A COMPARISON OF PROBABILISTIC AND FUZZY INFERENCE FOR EXPERT SYSTEMS

by

Garth Stuart Akin
B.A., San Diego State College, 1964

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF ARTS

in the Departments of Computer Science and Education

We accept this thesis as conforming
to the required standard

[redacted]
Dr. Jon C. Muzio, Co-Supervisor (Computer Science)

[redacted]
Dr. C. Brian Harvey, Co-Supervisor (Psychological Foundations)

[redacted]
Dr. Nikitas J. Dimopoulos, Outside Member (Electrical and Computer Engineering)

[redacted]
Dr. Walter Muir, Dept. Member (Psychological Foundations)

[redacted]
Dr. Nick Cercone, External Examiner (Computer Science, Simon Fraser University)

ISBN 0-315-62625-9

ABSTRACT

Investigators differ over approaches to managing inexact inference for expert systems, particularly concerning probabilistic and fuzzy inference. Analysis of the literature suggests this is due partly to glossing distinctions between types of inexactness, and partly to lack of a commonly accepted theoretical framework.

Probabilistic and fuzzy inference are compared by presenting a formal language, and interpreting it in a metric lattice. It is shown that if the lattice is modular, the truth-value assignments for Standard Uncertainty Logic (Gaines) can be derived; if the lattice is Boolean the value-assignments for Unconditional Probability Logic (Rescher) can be derived; if the lattice is a chain, the value assignments for a Fuzzy Logic (Zadeh) can be derived; if the lattice is a Boolean chain the value-assignments for Sentential Calculus can be derived.

It is shown how the set of models can be implemented as inference procedures for an expert system shell. A comparison of the probabilistic and fuzzy inference procedures shows that their results are practically the same.

Examiners:

[redacted]

Dr. Jon C. Muzio, Co-Supervisor
(Computer Science)

[redacted]

Dr. C. Brian Harvey, Co-Supervisor
(Psychological Foundations)

Examiners (continued):

[redacted]

Dr. Nikitas J. Dimopoulos, Outside Member (Electrical and Computer Engineering)

[redacted]

Dr. Nick Cercone, External Examiner (Computer Science, Simon Fraser University)

[redacted]

(Psychological Foundations)

Dr Maarten van Emden was a member of both the supervisory committee and the examining committee, but, owing to disagreement on the content of the thesis, and the nature of the subsequent examination, has declined to sign this document.

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

This section contains an overview of the present study. The sections and subsections are itemized and their contents are described.

1 NOTATIONS, DEFINITIONS, EXPLANATIONS

Notation and definitions are established, and the concepts treated in the sequel are informally explained.

1.1 An inference procedure for expert system applications

A simplified version of the Autometrics VRP "Versatile Rule Processor" inference procedure is explained, as an example of one type of rule-based expert system software.

1.2 Logic and Sentential Calculus.

Distinctions between syntax rules, semantic rules, conditional sentences, and inference rules are explained; the Sentential Calculus is defined and its correspondence with VRP rulebases and inference procedure is noted.

Proof-theoretic (axiomatic) and Model-theoretic (truth-table) approaches to defining a logic are distinguished, and the latter approach is selected for this study as more suited for VRP-type expert systems.

The traditional distinction between Inductive and Deductive inference is deemed not to be germain to this study, wherein "inference" means the computation of conclusion values from the values of antecedents and conditionals.

1.3 Sets and Probability Theory

Set theory and Probability theory are defined, and differences between (Bayesian) Conditional Probability and Unconditional Probability Logic are explained.

1.4 Fuzzy sets and Fuzzy Logic.

Fuzzy set theory is presented and compared with standard Set theory. Fuzzy logic is derived from Fuzzy set theory, and informally compared with Probabilty Theory.

1.5 Lattices and Uncertainty Logic.

Lattices (modular, Boolean, and chain) are defined and explained, and correspondences between Standard Uncertainty Logic and modular lattices are presented.

1.6 Summary

(Each system of logic is defined by truth-value assignments, and a generalization of the inference rule Modus Ponens is derived for each of sections 3 - 5. These inference rules are represented as implication operators)

2 APPROACHES TO INEXACT INFERENCE (Literature review)

A number of approaches to inexact inference are surveyed. The survey illustrates four points: 1) most approaches are chosen on an ad-hoc basis; 2) most approaches do not distinguish between different types of inexactness; 3) Some approaches do not distinguish between conditionals and inference rules 4) terminology is used with different meanings by different investigators.

3 RATIONALE AND METHOD

Analysis of the literature suggests controversy over Fuzzy vs Probabilistic inference is due in part to the following: 1) lack of commonly accepted definitions; 2) lack of distinction between different aspects of inference; 3) lack of distinction between different types of inexactness; 4) differing theoretical frameworks.

It is proposed that the mathematical theory of lattices be used to represent and clarify differences between Fuzzy logic and Probability logic.

