificial intelligence.
It would be audacious to depict even a superficial overview of the

representation of uncertain knowledge within this paper. I select some approaches that seemed to me especially promising in the future to model uncertainty in mathematical and logical models. Of course, the selection is subjective and in a certain sense arbitrarily. This is unavoidable in a domain that is still in development. In addition, the main goal of this paper is to show how uncertainty could be integrated into the declarative modeling language paradigm, but it certainly not the goal to give a complete survey on how to model uncertainty.

SURVEY OF APPROACHES

There has been and is still a lot of debate on which formalism to measure or to handle uncertainty is the most general and many researchers have defended the view that there is not a most general formalism but that different formalisms are better to measure different facets of uncertainty. Some defend the position that the right approach is numerical (p.e. probability theory), others argue that the only approach should be symbolical (p.e. non-monotonic logic).

Disputes could even arise about how to classify various approaches. Depending on the criteria and goal we are interested in, a classification of approaches could be very different.

Pearl [1988] classifies the approaches to uncertainty into three "schools": logicists, neo-calculists, and neo-probabilists. The logicists try to deal with uncertainty using nonnumerical techniques, typically non-monotonic logic; the neo-calculists uses numerical representations of uncertainty but regards probability calculus as inadequate and propose entirely new calculi, like Dempster-Shafer, fuzzy logic, and certainty factors; the neo-probabilists remain within the traditional framework of probability theory.

The viewpoint adopted in this paper is centered around the question: how can uncertainty be integrated into mathematical and logical modeling tools as a declarative language. From this perspective, one might distinguish three paradigms (see the references):

- the paradigm of probability theory,
- the paradigm of argumentation theory,
- the paradigm of fuzzy set theory.

The first paradigm of probability theory is based on the assumption that uncertainty can be ranged in degrees which are expressed as numbers. The paradigm is very powerful. All classical approaches of statistics and stochastics (stochastic programming), as well as Bayesian theory and nets and probability logic [Nilsson 1986] can be covered by this approach. The second approach of argumentation theory might not be known widely under this term. One could subsume most symbolical approaches, such as argumentative systems, non-monotonic logics, evidence theory, and hint theory; somewhat unexpectedly, but one could also subsume the numerous goal programming techniques under this approach. The reason will be clarified later on. The third approach of fuzzy set theory founded by Zadeh [1965] has already proven its usefulness in dynamic systems. Two examples will be given in this paper that illustrate this approach.

Uncertainty can appear in very different contexts within mathematical and logical models: In mathematical models, the data are random variables instead of crisp, the constraints are "soft", that is, they may be violated to a certain extent, constraints are true with a certain probability, or several inconsistent objectives enter the model. Logical models can be inconsistent, or become inconsistent by adding new knowledge, they may contain formulas that are not strictly true or false, but only true with a certain probability.

Random or vague data in mathematical models can be approached by stochastic programming or fuzzy programming. Soft constraints, multiple objectives and infeasible models are addressed by different variants of goal programming. Probabilistic logical formulas can be treated using Bayesian networks or probabilistic logic. Finally, bases argumentative systems are suitable to handle inconsistent knowledge bases.

For each kind of uncertainty, we will give examples in a unified framework of representing the model in a declarative way.

CLASSICAL APPROACHES IN MATHEMATICAL MODELING

There are some well known techniques to handle certain uncertainties in mathematical models. Sensitivity analysis on the cost vector and the right-hand-side in LPs belong to the standard package of every commercial LP code. It belongs to the LP folklore and is not discussed in this paper.

It is necessary to have methods to eliminate infeasibilities from mathematical models. In contrast to most textbook examples, few models are feasible straightaway. Experiences show that from a randomly generated set of LPs with a total of contraints+variables of 10, about 12% are feasible, half of the rest is unbounded and half is infeasible; if the number of constraints+variables is 20, the proportion of feasible LPs drops to 7%; if the total rise to 40 the percentage of feasible LPs goes down further to about 2% (see Figure 1). Of course, real live models are not random models, but the experiences show the chance to be feasible is very low for big LPs.

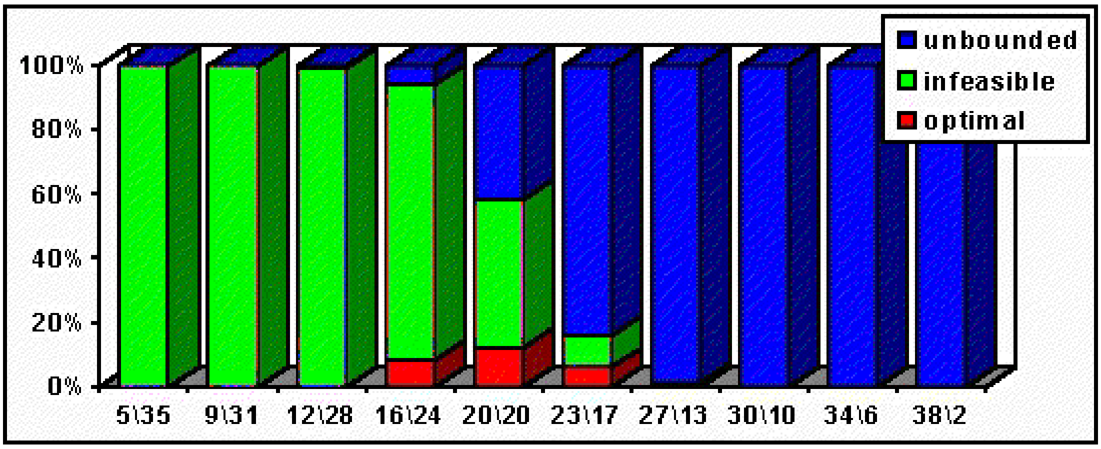


Figure 1: Each bar represents 100 uniform random LPs with n\m (var\const)

In this section goal programming [Ignizio 1976, Zionts 1988] and stochastic programming [Sengupta 1972, Kall 1976, Ermoliev/Wets 1988, Birge/Wets 1991, Wets 1991] are discussed very briefly. In the last section, LP modeling using fuzzy sets will be mentioned. All this techniques are well-known and there is nothing new about them. What is new is to use these techniques directly within a model language specification.

SLACK VARIABLES AND GOAL PROGRAMMING

Introducing slack variables is probably the most used method to eliminate infeasibilities from a mathematical model. These slack variables can be interpreted as penalties for violating the constraints. The objective is then to minimize the "amount of infeasibilities" by a weighted vector of the slack variables. This technique is well known in practical modeling and many variants are in use. Hence, a linear model $\{\min cx | Ax \le b\}$ is replaced by another linear model $\{\min cx + wp + vn | Ax - p + n = b\}$ where p and n are the positive and negative slack variable vectors and w and v are two weight vectors. Adding slacks can also be applied to handle conflicting objectives or goals in mathematical modeling. Suppose there are two objectives, one is to *maximize cx* and the other is to *minimize dx*. The modeler can introduce two aspiration levels C and D (defined as scalars), one for the first the other for the second objective, which indicate the desired goal to attain of the objectives. The objectives are now converted into constraints containing additional positive and negative slack variables as following:

$$cx - slackC_{pos} + slackC_{neg} = C$$
$$dx - slackD_{pos} + slackD_{neg} = D$$

The new objective function is now to minimize a weighted sum of the four slack variables, or to minimize the largest slack variable; many variants are possible here too.
Another way to look at slack variables is to interpret them as "softening the constraint". Instead of having precise, "hard" constraints which is almost never the case in practical modeling, one has constraints that can be violated to a certain extent. In this way the modeler can formulated all kind of fuzzy or uncertain situations. An example is the constraint that fixes the budget on *about* 1000. One could formulate this as $Budget \cong 1000$ or, using a positive and negative slack variable *p* and *n*, as $Budget - p + n = 1000$.
Still another situation where slack variables are useful is when interpreting the constraints as goals to be attained. The right-hand-side of the constraint is now considered as an aspiration level. Every constraint, in fact, is now restated as an objective and the approach of conflicting goals just exposed above can be applied. This fourth approach is also known as goal programming.

All four applications (elimination of infeasibilities, handling of conflicting objectives, soft constraints, and goals) show how useful the technique of introducing slack variables can be. We will see that this technique has a parallel in getting rid of inconsistencies in logical knowledge bases by introducing new propositions, called assumptions. Hence, this method is very general and it can easily be integrated into a mathematical modeling language.

Schniederjans [1995] reports four main variants of goal programming:

- The nonpreemptive, nonweighted GP model
$$\left\{ \min \sum_{i=1}^{m} (s_i^+ + s_i^-) \middle| \sum_{j=1}^{n} a_{ij} x_j - s_i^+ + s_i^- = b_i \right\}$$
- The nonpreemptive, weighted GP model
$$\left\{ \min \sum_{i=1}^{m} (w_i^+ s_i^+ + w_i^- s_i^-) \middle| \sum_{j=1}^{n} a_{ij} x_j - s_i^+ + s_i^- = b_i \right\}$$
- The preemptive, nonweighted GP model (sometimes called the lexicographic GP model) (defining a absolute priority structure)

$$\left\{\min \sum_{i=1}^{m} P_i (s_i^+ + s_i^-) \middle| \sum_{j=1}^{n} a_{ij} x_j - s_i^+ + s_i^- = b_i \right\}$$

– The preemptive, weighted GP model

$$\left\{\min \sum_{i=1}^{m} P_i \sum_{k=1}^{n_i} (w_{ik}^+ s_i^+ + w_{ik}^- s_i^-) \middle| \sum_{j=1}^{n} a_{ij} x_j - s_i^+ + s_i^- = b_i \right\}$$

To illustrate the modeling aspects of slack variables, we give an example found in the literature [Beilby/Mott 1983]: In allocating academic and research library budget for books and periodicals the librarians are faced with the task of satisfying the needs and interests of groups which are competing for library resources. Suppose the goals in allocating the budget are:

– Goal 1 (g1): Acquire at least 7500 titles but no more than 10700 titles.
– Goal 2 (g2): Do not exceed total acquisitions budget of 300000.
– Goal 3 (g3): Limit periodical expenditures to 60% of the total acquisitions expenditures.
– Goal 4 (g4) : Limit periodical acquisitions to the level which can be supported for a five-year period.
– Goal 5 (g5): Allocate titles by subject according to circulation data (550 is an arbitrary unit).
– Goal 6 (g6): Allocate titles by subject according to enrollment data.
– Goal 7 (g7): Limit research acquisitions to 15% of the total acquisitions and allocate on the basis of departmental research productivity.
– Goal 8 (g8): Limit retrospective acquisitions to 5% of total acquisitions and allocate on the basis of retrospective subject needs.
– Goal 9 (g9): Meet desired subject acquisition ranges.

In LPL, a mathematical modeling language [Hürlimann 1993], one can formulate these goals as following (the model data are cut to be concise):

```
SET  i = / Books , Periodicals / ;
     j = / Humanities , Social_Science , Sciences , Education ,
Interdisciplinary /;
COEF
  avrCost{i,j}; (* average cost of a title *)
  CostPp{j};    (* projected average costs for periodicals for five years *)
  cir{i,j};     (* circulation data *)
  e{i,j};       (* enrollment percentage of total book and periods enrollment
titles *)
  pr{j};        (* research productivity *)
  ret{j};       (* retrospective data *)
  low{i,j};     (* minimum percent level of acquisition *)
  up{i,j};      (* maximum percent level of acquisition *)
```

```
VAR x{i,j};

MODEL
  g1:          SUM{i,j}  x  =~ 7500;
  g1a:         SUM{i,j}  x  =~ 10500;
  g2:          SUM{i,j}  avrCost*x  =~ 300000;
  g3:          SUM{j} avrCost[2,j]*x[2,j] =~ 0.6*( SUM{j} avrCost[2,j]*x[2,j] );
  g4:          SUM{j} CostPp*x[2,j]  =~ 193261 ;
  g5:          SUM{i,j} cir*x =~ 550;
  g6{i,j | e}: x[i,j] - e*(SUM{j | e} x[i,j]) =~ 0;
  g7:          SUM{j | pr} pr*x[1,j] - 0.15*(SUM{i,j} x) =~ 0 ;
  g8:          SUM{j} ret*x[1,j] - 0.05*(SUM{i,j} x)  =~ 0;
  g9{i,j}:     x - low*(SUM{i,j} x)  =~ 0;
  g9a{i,j}:    x - up*(SUM{i,j} x)  =~ 0;

   (* preemptive priority A[1]-A[8] *)
EQUATION A :  Ng1+Pg1a, Pg2, SUM{i,j} (Ng9+Pg9a), Ng5,
              SUM{i,j|e} IF(j<=2,Pg6,Ng6), Pg7+Pg8, Pg3+Pg4, Ng2+Ng3+Ng4;

   (*--- preemptive priorities as multi-stage minimizing *)
MINIMIZE  M: A[1];  COEF Ng1; Pg1a;
MINIMIZE  M: A[2];  COEF Pg2;
MINIMIZE  M: A[3];  COEF Ng9; Pg9a;
MINIMIZE  M: A[4];  COEF Ng5;
MINIMIZE  M: A[5];  COEF Pg6; Ng6;
MINIMIZE  M: A[6];  COEF Pg7; Pg8;
MINIMIZE  M: A[7];  COEF Pg3; Pg4;
MINIMIZE  M: A[8];

PRINT x;
END
```

Note the operators =˜ that mean “about equal”. In producing a crisp LP, the LPL language interpreter actually produces positive and negative slack variables and replaces the =˜ operator by a hard equal sign (=). This might be considered as a trivial task and one could argue that the =˜ operator is not needed since it could be replaced by the equal sign together with slack variables manually. But one should then consider two points: first, the =˜ operator might be interpreted differently depending in another context, and second, in the library model, the modeler is forced to introduce explicitly 22 variables as following:

```
VAR x{i,j};
       Pg1; Ng1; Pg1a; Ng1a; Pg2; Ng2;
       Pg3; Ng3; Pg4; Ng4; Pg5; Ng5; Pg6{i,j}; Ng6{i,j}; Pg7; Ng7; Pg8; Ng8;
       Pg9{i,j}; Ng9{i,j}; Pg9a{i,j}; Ng9a{i,j};
```

LPL generates them automatically by adding a leading capital letter P or N to the constraint name, such that they can be used in the objective function.

Of course, there are other methods to eliminate infeasibilities from a model. One method is to find a irreducibly inconsistent set of constraints (IIS). If one member of a IIS is dropped, then the set becomes feasible. For further details see Chinneck/Dravnieks [1991] or Tamiz/Mardle/Jones [1995]. Some commercial solver include procedures to find the IISs.

Other methods are to order the constraints following some preferential criteria and select a maximally feasible subset of constraints in this order. This can also be seen as priority ordering of constraints.

STOCHASTIC PROGRAMMING

A mathematical model that includes one or several random components (data) can be approached by stochastic programming. It covers uncertainties about data. Crisp data are not very realistic in most practical decision models. Stochastic programming provides for a mean to replace the uncertainty about data by a probability distribution. Insofar this approach can be subsumed under the numerical approaches mentioned above.
As a rule, stochastic models are much more difficult to solve, and it is unlikely to find a single efficient solver to handle all of them. Much efforts have been concentrated on finding efficient solver for the stochastic linear programs (SLP) and many efficient solvers are known for special model cases [Kall/Mayer 1993], whereas in integer programs or non-linear models almost no progress has taken place.

If some data within an LP such as $\{\min cx | Ax = b, x \geq 0\}$, are replaced by random distributions as in $\{\min c(\omega)x | A(\omega)x = b(\omega), x \geq 0\}$ where $(\Omega, 2^{\Omega}, p_{\omega})$ is a probability space with $\omega \in \Omega$, then the model becomes, strictly speaking, meaningless, since it is not clear what to minimize in this case. There are, however, several methods to interpret such models. Two main interpretations can be distinguished and are often used [Kall 1976, Kall/Mayer 1992]. The first is the two-stage model where the model above is interpreted as $\{\min E[c(\omega)x + Q(Ax - b, \omega)] | x \geq 0\}$ where E[•] is the expectation of the random variable ω and Q is a penalty function for violating the constraints; the second is the chance-constrained models which can be formulated as: $\{\min E[c(\omega)x] | P\{\omega | A(\omega)x \geq b(\omega)\} \geq \alpha, x \geq 0\}$, if there is a single joint change constraint, or $\{\min E[c(\omega)x] | P\{\omega | A_i(\omega)x \geq b_i(\omega)\} \geq \alpha_i, x \geq 0\}$, if each constraint i is defined as a separate change constraint.

In the simplest case, where only the right-hand-side vector b is a random variable, one can translate the SLP into a (much bigger) crisp LP. To illustrate how stochastic data could be integrated into the specifications of a modeling language such as LPL [Hürlimann 1993], we take the already classical example of airplane allocation defined by Dantzig [see King 1988]:
An airline wishes to allocate airplanes of various types (i) among its routes (j)

to satisfy an uncertain passenger demand (*h*), in such a way as to minimize operating costs plus the lost revenue from passengers turned away. The following code is a complete formulation of the model in an extended (not yet implemented) version of LPL [Hürlimann 1993].

```
(* aircraft.lpl : formulated in a superset of LPL *)
SET
  i = /1:4/;  (* aircraft types *)
  j = /1:5/;  (* routes *)

COEF
  c{i,j};     (* cost of operating type i on route j *)
  q{j};       (* revenue lost per passenger turned away on route j *)
  b{i};       (* number of aircraft available of type i *)
  t{i,j};     (* passenger capacity on aircraft i and route j *)
  h{j} STOCHASTIC;    (* passenger demand (a random variable) *)

COEF
  c{i,j} = [18 21 18 16 10 , . 15 16 14 9  , . 10 . 9 6 , 17 16 17 15 10];
  q{j}   = [13 13 7 7 1];
  b{i}   = [10 19 25 15];
  t{i,j} = [16 15 28 23 81 , . 10 14 15 57 , .  5 . 7 29,  9 11 22 17 55];
  h{j};

  h{1} = RndDiscrete((200 0.2) (220 0.05) (250 0.35) (270 0.2) (300 0.2));
  h{2} = RndDiscrete((50 0.3) (150 0.7));
  h{3} = RndDiscrete((140 0.1) (160 0.2) (180 0.4) (200 0.2) (220 0.1));
  h{4} = RndDiscrete((10 0.2) (50 0.2) (80 0.3) (100 0.2) (340 0.1));
  h{5} = RndDiscrete((580 0.1) (600 0.8) (620 0.1));

VAR
  x{j,i | c};     (* aircraft type i assigned to route j *)
  (* vp{j}; vn{j};   (* empty seats , passenger turned away *) (* slack
variables *)

MODEL
  Cost: SUM{i,j} c*x + SUM{j} q*Nca;
  cap{i}: SUM{j} x <= b;
  ca{j}: Pca-Nca = SUM{i} t*x - h;
  (* or : ca{j}: SUM{i} t*x =~ h; *)
MINIMIZE cost;
PRINT cost; x;
END
```