4 LATTICE-THEORETIC COMPARISON OF SYSTEMS OF LOGIC

A formal language L is presented and interpreted in a metric lattice. It is shown that under similar interpretations, different types of lattices give models for different systems of logic.

4.1 The formal language L is presented, and the well-formed formulas (wffs) are defined.

4.2 Interpret L in a metric lattice M.

An equivalence relation E is defined on the set W of wffs of L. The relation is further constrained to guarantee that the interpretation is in a lattice, and to ensure the complete algebraic characterization of the lattice.

The set W (wffs of L) is partitioned by E into a set WW of equivalence classes, to be used as elements of a lattice (individual sentences cannot be used, because the partial order of the lattice requires a previous definition of equality, but individual sentences are distinct objects.)

A partial order [= is defined on WW, and it is shown that the E classes in WW generate a lattice M under [=. A measure function m is defined on the lattice M, which accordingly is modular. A distance measure d is defined on WW in terms of m, allowing implication and complementation to be defined in terms of the meet and join of M.

4.3 Truth-values are assigned for the connectives of L in terms of the metric m of the lattice M.

4.4 Truth-value assignments for Standard Uncertainty Logic are derived.

4.5 Truth-value assignments for Fuzzy Logic are derived, under the supposition that M is a chain.

4.6 Truth-value assignments for Unconditional Probability Logic are derived, under the supposition that M is Boolean.

4.7 Truth-value assignments for Sentential Calculus are derived, under the supposition that M is a Boolean chain.

5 IMPLEMENTATION CONSIDERATIONS

VRP is used as an example to show how the systems of logic of section 4 can be implemented in software.

5.1 VRP multivalued inference is described, and it is shown how its sentence-weight access functions can be modified to simulate inference procedures other than its standard one.

5.2 It is shown how Standard Uncertainty Logic (UL) truth-values can be translated into VRP sentence weights and coweights, and how a UL inference procedure can be implemented with VRP.

5.3 It is shown how a Fuzzy Logic inference procedure can be implemented with VRP.

5.4 It is shown how a Probability Logic inference procedure can be implemented with VRP.

5.5 Differences between the inference procedures for Fuzzy Logic and Probability Logic are discussed.

1 NOTATION, DEFINITIONS, AND EXPLANATIONS

1.0 Overview

This section establishes notation and definitions, and explains concepts used in the sequel. Subsections are:

Section 1.1 Rulebase inference procedure - example.
Section 1.2 Logic and Sentential Calculus.
Section 1.3 Sets and Probability Theory
Section 1.4 Fuzzy sets and Fuzzy Logic.
Section 1.5 Lattices and Uncertainty Logic.

1.1 Rule-based Inference - An Example

The Autometrics Versatile Rule Processor (VRP) version 1 [1] is used for a simplified explanation of one type of rule-based inference. Only those features are mentioned which are necessary to illustrate some points about rule-based inference. (VRP is also used in section 5 to show how the theoretical results can be implemented).

VRP performs inference on (processes) rulebases containing rules of the form

IF <antecedent1>
<antecedent2>
....
<antecedentn> THEN <consequent>

where the antecedents and consequent are sentences. The antecedent list represents a conjunction; some other systems require the AND between antecedents to be explicitly stated [2].

While VRP processes rules, arrays of sentence values are maintained in memory. Each sentence has two values, called the weight and the coweight, representing evidence for and against the sentence, respectively. For many applications the coweight is not used [3], and will be ignored in this discussion.

The weight of a sentence can range between 0 (false) and 100 (true). In this discussion, only weights of 100 or 0 are considered. In section 5 multivalued cases are explained.

At startup the weight array is initialized to a value of "indeterminate", which can be represented as 50, for the sake of this discussion [4]. A rule is processed by examining weights of the antecedents; "indeterminate" weights which cannot be derived from other rules are queried for. If all the antecedents are true (all have weights of 100) the consequent is considered to be true, and is given a weight of 100. If any antecedent is established as not true, processing stops for that rule, and the consequent retains its initial value of "indeterminate".

The list of antecedents of a rule can be thought of as one conjunctive sentence of the form A1 and A2 and ... and An, where the Ak's are antecedent sentences. If any conjunct is false, the whole sentence is false; the whole sentence is true only when all conjuncts are true. Thus any rule can be represented as

IF A THEN C

where A is a conjunctive antecedent sentence and C is a consequent sentence. The VRP inference procedure, as applied to an individual rule, can be represented by the inference policy:

When the sentence IF A THEN C is true, and the sentence A is true, infer that the sentence C is true

The sentence IF A THEN C is given as true, since it is one of the rules in the rulebase; VRP then needs to determine whether the sentence A is true, in order to determine whether the sentence C is true.