The only difference with a deterministic LP is the random vector *h*. It is defined to be *STOCHASTIC* (a new keyword in the specification language) and the values are assigned as a randomly discrete distribution (the function *RndDiscrete*). The model as specified above represents a complete SLP model; no other data or model structure is needed. Hence, it can be used to produce the corresponding crisp LP or whatever representation a special solver needs to solve the problem. Kall/Mayer [1993] describe an implementation of a model management system that does just that job. LPL already includes several random distribution function which could also be used: uniform, normal, neg-exponential and others.

A change constraint $prob\left(\sum_{j=1}^{n} a_{ij}x_j \le b_i\right) \ge \alpha_i$ could be formulated in the modeling language as

```
COEF alpha{i};
```

```
MODEL R{i} PROB alpha[i]: SUM{j} A[i,j]*x[j] <= b[i];
```

where PROB is a new keyword indicating a probability attribute for the model entity R.

There is, however, a sore point that is far to be solved at this time about this approach: how does the modeling language know the representation the solver needs? In a crisp LP, there exists a commecial standard to more simplex solver, that is the MPS-file. No such standard exists for SLPs. This problem appears in many different contexts too. It is not the objective of this paper to treat this problem. But let's briefly mentioned the two extreme answers to this problem: either the modeling language is an open toolbox (or a dynamic link library) that can be extended at will or it must be a complete programming language. Neither of both approaches is entirely satisfiable.
Besides this problem, the model specification above is powerful enough to be considered.

The next two sections describe numerical and symbolic approaches for specifying logical models.

NUMERICAL APPROACHES IN LOGICAL MODELS

Endless confusion have taken place since at the end of the 17th century the two concepts of chance and of (epistemic[1] or subjective) probability merged and are now known as “probability”. Before Bernoulli these two concepts were clearly distinct and used in different contexts. Change was used in the context of random games to express the long-run relative frequencies of an event, for example the event that the outcoming number from tossing a die is six. Probability was used in the sense of how much a person believed a statement to be true. Bernoulli himself brought these two concepts together but distinguished them clearly. He examinated even non-additive probability theory to model “subjective” probabilities. Such considerations fell into oblivion until recently. (For a historical and analytical overview of Bernoulli's probability concepts see Shafer [1978].)

subjectivist view see Lindley EJOR 1982 213-222.
Lindley 1985: Making decisions Wiley.

---

[1] epistemic: from the Greek word η επιστημη which means “the whole what we know and believe”.

On the one hand, probability was thought of as a measure of s feature of events, their chance; on the other hand, probability was thought of as a measure of opinion, a measure of degree of belief or credence.

Probability theory is, first of all, a mathematical theory that is based on the following three axioms (where Θ is the event space and p: $2^{\Theta}\varnothing[0,1]$):

(1) $p(\varnothing) = 0$

(2) $p(\Theta) = 1$

(3) if $A, B \subseteq \Theta$ and $A \cap B = \varnothing$ then $p(A \cup B) = p(A) + p(B)$

and has nothing to do with "frequencies of physical events", "degree of beliefs in a proposition", or whatever interpretation we may graft on the notion of probability. It is like geometry and vector spaces which have nothing to do with the physical space; one could model it using Euclidean or non-Euclidean space; Einstein chose the one that produced "simpler physical laws". That was one of the fundamental aspect in the revolution he initiated. He detached the mathematical theory from their interpretation in physics. Although most statisticians, today, have a pragmatic point of view on this matter concerning the theory of probability, the consequences from this observation are still not widely known. The main consequence is that it is not automatically given whether this theory can be used to model different aspects of our world. Most statisticians take the conjecture for granted that all phenomena that have to do with randomness, chance, degree of our beliefs, and other phenomena that are sometimes true and sometimes false, can be modeled using probability theory.

BAYESIAN NETS

The revival of Bayesian statistics is another example

see also Corman p 105ff.

Bayesian inference is a method for updating the probability of a hypothesis given a prior distribution and given – in general a huge number of – stochastically independencies.

The same is true for the Bayesian theory which has nothing to do with “objective” or “subjective” probabilities – although most Bayesians are (sometimes fervent) “subjectivists”.

Normally, it makes sense to model frequency as probability, since frequencies are additive (explain with die).
Sometimes it makes sense too to model degree of beliefs as probabilities (explain again with die). But sometimes this does not make sense (vase).

Bayesian approach = revision of theory
"the modification of beliefs is an essential activity for men as well as for evolutionary systems." three types of modifications: expansion of the knowledge base --> inconsistency may be introduced; revision --> due to expansion; contraction:
updating probabilities of a hypothesis given some evidence
Example:
Hypothesis: the stock of the company XYZ will raise

, testing hypotheses, diagnosis,

disadvantages:

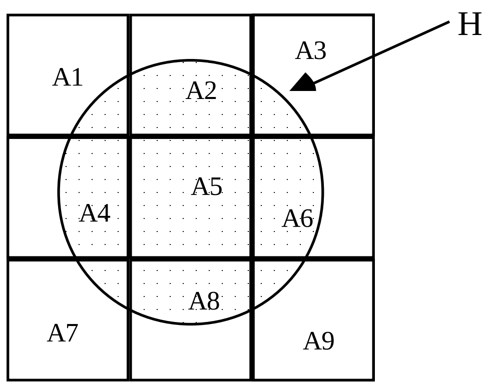
– huge amount of data is required

The philosophy of Bayesian statistics would be stated as:

revising current beliefs in the light of new information. (=learning!)

The question become then: how to represent beliefs, and how to update current beliefs into revised beliefs when new data are obtained? The answer to the first question is: as probabilities!

But the prior probability distribution inevitably dominate any calculation!



$$p(A_1|H) = \frac{p(A_1 \cap H)}{p(H)}$$
$$p(A_1|H) = \frac{p(H|A_1) \cdot p(A_1)}{\sum_{i=1}^{9} p(H|A_i) \cdot p(A_i)}$$

....(not finished...

PSAT

The first to make real progress in logic since the time of Aristotle was Boole . His brilliant idea was to apply the laws of algebra to logic. In doing so, he sought a way to analyze Aristotle's syllogistic logic[2]. But his main goal was nothing less than “to investigate the fundamental laws of those operations of the mind by which reasoning is performed”, or in modern terms: given the truth or falsity of any proposition, what is the truth of a compound statement or the more difficult problem – known as satisfiability problem (SAT) – given several compound statements, which assignment makes them all true.
This is all well known now! But Boole did much more: he was probably the last for more than hundred year who tried to unify logic with the probability theory. He intended to base the theory of probability on this logical caculus by identifying “event” with “proposition”. Suppose – he said – all propositions are not simply true (1) or false (0), but a probability between zero and one is attached to them, where the propositions must be some “unconditioned simple events” (as he said) [see Hailperin, 1986, p. 223], how could now the probability of any compound statement be calculated; or in modern terms, given a set of compound statements, what probabilities must the proposition have to make the system consistent? Again just apply the laws of algebra, he said. Given, for example, the probabilities of the propositions x and y as p(x) and p(y), then the probability of xy is just p(x)p(y), if the events are independent, the probability of x+y is p(x)+p(y), if the event are incompatible, and the probability of –x is 1–p(x). Conditional probability was seen by Boole only as another probability of a compound statement: the probability that the

---

2 As Devlin [1994, p. 45] noted, two syllogisms found in Aristotle's original treatment were false and they went undiscovered for 2000 years! The first has the form:

All M are P
All M are S

Some S is P

Using Boolean logic it is easy to verify that this sylogism is false:

from $m(1-p) = 0$ and $m(1-s) = 0$, it does not follow that $sp \neq 0$!

event y will occurs supposing the event x occured already is just p(xy)/p(x). Clearly this gives the correct value of the conditional probability, noted today as p(y|x). But it is not clear, whether Boole mixed the conditional probability up with the probability of a condition (if x then y, or in Boole notation: –x+y=0). He seemed not to have a clear view on this point (for further material see [Hailperin, chap 4]).

Hailperin [1986], who gives an entire reconstruction of Boole's thoughts, found that Boole had already formulated two fundamental problems that have been discovered several times in the last 50 years: They are

A Given a set $W = \{ w_1, w_2, \dots , w_m \}$ of wffs over the propositions $P = \{p_1, p_2, \dots , p_n \}$ and each wff w_i is true with probability π_i. Are these probabilities consistent?

B Given a set $W = \{ w_1, w_2, \dots , w_m \}$ of wffs again over P and each wff w_i is true with probability π_i. What is the consistent probability range of a wff w_{m+1} □ W?

Both problems also appear in the probability theory of Finetti [German edition 1981] and in Nilsson [1986]. Following the terminology of Hansen/Jaumard/al. [1995] we call problem A *probabilistic satisfiability* (PSAT). Problem B is also called *the probabilistic entailment* problem. The PSAT problem can be formulated as a linear system with an exponential number of variables. More precisely, let all 2^n interpretations be mapped into the variables X and let A be a $m x 2^n$–matrix such that $a_{ij} = 1$ if w_i is true in the *j*-th interpretation, then the probabilities in problem A are consistent if and only if the linear system

$$\begin{aligned} \mathbf{1} \cdot X &= 1 \\ A \cdot X &= \pi \qquad (1) \\ X &> 0 \end{aligned}$$

is feasible. The probabilistic entailment problem can be formulated with two LP's – a minimizing and a maximizing LP to find the lower and upper limit of the consistent probability range – defined as:

min $\pi_{m+1} = A_{m+1} \cdot X$ subject to (1) (for the lower bound) and

max $\pi_{m+1} = A_{m+1} \cdot X$ subject to (1) (for the upper bound).

Example 1, a PSAT problem: Given the following three wffs: $p, q, p \to q$ with probabilities 0.8, 0.3, and 0.6.Are the probabilities consistent? Hence, we have to see whether the following linear system is feasible:

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} \bullet X = \begin{pmatrix} 1 \\ 0.8 \\ 0.3 \\ 0.6 \end{pmatrix}$$

The system is infeasible, therefore, the assigned probabilities are inconsistent. Note that the corresponding SAT problem can be consistent, although the PSAT is infeasible. PSAT says nothing about the consistency of the wffs but only on the consistencies of the probabilities.

Example 2 a probabilistic entailment problem: Given the two wffs p and q with probabilities 0.8 and 0.6, what is the consistent probability range z for $p \rightarrow \bar{q}$? The two problems to solve are

$$\max(\min) \quad z = x_1 + x_2 + x_3$$

$$\text{subject to} \quad \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix} \bullet \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 0.8 \\ 0.6 \end{pmatrix}$$

We find: $0.4 \le z \le 0.6$.

The PSAT problem can be extended in several interesting ways [Hansen/Jaumard/al. 1995]:

– Instead of exact probabilities π one can use probability intervals $[\underline{\pi}, \overline{\pi}]$. The constraints $A \cdot X = \pi$ will then be replaced simply by $\underline{\pi} \le A \cdot X \le \overline{\pi}$.
– Conditional probabilities $p(w_i | w_j)$. Add two new rows (r_x and r_y), if they are no yet in the matrix A, one for the wff w_j, the other for $w_i \wedge w_j$. Add a new column (new variable y) to A where all entries are zero except in rows r_x and r_y: in (r_x the entry is –1, in r_y the entry is the given probability $p(w_i | w_j)$:

$$A_{(w_j)} \cdot X = y$$

$$A_{(w_i \wedge w_j)} \cdot X - p(w_i | w_j) \cdot y = 0$$

– Conditional probabilities in the objective functions can also be added. This produces the problem: $\min \left\{ \frac{cx}{ax} | Ax = b, x \ge 0 \right\}$. Using the Dinkelbach Lemma, this problem can be solved by iterating the solution to the problem $\min \left\{ cx - ax \frac{cx^0}{ax^0} | Ax = b, x \ge 0 \right\}$ until the expression $cx - ax\frac{cx^0}{ax^0}$ becomes negative. The solution converges. In practice this take about 10 iterations (personal communication from Jaumard/Hansen).

Another way to solve the problem $\min\left\{\frac{cx}{ax} | Ax = b, x \geq 0\right\}$ is to use fractional programming (Charnes/Cooper). One can solve instead the LP: $\min\{cx | Ax = tb, ax = 1, t \geq 0, x \geq 0\}$. The advantage is that there is only one LP to be solved, on the other hand, the disadvantage is that this homogenous system might be highly degenerative.

Example: It is easy to see that $\min p(w_i | w_j \vee w_k)$ is the same as $\min \frac{p(w_i)}{p(w_j) + p(w_k)}$.

Compare: ATMS and PSAT:

Suppose we have the following model:

$$prob(\alpha_1 \rightarrow b) = 1$$

$$prob(\alpha_2 \rightarrow \bar{b}) = 1$$

$$prob(\alpha_1) = p$$

$$prob(\alpha_2) = q$$

ATMS solves this model as following (note that independency is supposed):

$$\text{support}(b) = \alpha_1 \wedge \bar{\alpha}_2 \quad , \quad sp(b) = \frac{p(1-q)}{1-pq}$$

$$\text{support}(\bar{b}) = \bar{\alpha}_1 \wedge \alpha_2 \quad , \quad sp(\bar{b}) = \frac{(1-p)q}{1-pq}$$

PSAT gets the solution:

$$p \leq prob(b) \leq 1-q$$

PSAT gives sharper bounds than the ATMS approach since

$sp(b) \leq p$ & $1-q \leq sp(\bar{b})$ Why??? (Modeling ignorance!!!)

However note that the ATMS sopposes that the assumptions are indepentent. In the PSAT formulation this is not taken into account. To consider this in the PSAT, we need to add the following constraint:

$$prob(\alpha_1 \wedge \alpha_2) = pq.$$

But the lower and upper bounds ob b of the so modified PSAT model can only become smaller not wider!!!

(Another way to model the intention that the assumption are "indepented" (in any other sense), is to say that either the first or the second assumption must hold. This could be formulated in PSAT as:

$$prob(\alpha_1) = prob(1-\alpha_2) \quad \text{or} \quad prob(\alpha_2) = prob(1-\alpha_1).$$

or

$$prob(\alpha_1 \leftrightarrow \overline{\alpha}_2) = 1 \ )$$

The conditional probability is not identical to the probability of an implication. Suppose they were then we could do the following deduction:

$$p(a \to b) = p(b|a) \Leftrightarrow$$
$$p(\overline{a} \vee b) = \frac{p(a \wedge b)}{p(a)} \Leftrightarrow$$
$$p(a)\cdot[1-p(a \wedge \overline{b})] = p(a \wedge b) \Leftrightarrow$$
$$[\,p(a \wedge b) + p(a \wedge \overline{b})]\cdot[1-p(a \wedge \overline{b})] = p(a \wedge b) \Leftrightarrow$$
$$p(a \wedge b) + p(a \wedge \overline{b}) - p(a \wedge b)\cdot p(a \wedge \overline{b}) - [p(a \wedge \overline{b})]^2 = p(a \wedge b) \Leftrightarrow$$
$$p(a \wedge \overline{b}) + p(a \wedge b)\cdot p(a \wedge \overline{b}) + [p(a \wedge \overline{b})]^2 = 0 \Leftrightarrow$$
$$p(a \wedge \overline{b})\cdot[1-p(a \wedge b) - p(a \wedge \overline{b})] = 0 \Leftrightarrow$$
$$p(a \wedge \overline{b})\cdot[1-p(a)] = 0$$
$$[1-p(\overline{a} \vee b)]\cdot[1-p(a)] = 0$$

Thus, either $p(a) = 1$ or $p(a \to b) = 1$ (or both). It follows that the conditional probability is equal to the probability of a condition if and only if the latter is one or the event *a* is certain.

...not finished...

SYMBOLICAL APPROACHES IN LOGICAL MODELS

> "The Calculus of Probability should be applied to plausible reasoning, but only qualitatively." [Polya 1954a].

Probability theory has so well served

Numerical approaches and especially the Bayesian approach have been criticized .

Logic is a formal declarative language that gives us a powerful mean to reason in a deductive manner.

Its expressiveness is so convincing that it has lead many researcher to belief that it could be used for inductive deduction too. But it is difficult to adapt to take into account certain aspects of human reasoning. This observation led to a number of non-classical logics, which were proposed to describe various forms of inference. They were developed in manifold directions:

- The syntax is extended by introducing a) new operators such as “possible” or “necessary” (modal logic), different implication operators, fuzzy ∀- and ∃-operators such as “almost all”, “normally”, “some”; b) uncertain predicates or statements a certainty or uncertainty degree.
-

At the beginnings many ad-hoc rule-based systems (MYCIN etc.)

Statistical reasoning

Bayes' Theorem

Certainty factors and rule-based systems (MYCIN)

Bayesian Networks (conditional probabilistic networks (CPN) (Pearl: influence diagrams)

Probabilistic reasoning in intelligent systems

Dempster-Shafer Theory, Hint-theory (see SEV-Articles)

Fuzzy Logic

PSAT

Polya

...not finished...

EVIDENCE THEORY

> “The chances governing an aleatory experiment may or may not coincide with our degrees of belief about the outcome of the experiment. If we know the chances, then we will surely adopt them as our degrees of belief. … … Even after an immense number of observations, we may only have … a guess about the true chances based on speculative assumption.”
> Shafer [1976], p.16.