1.2 Logic

A logic is a formal language whose symbols are manipulated according to explicit rules. The symbols represent variables, connectives, and punctuation. The rules applying to the symbols have to do with grammar (syntax), meaning (semantics), and reasoning (inference) [5].

There are two principle approaches to presenting systems of logic, through axioms (proof theory) and through truth tables (model theory). The proof-theoretic approach uses deduction of valid formulas from other valid formulas, while in the model-theoretic approach truth values of sentences are computed from the truth values of other sentences. For the logic considered in this section, the two approaches give equivalent results [6].

Since the results of this study are motivated by VRP type inference procedures, which are implemented through computation of truth values, the systems of logic herein are presented model-theoretically.

Inference has traditionally been categorized into two principle kinds, deductive and inductive. Deductive inference has been characterized as drawing particular conclusions from general premises, usually with two-valued logic (thus presupposing no uncertainty). Inductive inference has been characterized as drawing general conclusions from particular premises, usually with multi-valued (probability) logic to accommodate uncertainty [7].

This study is concerned with inexact (uncertain/imprecise) inference, not with differences between deduction and induction. The term "inference" is used herein to mean the computation of conclusion values from the values of premises and conditionals.

1.2.1 Sentential Calculus as a formal language

The rules of the grammar (formation rules) govern valid combinations of symbols (well-formed formulas, or wffs). The connective symbols, such as &, V, -, represent relationships between wffs. The Sentential Calculus (SC) [8] is an example:

1-2-1 Let SC = < (,), -, &, V, S, >, Where

- (,) are punctuation symbols
- \- is a unary connective
- &, V are binary connectives
- S denotes a set of individual constants

1-2-2 Let p,q be constants in S. Then the wffs of SC are

p, (-p), (p & q), (p V q)

In the sequel the individual constants of S are construed to be finitely many sentences.

1.2.2 Semantics for SC

The rules of meaning (interpretation) specify semantics for the language. The interpretation uses a valuation function t which maps formulas into truth-values (hereafter, "values"). A formula is said to be valid when it is true for any model (any combination of truth-value assignments) [9].

The usual interpretation for Sentential Calculus (SC) is that formulas are sentences whose values are true = 1 and false = 0; that -p means "not p", p & q means "p and q" p V q means "p or q". Using "iff" to mean "if and only if", the semantics of SC can be specified as follows [10]:

Let t be the valuation function t --> {0,1}

1-2-3 t(p) = 1 iff p is true
t(p) = 0 iff p is false

1-2-4 t(-p) = 1 iff t(p) = 0
t(-p) = 0 iff t(p) = 1

1-2-5 t(p & q) = 1 iff t(p) = 1 and t(q) = 1
t(p & q) = 0 iff t(p) = 0 or t(q) = 0

1-2-6 t(p V q) = 1 iff t(p) = 1 or t(q) = 1
t(p V q) = 0 iff t(p) = 0 and t(q) = 0

A **conditional** is a formula which represents a dependency of a formula q upon a formula p. For SC the conditional p => q means "p entails q". **p** is called the **antecedent**, q is called the **consequent**) [11]. Some other translations are "p implies q", "if p then q". The latter has become a standard format for production rules in rulebased expert systems [12]; eg, VRP rules correspond to conditionals.

An **Implication operator** is an expression which assigns (truth) values to a conditional p => q in terms of the (truth) values of its constituent sentences p and q [13].

Using "=" to mean "is equivalent to", the conditional p => q can be defined as -p V q [14]:

1-2-7 (p => q) = (-p V q)

Then the implication operator t(p => q) has the same value as t(-p V q):

1-2-8 t(p => q) = 1 iff t(-p) = 1 or t(q) = 1
t(p => q) = 0 iff t(-p) = 0 and t(q) = 0

1.2.3 Inference for SC

The rules of inference (transformation rules) govern valid manipulation of formulas. Applying transformation rules to valid formulas allows the derivation of other valid formulas. An inference rule for SC, which is used in some expert systems is Modus Ponens [15]:

1-2-9 From t(p) = 1 and t(p => q) = 1 infer t(q) = 1

Thus the inference rule for VRP corresponds to Modus Ponens, the only difference being the range of values for the sentences.

1.2.5 Discussion

Among the valid formulas of SC are the following [16]:

1-2-10 p V -p
1-2-11 -(p & -p)
1-2-12 (p & (p => q)) => q

Expression 1-2-10 is sometimes called the law of excluded middle. Hereafter, any logic which has a valid sentence of the form 1-2-10 is said to be "Exclusional".

Expression 1-2-12 is a counterpart to Modus Ponens,but it is not an inference rule, it is just a (valid) sentence of SC. Inference rules are used in a metalanguage to apply to (mention) sentences in the object language, in this case SC [17].

The connectives are interpreted such that the truth-values of any sentences p, q uniquely determine the truth-values of -p, p V q, p & q, and p => q. Since any compound sentence can be reduced to combinations of these forms, the value of any sentence can be determined from the values of its constituent sentences; the value of a compound sentence is a function of the values of its constituents. A logic with this property is said to be Truth-functional [18].

It is difficult to represent inexact reasoning in this two-valued interpretation of SC, as its sentences must be either true or false. Since inexact reasoning is necessary in many applications [19], other schemes have been tried, among them the use of probabilities.

1.3 Sets and Probability Theory

Probability measures have been interpreted in different ways. One approach treats probabilities as measures on sets, while another approach treats probabilities as degrees of belief in the truth of sentences [20]. In this section the former approach is presented, and the latter approach is discussed. In section 6 the latter approach is developed in more detail.

Section 1.3.1 contains a review of some properties of sets. Section 1.3.2 contains a description of statistical probabilty measures.

1.3.1 Sets

Set theory concerns relations between certain kinds of objects, and operations upon them. The objects are elements and sets. Elements can be any kind of object, concrete or abstract, including sets. Sets are abstract collections of elements. The set with no elements is called the Null set [21].

A relation which holds between elements and sets is called the membership relation, specifying which elements belong to a given set. A relation which holds between sets and other sets is called the subset relation, specifying which sets are included in a given set. The set including all sets in some context is called the Universe of discourse.

1-3-1 Let ST = < (,), C, ~, O, U, D, W >, where

(,)	are punctuation symbols
C	is a binary relation
~	is a unary operation
O, U	are binary operations
D	denotes a set of elements
W	denotes the set which includes all sets of D

Let m be the membership relation m--> {0, 1}. For x belonging to D in W, A in W, define [22]:

1-3-2 m(x,A) = 1 iff x is a member of A
m(x,A) = 0 iff x is not a member of A

The customary notation "x e A" is used in the sequel:

1-3-3 x e A iff m(x,A) = 1 (x belongs to A)
x ~e A iff m(x,A) = 0 (x doesn't belong to A)

1-3-4 Subset relation; A is a subset of B (A C B):

A C B iff for every x, if x e A then x e B

1-3-5 Null set: there exists 0 C W such that,
for every x e W, x ~e 0

1-3-6 Equality: A = B iff A C B and B C A

Operations allow sets to be combined into new sets. The union of sets A and B is the set whose elements belong to either A or B, that is, all elements of both sets. The intersection of sets A and B is the set whose elements belong both to A and to B, that is, only the elements common to both sets. The complement of a set A is the set of elements of the universe which do not belong to A.

1-3-7 Operations: union, intersect, complement [23]:

For every x e D, and A, B C W

x e A U B iff x e A or x e B (union)
x e A O B iff x e A and x e B (intersection)
x e A ~ B iff x e A and x ~e B (complementation)

The operations of union, intersection, and complementation as defined above have certain algebraic properties [24], which are given next without proof:

1-3-8 Algebraic properties of set operations

U, O have the following properties:

(A U B) U C = A U (B U C),
(A O B) O C = A O (B O C) associative

A U B = B U A, A O B = B O A commutative

A U A = A, A O A = A idempotent

(A U B) O C = (A O C) U (B O C),
(A O B) U C = (A U C) O (B U C) distributive

For the universe W, the null set 0, and any set A:

W U A = W 0 O A = 0
0 U A = A W O A = A identity
~A = W - A
A O ~A = 0 A U ~A = W contradiction, exclusion

1.3.2 Statistical Probabilities

An event is defined as a subset of points in a sample space S. The size of an event is the number of points belonging to it. A simple event is a singleton set, and a composite event is a combination of simple events. For example, the composite event (P & Q) is defined as the set intersection of events P and Q. The probability of an event P, p(P), is defined as the ratio of the size of P n(P) to total size of the Sample space n(S): [25].

Let PC be ST augmented by:

S C W is a sample space of events.
p is the probability assignment p --> [0,1].
n(P) = the number of points in P C S
n(S~P) = n(S) - n(P)

Define:

p(P) = n(P)/n(S) = probability of event P in S
p(-P) = p(S~P)

Among the consequences of these definitions are [26]:

1-3-11 p(-P) = 1 - p(P) Complementation
1-3-12 p(PV-P) = 1 Exclusion
1-3-13 p(P&-P) = 0 Contradiction

1-3-14 p(PVQ) = p(P) + p(Q) - p(P&Q)

1-3-15 p(P & Q) <= min [p(P), p(Q)]

Sometimes it is not feasible to find the probability of a composite event by determining its size. For example, to find p(P & Q) from p(P) and p(Q), we must know the size of P O Q. In many non-trivial applications this can be difficult or impossible to determine. One way of estimating the probability of a composite event is the Baysian approach, using conditional probabilities [27].