Shafer [1976] distinguishes between “chance” and “degree of belief”. Chance arise in the context of random (aleatory) experiments, like the throw of a die, and is independent of our knowledge – some probabilists would call this "objective probability". Chance could be interpreted as "frequency in the long run" (Mises). Degree of belief arises in an act of judgment and expresses our knowledge or beliefs about the would – some call this "subjective probability". We do not discuss here whether these terms are justified or not. Some have claimed that chance do not exist that it is only a feature of our knowledge. The first who made such a claim was Laplace. From his deterministic point of view randomness does not exist. This is equally erroneous than the contrary which is to belief that degree of belief ought always be like chances. That is that beliefs can be modeled using probability. (Bayesian, but Bayesian theory is a special case of the theory of evidence).

"Jeffrey H and Keynes (logical Bayesian view) insist that numerical degrees of support are indeed objectively determined by given evidence; Ramsey F. and de Finetti B (personalist Bayesian view) chose to analyze degrees of beliefs as psychological facts, facts that can be discovered by observing an individual's preferences among bets or risks but which may not bear any particular relation to any particular evidence." Shafer (not directly cited). Personalist view has gained predominance since Savage Foundations of Statistics.

The main short-comings of the Bayesian view is in the representation of the ignorance. Distinguish between lack of belief and disbelief.

Combination vs Conditioning (Carnap's unviable program).
Probable reasoning: eine weitere Schwäche der Bayesianer.

Shafer asserts that chance can be modeled with probability (in the sense of the three axioms), but not so degree of beliefs. Degree of beliefs can be modeled by a belief function *bel*: $2^{\Theta}\varnothing[0,1]$ which is defined as (where Θ is a finite set):

(1) $bel(\varnothing) = 0$

(2) $bel(\Theta) = 1$

(3) if $A_{i(1,n)} \subseteq \Theta$ then $bel(\bigcup_i A_i) \geq \sum_{i_1 \ldots i_m \subseteq 1 \ldots n} (-1)^{m-1} bel(\bigcap_{j=1}^{i_m} A_j)$

The additivity of probabilities does not hold for the belief function. Shafer gives a plausible small example: Suppose you contemplate a vase

We need a theory of arguments: the beginnings have been developed by Bernoulli [Shafer 1978].

Interpret "**degree of beliefs**" as probability [Gottinger, p 87]: Suppose you are faced with the following decision problem: You have to choose between two lottery tickets:

(1) receive one million dollars if it rains tomorrow

(2) receive one million dollar if a black ball is drawn from the urn of 100 balls.

How would you decide? Of course, if there were no black balls in the urn, you would choose (1), on the other hand if all balls in the urn were black, you would choose (2). Consider now an intermediate case, were we add a growing number of black balls (by removing non-blacks). When the number is small, you may still choose ticket (1), but you will come to a point where you will switch your choose (say at 24 black balls). At this point, you are indifferent to the choice. Now 24/100 could be interpreted as your "degree of belief" that it will rain tomorrow.

Another closely related concept is *ignorance* amalgamated with the concept of uncertainty.

ARGUMENTATIVE SYSTEMS

"goal programming" in logical systems......

Database

r = it rains
s = sprinkler is on
w = grass is wet
o = shoes are wet

deterministic database
r
s
r ∅ w
s ∅ w

w
w ∅ o

PSAT database:

p(r) = 0.6
p(s) = 0.1
p(r ∅ w) = 0.8
p(s ∅ w) = 0.2
p(w) = 0.7
p(w ∅ o) = 0.9

p(rswo)	r	s	w·r	w	w·s	w·o
p(1111)	≤ 0.6	≤ 0.1			**≤ 0.02**	
p(1110)	≤ 0.6	≤ 0.1			**≤ 0.02**	
p(1101)	≤ 0.6	**≤ 0.1**				
p(1100)	≤ 0.6	**≤ 0.1**				
p(1011)	≤ 0.6		**≤ 0.48**			
p(1010)	≤ 0.6		≤ 0.48			**≤ 0.37**
p(1001)	≤ 0.6			**≤ 0.3**		
p(1000)	≤ 0.6			**≤ 0.3**		
p(0111)	≤ 0.4	≤ 0.1			**≤ 0.02**	
p(0110)	≤ 0.4	≤ 0.1			**≤ 0.02**	
p(0101)	≤ 0.4	**≤ 0.1**				
p(0100)	≤ 0.4	**≤ 0.1**				
p(0011)	**≤ 0.4**					
p(0010)	≤ 0.4					**≤ 0.37**
p(0001)	≤ 0.4			**≤ 0.3**		
p(0000)	≤ 0.4			**≤ 0.3**		

PSAT database:

p(r) = 0.6
p(s) = 0.1

p(w | r) = 0.8

p(w | □r) = ?

p(w | s) = 0.2

p(w | □s) = ?

p(w) = 0.7

p(o | w) = 0.9

p(o | □w) = ?

Argumentative database

a_1 ∅ r

a_2 ∅ s

a_3 ∅ (r ∅ w)

a_4 ∅ (s ∅ w)

a_5 ∅ w

a_6 ∅ (w ∅ o)

(Analogon: do make an in feasible system $Ax \leq b$ feasible, introduce slacks such that $Ax + sp - sn = b$.)

REFERENCES

DARWICHE A.Y., [1993], Argument Calculus and Networks, in: Proceedings of the Ninth Conference of Uncertainty in Artificial Intelligence, 1993, eds.: Heckerman D., Mamdani A.

FOX J., KRAUSE P., ELVANG–GØRANSSON M., [1993], Argumentation as a General Framework for Uncertain Reasoning, in: Proceedings of the Ninth Conference of Uncertainty in Artificial Intelligence, 1993, eds.: Heckerman D., Mamdani A.

ELVANG–GØRANSSON M., KRAUSE P., FOX J., [1993], Acceptability of arguments as 'logical uncertainty', in: Symbolic and Quantitative Approaches to Reasoning and Uncertainty, Proceedings of the European Conference ECSQARU'93, Granada, Spain, November 1993, Springer.

NON-MONOTONIC LOGIC

Classical logic obeys the monotonic property, which means the following: Let p, q and Q be three wffs, then if p is derivable from Q then it is also derivable from $Q \wedge q$, in other words $(Q \rightarrow p) \rightarrow (Q \wedge q \rightarrow p)$ is a tautology.

Commonsense reasoning does normally not follow the monotonic property. This can be shown by a simple example. Suppose the model consists of the following three statements:

(1) All birds can fly.

(2) All penguins are birds.

(3) Tweety is a penguin.

Of course, we can derive that Tweety – by the way, the most famous animal in AI circles – can fly.

Now suppose that later we make the discovery that

(4) No penguin can fly.

Can we derive now that Tweety can fly? Yes by the same deduction as before. But we can also derive that Tweety cannot fly, which produces a contradiction! The model becomes inconsistent. New evidence lead to a revision of already correctly derivable statements, this is clearly a violation of the monotonic property.

How to manage such a situation? A radical step would be to completely modify the model. For example, to modify axiom (1), in our example, to "All birds except penguins can fly"[3]. But this does not really solve our problem; maybe Tweety is not a penguin but an ostrich. Now the statement (1) must be modified again to "All birds except penguins and ostriches can fly". Each time an exception is found it would have been added to the general axiom; it is impossible to add all exceptions. Furthermore, we do not want to modify (1), because we feel that it expresses a important law: even if penguins cannot fly they have still many other characteristics in common with all kind of birds. For the same reason we do not want to modify (2).

---

[3] It is interesting to note that PROLOG's backtracking mechanism is non-monotonic due to its processing of the negation. If our Tweety-example is coded by the following three predicates

```
Fly(X) :- Bird(X) , not IsExceptionalBird(X).
IsExceptionalBird(X) :- Penguin(X).
Bird(tweety).
```

then *IsExceptionalBird(tweety)* cannot be proven by PROLOG. Therefore, PROLOG derives *not IsExceptionalBird(X)* and together with *Bird(tweety)* it derives *Fly(tweety)*. If, on the other hand, we add the predicate

```
Penguin(tweety).
```

then *Fly(tweety)* can no longer by derived since now *IsExeptionalBird(tweety)* is provable [see Brewka, p. 3].

Another solution would be to replace (1) by a somewhat weaker statement such as “Almost all birds can fly”, or “Typically birds can fly”. This leave us with the problem of how to interpret such a statement. Many extensions (default logic [Reiter, 1980], autoepistemic logic [McDermott/Doyle 1980], Circumscription [McCarthy, 1980]) to the classical logic have been proposed to interpret such a statement (for an introduction see [Brewka, 1991]).

Yager [1987] has proposed to translate the logical model into a 0-1 model and to model default rules using similar techniques as in goal programming.

Probably, the unique consistent framework for non-monotonic statements such as *Normally(α∅β)* is to formulate them as probabilistic statements within the PSAT problem as *prob(α∅β)* = 0.9, for instance. At least it *is* a possibility.

FUZZY SET THEORY

In the probability paradigm and in the paradigm of logic, a number of well defined events or propositions are given. But it is in general not known exactly which events are the case, or if known only with a certain probability.