The conditional probability of Q given P, p(Q|P), is defined as the ratio of the joint probablity of P & Q, to the probability of P [28]:

1-3-16 p(Q|P) = p(P & Q)/p(P)

Noting that p(P & Q) = p(Q & P), a simplified form of Bayes' rule can be derived from 1-3-16:

1-3-17 p(Q|P) = p(P|Q) x [p(Q) / p(P)]

where p(P), p(Q) are called the prior probabilities of P and Q, p(P|Q) is called the likelihood of P given Q, and p(Q|P) is called the posterior probability of Q given P.

Bayes' rule is useful when the likelihood of P given Q can be estimated even though the size of overlap of P and Q is unknown [29].

1.3.3 Discussion

By 1-3-14, to find p(PVQ) from p(P) and p(Q), we must know p(P&Q); ie, we must know the degree of overlap between sets P and Q. However, if P and Q are disjoint, n(POQ) = 0, so

p(PVQ) = p(P) + p(Q), if P O Q = 0

Thus in order to be computationally applicable, P and Q must be known to be disjoint. In many non-trivial applications this can be difficult or impossible to determine [30]. When P and Q are not disjoint and the dependence of Q on P can be estimated, Bayes' rule can be used.

Investigators differ over whether Bayes' rule should or can be used for deductive or inductive inference (Literature Survey (section 2.2.5). Another form of probabilistic inference, analogous to Modus Ponens, can be derived in a different version of probability theory. In this version probability values are assigned to sentences, rather than to sets of events [31]. The conditional is interpreted as

p(P => Q) = p(-P V Q)

where P, Q are sentences, as in SC. When probabilities are assigned to sentences, they play the role of (multi-) truth-values, and the result is a probability logic PL.

It was noted that Sentential Calculus (SC) is both exclusional and truth-functional. Probability theory is weaker than SC in that it too is exclusional (1-3-12), but it is not truth-functional (1-3-15); the most that can be deduced about (P & Q) is that it is less or equal to the least of p(P) or p(Q). As a result, probability theory can be computationally intractable [32].

Another shortcoming arises from the use of a single number to represent a degree of confidence or belief, based on available evidence. In cases where not enough is known to reach a conclusion with certainty, the degree of ignorance cannot be represented [33]. Moreover, there may be some evidence for a conclusion, but also conflicting evidence against the conclusion. Two measures would be required in this case, one pro and the other con [34].

These difficulties have led to other ways of dealing with inexactness. One approach involves generalizing Bayes' rule to interval-valued probabilities, by maintaining upper and lower limits [35]. Another approach involves retention of truth-functionality, but weakening the principle of exclusion. This is the approach taken by fuzzy logic.

1.4 Fuzzy sets and Fuzzy logic.

In preceding sections sets were mentioned in presenting logic and logic was used in presenting sets. In this section the principle of extensionality is introduced to relate fuzzy sets and fuzzy logic. The subsections are:

Section 1.4.1 Fuzzy set theory.
Section 1.4.2 Extensionality and Fuzzy Logic.
section 1.4.3 Discussion of results.

1.4.1 Fuzzy sets

The theory of fuzzy sets [36] is a formal method for representing the fuzziness of situations in the world. In many cases categories are not precise but are matters of degree. For example, the set of tall people can be represented as a fuzzy set; there is no precise boundary between being tall and not being tall.

For a crisp set, an item either is or is not a member of it. The membership function for a fuzzy set takes values from the interval [0,1]. A fuzzy set can be represented as a set of ordered pairs <x,n>, where x is an element of the set and n is the value of the membership of x in the set. Thus fuzzy sets are sometimes written in enumerated form as

$$Af = \{ x1|m1, x2|m2, \ldots \}$$

where mi is the degree of membership of xi in Af [37].

1-4-1 Let FS = < (,), C, ~, O, U, D, W >, where

(,) are punctuation symbols

C is a binary relation

~ is a unary operation

O, U are binary operations

D denotes a set of elements

W denotes a set of sets

Let f be the membership function f --> [0, 1]. For x belonging to D, D in W, A in W, define

1-4-2 x(A) = f(x, A) the degree of membership of x in A

1-4-3 x(A) > 0 iff x is a member of A

x(A) = 0 iff x is not a member of A

1-4-4 A = B iff for every x, x(A) = x(B)

1-4-5 Subset relation; A is a subset of B (A C B):

A C B iff for every x e A, x e B, x(A) <= x(B)

1-4-5 Null set 0: there exists 0 C W such that for every x, x(0) = 0

A crisp set can be represented as a fuzzy set whose elements all have a membership of 1. Thus the theory of fuzzy sets can be viewed as a generalization of crisp set theory. Then fuzzy set operations of intersection and union should have the same properties as for crisp sets, when the membership is 0 or 1 [38].

1-4-7 Operations: For every x e D, and A, B C W

x(A U B) = max[x(A), x(B)] (union)

x(A O B) = min[x(A), x(B)] (intersection)

x(~A) = 1 - x(A) (complementation)

1-4-8 Algebraic properties

U, O are associative, commutative, idempotent and distributive [39].

For the universe W, the null set 0, and any set A:

W U A = W   0 O A = 0
0 U A = A   W O A = A   identity

But A O ~A <> 0, A U ~A <> W unless x(A) = 1 or x(A) = 0.

Thus all the properties of crisp set theory also hold for fuzzy set theory, except exclusion.

1.4.2 Fuzzy Logic and Extensionality

Fuzzy logic can be related to fuzzy set theory through the principle of extensionality [40]: a predicate P' can be formed which is assertable of elements of a set P. For example, the predicate "is an integer" is assertable of elements of the set of integers. Then the sentence form "x is an integer" is true iff x belongs to the set of integers.

Similarly, truth-values of fuzzy predicates can be defined in terms of fuzzy set membership functions; the fuzzy set P determines the fuzzy predicate P', and the truth-value of "x is P'" is the same as the membership of x in P. Thus the notion of binary truth-values is extended to a continuous gradation in truth-values between 0 and 1.

The set of elements which are P' (ie, which are in P) is called the extension of the predicate P'. This is a semantic assignment, specifying the connection between truth values of statements [41] in a logic and relationships between the referents of the logic.

1-4-9 Let FL = < (,), -, &, V, W, S, A >, Where

- (,) are punctuation symbols
- \- is a unary connective
- &, V are binary connectives
- W denotes a set of individual variables
- S denotes a set of fuzzy subsets of W
- A denotes a set of predicate constants

1-4-10 Let x, y be elements of W
Let P', Q' be predicate constants
Let p denote a predicate-element pair P'x

Then the wffs of FL are

p, (-p), (p & q), (p V q)

1.4.2.1 Semantics of FL

The semantics of FL [42] can be specified as follows:

Let v be the valuation function v --> [0,1]
Let f be the membership function f --> [0,1]

1-4-11 v(p) = f(P), p = P'x, P a fuzzy subset of S

As a generalization of SC, connectives for fuzzy logic should give the same truth tables as for SC, when the truth values are restricted to {0,1}.

1-4-12 v(-p) = 1 - v(p)

1-4-13 v(p & q) = min[v(p), v(q)]

1-4-14 v(p V q) = max[v(p), v(q)]

The implication operator v(p =) q) is defined as

1-4-15 v(p =) q) = 1 - v(p) + v(p & q)

1.4.2.2 Inference for FL

Since FL is a generalization of SC, an inference rule for FL could be a generalization of Modus Ponens [1-11]. Thus a fuzzy Modus Ponens would have the form

From $v(p) = i$ and $v(p =) q) = j$ infer $v(q) = k$

Consider two cases, $j < 1$ and $j = 1$.

a) Assume $j < 1$. Then by 1-4-13,15

$1 - v(p) + \min(v(p), v(q)) < 1$ so $\min(v(p), v(q)) < v(p)$, and by 1-4-15 $j = 1 - i + v(q)$, or $v(q) = i + j - 1$.

b) Assume $j = 1$. Then by 1-4-13,15

$1 - v(p) + \min(v(p), v(q)) = 1$ so $\min(v(p), v(q)) = v(p)$, ie, $v(q) >= i$. Unlike case a), a specific value cannot be determined for $v(q)$, because that term drops out when $v(p)$ is substituted for $\min(v(p), v(q))$ in 1-4-15.

Thus an inference rule Fuzzy Modus Ponens for FL is

1-4-16 From $v(p) = i$ and $v(p =) q) = j$ infer
When $j < 1$, $v(q) = i + j - 1$
When $j = 1$, $v(q) >= i$

To show that Modus Ponens is a special case of Fuzzy Modus Ponens, When $i = 1$ and $j = 1$, by 1-4-13,15:

$1 = 1 - 1 + \min(v(p), v(q))$, so $\min(v(p), v(q)) = 1$, and thus $v(q) = 1$.

Note that when v(p =) q) < 1, v(p) + v(p =) q) = v(q) + 1, so v(p) + v(p =) q) >= 1 [by 1-4-16]. Therefore it is not possible for both v(p) < .5 and v(p =) q) < .5. When both antecedent and conditional are too low in value, no inference is possible. This property is consistent with Modus Ponens, where no inference is warranted unless both antecedent and conditional are sufficiently high in value (= 1); for Fuzzy Modus Ponens as a generalization of Modus Ponens, "sufficiently high" in value means >= .5.

1.4.3 Discussion

Among the valid formulas of FL is the following:

1-4-17 v(p V -p) + v(p & -p) = 1

Expression 1-4-17 is a generalization of the law of excluded middle. When v(p V -p) = 1, v(p & -p) = 0, but v(p V -p) can vary between .5 and 1. Thus FL is not exclusional.