The theory of fuzzy set, on the other side, deals with propositions which have *vague meaning*. Some events or propositions are not well defined or *fuzzy*. A vague description of an object or a process seems to be easier to remember by human and are apparently more economical and hence more efficient. The reality itself might not be fuzzy, but our knowledge and the contents of our theories about reality may be fuzzy. Fuzziness is vagueness, which makes it a central element in human thoughts and perceptions, as well as in human language. “The theory of fuzzy sets enables us to structure and describe activities and observations which differ from each other only vaguely, to formulate them in models and to use these models for various purposes – such as problem-solving and decision-making” [Zimmermann/Zadeh/Gaines 1984, p 18].

On the other side, as systems become more complex, it becomes increasingly difficult to make analytical (Zimmermann uses mathematical??) statements about them which are both meaningful and precise. In complex problem situation in which human beings are involved, much of the knowledge about the situation is expressible in linguistic terms only [Zimmermann/Zadeh/Gaines 1984, p 11].

Sometimes we use fuzzy concepts and words also to describe the object of our world, because the boundary of an object is uncertain, unclear, or unreliable (e.g. “all trees in Canada”). The natural language is full of words with vague or fuzzy meaning. There is, for example, no such thing as the boundary of the Swiss Alps; any such boundary would be artificial. That does not mean that one cannot fix a boundary for a special purpose and work with that “precise definition”. But this is not always entirely satisfactory. An illustrative example is the ancient Sorites[4] paradox: If there is a heap of sand and a single grain of sand is taken away, then one could say that the remaining grains still form a heap of sand. But by induction one could then take away all the sand and still have a heap of sand – an absurdity. The question then arises how many grain of sand a heap must contain such that it could still be called “heap”. This is a question that cannot be answered in an entirely satisfactory manner. The fact is that human beings are doing very well by using the notion of “a heap of sand” without ever knowing and defining of how many grain they mean.

Almost every sentence in a natural language contains fuzzy concepts. Some simple examples are:

- “Peter is *about* 40 years old”, “Paul owns *few* money”.
- “Mary is *tall*”. “If the growth is *quite large*, then there is a good chance the tumor is cancerous”.

In the first two examples, the exact age of Peter or the amount of money Paul owns, are <u>uncertain</u>. In second kind of examples, the exact meaning of “tall” is <u>unclear</u>, but most would agree on a range between 1.75m–2.30m with a most likely value between 1.80m and 1.90m.

But unclear meaning and uncertainty are to a certain extent only the two sides of the same coin. To say that “Mary is tall” could express our uncertainty of Mary's height; and “few money” could just express our unwillingness to pronounce the exact amount, even if we know it, or it would be pointless to give a more precise meaning. Hence, one could use probabilities to describe vague meaning. But this is not entirely appropriate, because we do not think of *tall* as a probability distribution of height. Rather we think of all possible heights as more or less appropriate members of the set of all tall persons. Uncertainty about the statement such as “Mary is tall” is not represented by the *probability* of Mary being tall, but rather by the *possibility* of Mary being tall, that is, by a membership function of all tall persons $\mu_{tall}(x)$ that maps all

---

4 sorites: from the Greek word σωροσ which simply means “heap”.

heights x to a number in the range [0,1], such as:

$$\{\mu_{tall}(1.6)=0, \mu_{tall}(1.7)=0.5, \mu_{tall}(1.8)=1.0, \mu_{tall}(1.9)=1\}$$

[or more general: $\mu_{tall}(x)=\max(1,\min(0,5x-8))$].

The same could be done for other vague term such as *small* or *medium*:

$$\mu_{small}(x)=\max(1,\min(0,-5x+9))$$

$$\mu_{medium}(x)=\min(0,1-|10x-17|)$$

The graph of the three membership functions is shown in Figure 2

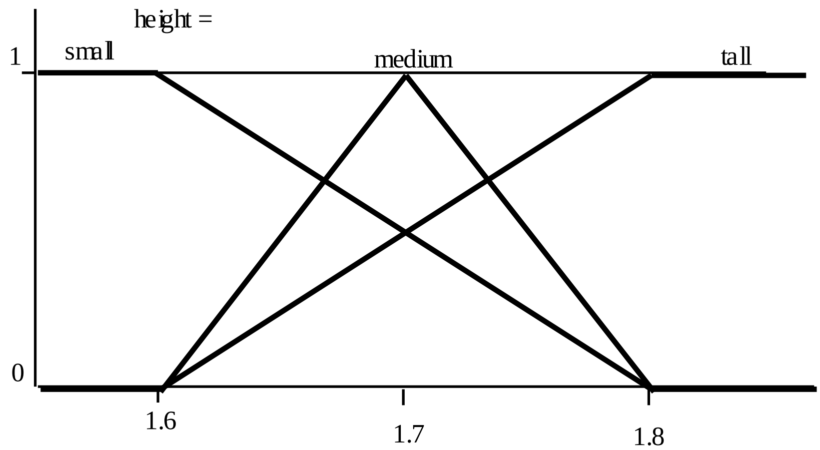


Figure 2: fuzzy sets for *small*, *medium*, and *tall*

Small, *medium* and *tall* are three fuzzy sets belonging to the fuzzy variable *height*. One can say that height may have three fuzzy values: *small*, *medium* or *tall*. To make things more explicit let's take an example.

MODELING DYNAMIC SYSTEMS

Fuzzy sets and fuzzy variables have been used to model dynamic real-time systems. Modeling dynamic systems normally involves a differential equation system. Often this is so complex that it cannot be solved sometimes even not modeled in an analytical manner. Even if the differential system could be solved, it might sometimes be advantageous to use a fuzzy set approach. Hence, there are two reasons to use a fuzzy set approach to model dynamic systems: precision is related to complexity, therefore analytical approaches such as differential systems are unworkable. The other reason is that systems are normally very robust with an fuzzy set approach. But there is also an important disadvantage. The parameter tuning is critical and somewhat arbitrarily: the fuzzy system parameters are manually modified and adjusted until a reasonable behaviour results for the concrete problem at hand without sometimes having the slightest idea why it works. Nevertheless, the fuzzy set approach is very powerful to model complex behaviour, especially if there are no analytical approaches available.

To illustrate the fuzzy set approach, we use the problem of the inverted pendulum. The problem is easy enough to be formulated and solved as an differential system [Yamakawa 1989], but sufficiently interesting to use a fuzzy set approach as well [Togai 1991]. A pole is fixed on a car, it can fall due to gravity in both directions in which the car can move. The object is to move the car forward and backward such that the pole keeps its balance, that is it maintains its vertical position (Figure 3).

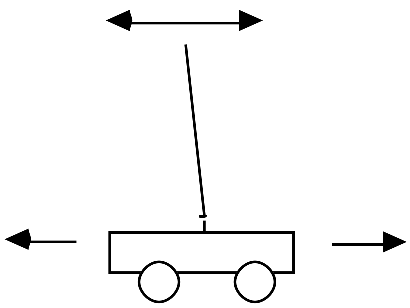


Figure 3: the inverted pendulum

There are different techniques to formulate the problem using the fuzzy set approach. The easiest I have found and which works quite well (I have tested it) is used in Togai 1991. They use three fuzzy variables: *Theta*, the angle of the pole deviating from the vertical line; *dTheta* the velocity of the pole; and *Velocity* the velocity of the car. Several fuzzy sets are assigned to the variables. *Theta*, for example, can be "negative", "small negative", "about zero", "small positive", or "positive". "Negative", for example, is defined by the membership function μ as following:

$$\{\mu_{negative}(x \le -30) = 1, \mu_{negative}(x \in]-30,-10[) = -\tfrac{x}{10} - 2, \mu_{negative}(x \ge -10) = 1\}$$

All other sets are defined in the same fashion. When the variables and the fuzzy sets are defined, one can uses then to model the *rules* of the models. A rule is a instruction such as:

> If the angle of the pole is slidely negative and the velocity of the pole is slidely positive then the velocity of the car should be slowed down to be about zero.