How is the "shape" of a given fuzzy set determined? Figures 1 and 2 show graphs of two different plausible definitions of the fuzzy subset "tall" of the set of heights. Graph (1a) is a curve constructed to pass through three arbitrary but plausible points representing "not tall", "kind of tall", and "definitely tall". Graph (2a) is a curve passing through points representing various degrees of tallness; these points are averages of ratings given by a small sample of people. Which definition is "correct"?

Whatever sort of definition is selected for particular fuzzy sets, a similar problem arises for defining The fuzzy set operations of union and intersection. The operations defined in 1-4-7 are not the only ones which have crisp union and intersection as special cases. The following have also been proposed [43]:

1-4-18 x(A U B) = x(A) + x(B) - x(A) X x(B)
1-4-19 x(A O B) = x(A) X x(B)

Although both sets of definitions give the same values for x(A), x(B) = 1 or 0, they give different values for other values of x(A), x(B); for example, if x(A) = .4, x(B) = .6,

min(x(A), x(B))	= .4	max(x(A),x(B))	= .6
x(A) X x(B)	= .24	x(A)+x(B) - x(A) X x(B)	= .76

Some investigators regard the arbitrary choice of operators as unsatisfactory from a theoretical point of view [44]. But others regard this flexibility as an opportunity, from a practical point of view, to select the operators best suited to a particular application. Pal et al [45] find uses for several different operators in the field of pattern recognition, so they formally define several different operators within one theory of fuzzy sets.

Other fuzzy operations without crisp counterparts have been defined. Zadeh [46] proposed various operations on fuzzy sets to represent common linguistic hedges, such as

1-4-20 very A = A^2 = (x(A))^2 for all x

Graphs for "very tall" are depicted in figures 1b and 2b.

The same problem/opportunity arises for the connectives of fuzzy logic. As with fuzzy set operators, different connectives can be defined which have the corresponding connectives of SC as special cases. In particular, given common definitions for conjunction, disjunction, and complementation, different implication operators can be defined which have the SC implication operator as a special case. For example, another implication operator which has been proposed for fuzzy logic is [47]

1-4-21 v(p =] q) = min [1, v(q)/v(p)]

Figures 3 and 4 show truth-tables for the operators 1-4-15 and 1-4-21, respectively.

Fuzzy Modus Ponens 1-4-16 has the property that, if we know v(p =) q) with certainty, we cannot determine v(q) with specificity. But if we do not know v(p =) q) with certainty, then we can determine a specific value for v(q). A fuzzy Modus Ponens corresponding to the implication operator 1-4-21 has the same property.

Although fuzzy logic has been criticized as being arbitrary, Gaines [48] argues that the particular forms of connectives used is not so important as the structures of their applications. He proposes a generalization encompassing both probability and fuzzy logic, called Standard Uncertainty Logic (UL). While probability logic is defined over crisp sets, and fuzzy logic is defined over fuzzy sets, UL is defined over a lattice. Lattices are described in the next section.

1.5 Lattices and Standard Uncertainty Logic

Standard Uncertainty Logic (UL) [49] is defined in terms of a lattice whose elements are sets of equivalent sentences. Special cases of UL are fuzzy logic, probability logic, and standard two-valued logic. Subsections are:

1.5.1 Lattices

A lattice is a partially ordered set in which every pair of elements has a greatest lower bound and a least upper bound [50]. In a partially ordered set some pairs of elements may not be comparable. For example, figure 5a depicts a lattice whose elements are subsets of the set {a, b, c}. Here the partial order relation is set inclusion; for sets {a,b} and {a,c} it is not the case that {a,b} C {a,c} or {a,c} C {a,b}.

However, the two elements are indirectly related through other elements in two ways. First, they are both subsets of {a,b,c}. This relation is an operation called "join", labelled V: {a,c}V{a,b} = {a,b,c}. Second, they both contain a common subset {a}. This relation is an operation called "meet", labelled A: {a,c}A{a,b} = {a}. With lattice elements interpreted as subsets, join and meet correspond to set union and set intersection, respectively.

As another example, consider the lattice shown in figure 5b, which happens to have the same structure as 5a. Here the partial order is interpreted as "divisor of". In this case, 6 is not a divisor of 10, nor conversely, but they are both divisors of 30; thus 6 V 10 = 30, and 6 A 10 = 2.

1-5-1 Let LAT = < A, V, S, [=, >, where

A, V are binary operations
S denotes a set of elements
[= is a binary relation such that, For every p, q e S,

p [= p reflexive

if p [= q and q [= p then p = q antisymmetric

if p [= q and q [= s then p [= s transitive

and r,s exist such that r [= p, q [= s and pAq = r, pVq = s

but not trichotomous: not the case that for all p, q in S: p [= q or q [= p. Then the relation [= is a partial order; ie, it is reflexive, antisymmetric, and transitive [51].

Let T be a subset of the set S partially ordered by [=, where p e S; then

p is a lower bound (LB) of T iff for every q e T, p [= q.

p is an upper bound (UB) of T iff for every q e T, q [= p.