Experiences show that seven rules are sufficient to formulate the model. In a ad hoc modeling language, one might formulate the model as following:

```
 (* model: inverted pendulum : pseudo code in LPL [Hürlimann 1994]*)
VAR Theta FUZZY
    ( NM: -50 1, -30 1, -20 0;   (* negative *)
      NS: -30 0, -15 1,   0 0;   (* small negative *)
      ZO: -10 0,   0 1,  10 0;   (* about zero *)
      PS:   0 0,  15 1,  30 0;   (* small positive *)
```

```
      PM:  20 0,  30 1,  50 1 ); (* positive *)

VAR dTheta FUZZY
    ( NS: -20 1, -10 1,  0 0;    (* small negative *)
      ZO: -10 0,   0 1, 10 0;    (* about zero *)
      PS:   0 0,  10 1, 20 1 );  (* small positive *)

VAR Velocity FUZZY
    ( NM: -50 1, -20 1, -10 0;   (* negative *)
      NS: -20 0, -10 1,   0 0;   (* small negative *)
      ZO: -10 0,   0 1,  10 0;   (* about zero *)
      PS:   0 0,  10 1,  20 0;   (* small positive *)
      PM:  10 0,  20 1,  50 1 ); (* positive *)

MODEL
  Rule1: IF Theta=PM AND dTheta=ZO THEN Velocity=PM;
  Rule2: IF Theta=PS AND dTheta=PS THEN Velocity=PS;
  Rule3: IF Theta=PS AND dTheta=NS THEN Velocity=ZO;
  Rule4: IF Theta=NM AND dTheta=ZO THEN Velocity=NM;
  Rule5: IF Theta=NS AND dTheta=NS THEN Velocity=NS;
  Rule6: IF Theta=NS AND dTheta=PS THEN Velocity=ZO;
  Rule7: IF Theta=ZO AND dTheta=ZO THEN Velocity=ZO;
END
```

L.P.L.[5] was one of the first fuzzy modeling language that allowed to formulated dynamic systems in a similar manner. Togai InfraLogic [1991] have developed a system that translates a model similar to the syntax above into C. It would be somewhat technical to explain in this paper, how this is done. Essentially, using the seven rules the translator produces a C-function

```
float VelocityOfTheCar(FuzzySet Theta, FuzzySet dTheta);
```

which returns the exact value the car must get to balance the pole at a specified moment. The theory can be found in Zimmermann [1991 or 1987].
It is – at least to me – perplexing that such a simple model specification can solve the inverted pendulum problem and it shows that the fuzzy set approach can be powerful to model dynamic systems. It shows also that fuzzy sets theory has never been an invitation to fuzzy thinking!

Let's take another example of a complex control process: Rotary kilns are use in a number of material processing operations – such as producing cement – to raise the temperature of the solid material to a specified level to make sure that certain chemical processes take place. The kiln axis is tilted and rotates slowly. The material enters at the high end and tumble down to the exit. The gas enters at this point and a long combustion flame distributes the heat along the length of the kiln. A manual (human) operator can control the heat process in several ways by open and closing several valves.

---

[5] The fuzzy language L.P.L. [Adamo 1980] is not related to the modeling language LPL [Hürlimann 1993], but – maybe in the distant future – L.P.L. will be a subset of LPL.

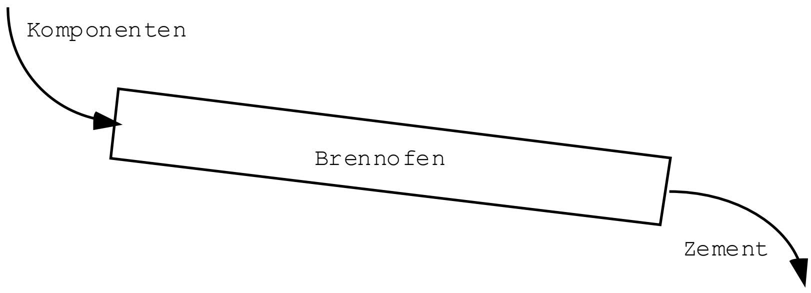


Figure ???

The operators model to decide what to do consists in general of a set of instructions as following:

IF the gas at the exit is blue THEN open slightly valve 3.

About 90 such rules have been identified. They have been translated into fuzzy rules as described above and feeded successfully to an automatic operator.

In the same sense, fuzzy set theory could be used to model fuzzy syllogisms that are used in commonsense reasoning. From the two statements:

Socrates is very healthy and

Healthy men live a long time.

a human being would automatically infer that

Socrates will live a very long time.

Such a deduction is not possible in classical logic.

FUZZY LP'S

Uncertainty in a LP $\{\max cx \mid Ax \leq b, x \geq 0\}$ occurs in different ways: the data *A*, *b*, and *c* are not known exactly, the ≤-operator is fuzzy, or several objectives compete with each other. For all three problems, the fuzzy set theory has been successfully used in the past [Zimmermann 1991, Rommelfanger 1993, Czyzak/Slowinski 1989]. The easiest problem is when the ≤-operator is fuzzy ($Ax \tilde{\leq} b$). Each row *i* can be represented by a fuzzy set in the following way:

$$\mu_i(x) = \begin{cases} 1 & \text{if} \quad A_i x \leq b_i \\ 1 - \frac{A_i x - b_i}{p_i} & \text{if} \quad b_i < A_i x \leq b_i + p_i \\ 0 & \text{if} \quad A_i x > b_i + p_i \end{cases} \quad \text{[Zimmermann 1991, p. 251]}$$

where p_i is a maximal violation parameter for the *i*-th row. Translating the objective function *cx* into an upper bound constraint of the form $cx \tilde{\geq} z$ where *z* is an aspiration level for the value of the objective function, it is easy to translate the fuzzy LP into the crisp LP which contains just one more variable:

$$\left\{\max\ \lambda \middle| \lambda p_0 - cx \le -z + p_0\ ,\ \lambda p + Ax \le b + p \right\}$$

The multi-objective problem can be approach by a similar procedure.
We conclude that the fuzzy set approach can be a real alternative to stochastic programming. The main advantage is that the former produces a crisp LP that is as easy to solve than the original model.

It is easy to integrate fuzzy concepts into a mathematical modeling language. Like in stochastic programming the datatype STOCHASTIC, one could add the datatype FUZZY to indicate that certain data are uncertain. In the case where the right-hand-side vector *b*, for example, is defined as the fuzzy set $\{\mu(100) = 0.0, \mu(150) = 0.5, \mu(220) = 1.0\}$, one could formulate the general LP structure in an extended version of LPL as following:

```
SET i; j;
COEF
  c{j}; A{i,j};
  b{i} FUZZY ((100 0.0) (150 0.5) (220 1.0));
VAR x{j};
MODEL
  R{i} : SUM{j} A*x = b;
MINIMIZE
  obj: SUM{j} c*x + slack(R);
END
```

REFERENCES

DAVIS E., [1990], Representations of Commonsense Knowledge, Morgan Kaufmann Publ., San Mateo, California.

DELVIN K., [1994], Mathematics, The Science of Pattern, Scientific American Library, NewYork.

POPPER K., [1978], An intellectual Autobiography, Unended Quest, Fontana/Collins.

Van HENTENRYCK P., [1989], Constraint satisfaction in logic programming, Logic Programming Series, MIT Press.

LP and Goal Programming

BEILBY M.H., MOTT T.H. [1983], Academic Library Acquisitions, Allocation based on Multiple Collection Development Goals, in: Comput. & Ops. Res. Vol. 10, No. 4, pp 335–343.

CHINNECK J., DRAVNIEKS E, [1991], Locating Minimal Infeasible Constraint Sets in Linear Programs, in: ORSA Journal on Computing, Vol 3, Nr 2, pp 157–168.

IGNIZIO J.P., [1976], Goal Programming and Extensions, Lexington Bokks, D.C. Health and Comp., Toronto.

HÄTTENSCHWILER P., [1988], Goal Programming becomes most useful using L_1–smoothing functions, in: Computational Statistics & Data Analysis 6, pp. 369–383.

HÜRLIMANN T., [1994], IP, MIP and Logical Modeling using LPL, Working Paper No 205, Institute of Informatics, University of Fribourg, updated November 1994.

HÜRLIMANN T. [1993], LPL: A mathematical programming language, in: OR Spektrum, Vol 15, pp 43–55, Springer.

SCHNIEDERJANS M.J., [1995], Goal Programming, Metholology and Applications, Kluwer Acad. Publ., Boston.

TAMIZ M., MARDLE S.J., JONES D.F., [1995], Detecting IIS in Infeasible Linear Programmes using Techniques from Goal Programming, Talk presented at the APMOD 95 conference, 3–5 April 1995, at the Brunel University, West London

ZIONTS S. [1988], Multiple Criteria Mathematical Programming: An Updated Overview and Several Approaches, in: Mathematical Models for Decision Support, Mitra G. (ed.), NATO ASI Series F, Vol 48, Springer Verlag.

Stochastic Optimization

BIRGE J.R., WETS R. J-B., (eds), [1991], Stochastic Programming – Part I, Annals of Operations Research (ed-in-chief: Hammer P.L.), Vol 30, J.C. Baltzer AG, Basel.

BIRGE J.R., WETS R. J-B., (eds), [1991], Stochastic Programming – Part II, Annals of Operations Research (ed-in-chief: Hammer P.L.), Vol 31, J.C. Baltzer AG, Basel.

ERMOLIEV Y., WETS R. J-B., [1988], Numerical Techniques for Stochastic Optimization, Springer, Berlin.

HEYMAN D.P., SOBEL M.J., [1984], Stochastic Models in Operations Research, Vol II, Stochastic Optimization, McGraw-Hill Book Company, New York.

INFANGER G., [1992], Planning under Uncertainty, Solving Large-Scale Stochastic Linear Programs, Habilitationsschrift, Department of Operations Research, Standford University.

KALL P., [1976], Stochastic Linear Programming, Springer, Berlin.

KALL P., [1992], A Model Management System for Stochastic Linear Programming, in: Kall P. (ed), System Modelling and Optimization, Springer, 1992, pp 580–587.

KALL P., MAYER J., [1993?], SLP-IOR: An Interactive Model Management System for Stochastic Linear Programs, IOR, University of Zurich, Switzerland, (submitted to Math Programming Ser. B).

KALL P., WALLACE S.W., [1994], Stochastic Programming, John Wiley & Sons, Chichster.

KING A.J., [1988], Stochastic Programming Problems: Examples from the Literature, in: Ermoliev/Wets, pp 543–567.

REMBOLD J.Th., [1977], Stochastische lineare Optimierung, Eine anwendungsbezogene systematische Darstellung, Verlag Anton Hain, Meisenheim am Glan.

SENGUPTA J.K., [1972], Stochastic Programming, Methods and Applications, North-Holland, Amsterdam.

WETS R.J-B., [1991], Stochastic Programming, in: Handbooks in Operations Research and Management Science, Vol 1, Optimization (eds. Nemhauser G.L., Rinnooy Kan A.H.G., Todd M.J., North-Holland, Amsterdam.

PSAT, Probabilistic Logic

ANDERSEN K.A., HOOKER J.N., [1994], Bayesian Logic, in: Decision Support Systems, Vol. 11, No. 2, pp. 191–210.

DOUANYA NGUETSE G-B.., HANSEN P., JAUMARD B., [1994], Probabilistic Satisfiability and Decomposition, Working Paper no 94-26, December, Institute of Informatics, University of Fribourg (Switzerland).

HAILPERIN T. [1986], Boole's Logic and Probability, 2nd ed., North-Holland, Amsterdam.

HANSEN P., JAUMARD B., DOUANYA NGUETSE G-B-., de ARAGÃO P., [1995], Models and Algorithms for Probabilistic and Bayesian Logic, Working Paper no 95-01, January, Institute of Informatics, University of Fribourg (Switzerland).

NILSSON N.J., [1986], Probabilistic Logic, Artificial Intelligence, 28 (1986), p.71-87.

Statistics, Bayesian Statistics, Bayesian Networks

de FINETTI B., [1981], Wahrscheinlichkeitstheorie, Oldenbourg, München.

GOTTINGER H.W., [1974], Bayesian Analysis Probability and Decision. Vandenhoeck & Ruprecht, Göttingen.

LAURITZEN S.L., SPIEGELHALTER D.J., [1988], Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems, in: Journal of the Royal Statistical Society, Ser. B, Vol. 50, No. 2, p 157–224.

LI Z., D'AMBROSIO B., [1994], Efficient Inference in Bayes Networks as a Combinatorial Optimization Problem, in: International Journal of Approximate Reasoning, Vol 11, no 1, July, p 55–81.

LU P., [1992], Formal Desciption of Uncertainty in Decision Support and Information Systems, Dissertation at the Technische Universität Wien, Technisch-Naturwissenschaftliche Fakultät, Wien.

NEAPOLITAN R.E., [1990], Probabilistic Reasoning in Expert Systems: Theory and Algorithms, Wiley, Ney York.

PEARL J., [1988], Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, San Mateo, CA.

WALLEY P., [1991], Statistical Reasoning with Imprecise Probabilities, Chapman and Hall, London.

WICKMANN D., [1990], Bayes-Statistik, Einsicht gewinnen und entscheiden bei Unsicherheit, Math. Texte, Bd. 4, Hrsg.: Knoche N. & Scheid H., BI Wissenschaftsverlag, Mannheim.

Theory of Evidence, Theory of Hints, Argument Systems

KOHLAS J., [1987], The Logic of Uncertainty, Potential and Limits of Probability Theory for Managing Uncertainty in Expert Systems, Working Paper No 142, Institute of Informatics, University of Fribourg, Switzerland.

KOHLAS J., MONNEY P.-A., [1994], Theory of Evidence – A Survey of its Mathematical Foundations, Applications and Computational Aspects, in: ZOR – Mathematical Methods of Operational Research, (1994) Vol 39, p35–68.

KOHLAS J., MONNEY J-P., [1995], A Mathematical Theory of Hints, An Approach to the Dempster-Shafer Theory of Evidence, Springer, Berlin.

Logic – Non-monotonic Logic

BACCHUS F., [1990], Representing and Reasoning with Probabilistic Knowledge, A Locical Approach to Probabilities, The MIT Press, Cambridge, Mass.

BREWKA G., [1991], Nonmonotonic Reasoning: Foundations of Commonsense, Cambridge Tracts in Theoretical Science 12, Cambridge University Press.

McDERMOTT D., DOYLE J., [1980], Non-Monotonic Logic I, in: Artificial Intelligence 13, p 41–72.

POLYA G., [1954], Mathematics and Plausible Reasoning, Volume I: Induction and Analogy in Mathematics, Princeton University Press, New Jersey, 12th printing, 1990.

POLYA G., [1954a], Mathematics and Plausible Reasoning, Volume II: Patterns of Plausible Inference, Princeton University Press, New Jersey, 12th printing, 1990.

McCARTHY J., [1980], Circumscription – A Form of Non-Monotonic Reasoning, in: Artificial Intellicene 13, p 27–39.

REITER R., [1980], A Logic for Default Reasoning, in: Artificial Intellicene 13, p 81–132.

YAGER R.R. [1987], A mathematical programming approach to inference with the capability of implementing default rules, in: Gaines B.R., Boose J.H. (eds), Machine Learning and Uncertain Reasoning, Knowledge-Based Systems Vol 3, Academic Press, London, 1990, pp.261–290.

Fuzzy Logic and Fuzzy Set Theory

ADAMO J.M., [1980], L.P.L. – A Fuzzy Programming Language, 1. Sytactic Aspects, 2. Semantic Aspects, in: Fuzzy Sets and Systems Vol 3, pp 151–179 and pp 261–289.

ALTROCK C., [1991], Über den Daumen gepeilt, Fuzzy Logic: scharfe Theorie der unscharfen Mengen, in: c't Magazin, Heft 3.

BANDEMER H., GOTTWALD S. [1990], Einführung in FUZZY-Methoden, Theorie und Anwendungen unscharfer Mengen, Verlag Harri Deutsch, Thun, Frankfurt/Main.

CZYZAK P., SLOWINSKI R., [1989], Multiobjective Diet Optimization Problem under Fuzziness, in: The Interface between Artificial Intelligence and OR in Fuzzy Environment, Verdegay/Delgado (eds.), TÜV Rheinland, Köln.

GOTTWALD S., [1989], Mehrwertige Logik, Eine Einführung in Theorie und Anwendungen, Akademie Verlag, Berlin.

GUPTA M.M., YAMAKAWA T. (eds), [1988], Fuzzic Logic in Knowledge-based Systems, Decision and Control, North-Holland, Amsterdam.

HELLENDOORN H., [1990], Reasoning with Fuzzy Logic, Proefschrift Technische Universiteit Delft (The Netherlands), CIP-Gegevens Koninklijke Bibliotheek, Den Haag.

MATTILA J.K., [1989], Calculus of Modifier Operators in Fuzzy Logic, University of Turku, Institute for Applied Mathematics, Turku, Finland.

NEGOITA C.V., [1981], Fuzzy Systems, Abacus Press.

NEGOITA C.V., [1985], Expert Systems and Fuzzy Systems, The Benjamin/ Cummings Publ. Comp. Calofornia.

NOVAK V., [1989], Fuzzy Sets and their Applications, Adam Hilger, Bristol.

PEDRYCZ W., [1989], Fuzzy Control and Fuzzy Systems, Research Studies Press Ldt. & John Wiley, Taunton.

ROMMELFANGER H., [1993], Fuzzy Decision Support-Systeme, Entscheiden bei Unschärfe, (2. Auflage), Springer-Lehrbuch, Berlin.

PRADE H., NEGOITA C.V., [1986], Fuzzy logic in knowledge engineering, Verlag TÜV Rheinland, Köln.

SCHMUCKER K.J., [1984], Fuzzy Sets, Natural Language Computations, and Risk Analysis, Computer Science Press, Rockville.

SUGENO M, NISHIDA M., [1985], Fuzzy Control of Model Car, in: Fuzzy Sets and Systems Vol 16, pp 103–113.

TOGAI InfraLogic Inc., [1991], Fuzzy-C Development System, User's Manual, 30 Corporate Park, Suite 107, Irvine, CA 92714.

YAMAKAWA T., [1989], Stabilization of an inverted Pendulum by a high-speed fuzzy Logic Controller Hardware System, in: Fuzzy Sets and Systems Vol 32, p.161ff.

ZADEH L.A., [1965], Fuzzy Sets, in: Information and Control Vol 8, pp. 338–353.

ZADEH L.A., [1968], Fuzzy Algorithms, in: Information and Control Vol 12, pp 94–102.

ZIMMERMANN H.J., [1976], Description and Optimization of Fuzzy Systems, in: Int. J. General Systems, Vol. 2, pp.209–215.

ZIMMERMANN H.J., ZADEH L.A., GAINES B.R. (eds.) [1984], Fuzzy Sets and Decision Analysis, in: Studies in the Management Sciences, Vol 20, North Holland, Amsterdam.

ZIMMERMANN H.J., [1987], Fuzzy Sets, Decision Making, and Expert Systems, Kluwer Acad. Press, Boston.

ZIMMERMANN H.J., [1991], Fuzzy Set Theory and its Applications, Second Edition, Kluwer Academic Publ., Boston.