Thus in figure 5b, 15 is an upper bound of 3, 5 and 1, while 3 is a lower bound of 6, 15 and 30.

p is the greatest LB (GLB) of T iff for every LB q of T,
q [= p.

p is the least UB (LUB) of T iff for every UB q of T,
p [= q.

When a GLB (LUB) exists, it is unique, for then the set of LBs (UBs) is not empty, and the GLB (LUB) is the greatest (least) element of this set.

A LUB which has no UB (other than itself) is called the greatest element. In figure 5a the greatest element is {a,b,c}; in figure 5b it is 30. A GLB which has no LB (other than itself) is called the least element. In figure 5a the least element is {}; in figure 5b it is 1.

Given 1-5-1, it can be shown that for every p, q in S there are r, s, T, F in S such that

1-5-2 r = pVq r is least upper bound of p,q
s = pAq s is greatest lower bound of p,q
F = FAp, p = FVp F is the least element
p = TAp, T = TVp T is the greatest element

Then LAT is a lattice [52] and A, V can be shown have the following algebraic properties [53]:

1-5-3 pAp = p, pVp = p idempotent

1-5-4 pAq = qAp, pVq = qVp commutative

1-5-5 pA(qAs) = (pAq)As

associative

1-5-6 pV(qVs) = (pVq)Vs

1-5-7 pA(pVq) = pV(pAq) = p absorptive

1-5-8 p [= q iff pAq = p and pVq = q consistency

Additional algebraic properties can be specified to produce different types of lattices. In this study the following types are considered: Chain, Complemented, Modular, and Distributive lattices.

A Chain is a lattice whose partial order is also a simple order: each element is comparable with every other, as in figure 6a.

Let CHN be a lattice as in 1-5-1 through 1-5-8, and also

For every p,q in S, either p [= q or q [= p

Then CHN is a chain [54]. By definition, any two elements of a chain are comparable.

An element p of a lattice has a complement p' if p A p' = F and p V p' = T. In figure 6b both 6 and 12 are complements of 10; in figure 6c x and y are complements; but in the chain of figure 6e u and v have no complements, although T and F are complements [55].

If every element of a lattice has at least one complement, the lattice is said to be Complemented. Thus the lattice of figure 6b is complemented, as are 6c and 6d, but neither 6e nor 6f are complemented.

Let CMP be a lattice as in 1-5-1 through 1-5-8, and also, for every p in S there is a q in S such that

p A q = F, p V q = T

Then CMP is Complemented [56].

A lattice is modular if, for any pair of comparable elements, there is some element comparable to one of them. That is, if there is a pair of comparable elements and another element comparable to neither, the lattice is not modular. The lattices of figures 6e, 6c, 6d, 6f are all modular; but 6b is not, since 6 and 12 are comparable, but 10 is comparable to neither. In fact, every non-modular lattice contains a sublattice of the form diagrammed in 6b.

Let MOD be a lattice as in 1-5-1 through 1-5-8, and also

For every p,q,r in S, if q [= p then

p A (q V r) = (p A q) V (p A r) = q V (p A r);

Then MOD is Modular [57]

A lattice is distributive iff relative complements are uniquely determined (relative complements are complements relative to a sublattice). The lattices of figure 5 are distributive, but 6b, 6d are not; in 6b, 10 has both 6 and 12 as complements, while in 6d, 4 has both 6 and 10 as complements.

Let DST be a lattice as in 1-5-1 through 1-5-8, and also

For every p,q,r in S

p A (q V r) = (p A q) V (p A r)

Then DST is Distributive [58].

A Boolean lattice is distributive and complemented [59].

A lattice is Metric iff there is a real-valued function v defined on it such that, for every p,q in L

i 0 <= v(p)
ii p [= q ==> v(p) <= v(q)
iii v(p) + v(q) = v(p A q) + v(p V q)

Any metric lattice is modular [60].

1.5.2 Standard Uncertainty Logic

The material presented in this section is developed in detail in section 4; consequently this section contains only a sketch of the interpretation of Standard Uncertainty Logic (UL) in a modular lattice M, with many details omitted.

Figures 5a and 5b depict different interpretations of the same abstract lattice. Each interpretation is a representation of a different domain. The meaning of join (V) and meet (A), and the elements of the lattice, depends upon the domain being represented.

For UL [61] the elements of the lattice M are interpreted as sets P of equivalent sentences p. The join P V Q is interpreted as "p or q" and the meet P A Q is interpreted as "p and q". The greatest and least elements are interpreted as special sets of sentences "T" (eg, "1=1") and "F" (eg, "1=0") respectively.

A measure function m is defined on the lattice M, which thus is modular; implication and complementation can then be defined in terms of meet and join, even though M itself is not complemented. Implication and complementation are defined in terms of the distance between elements.

Let M = < WW, F, T, A, V > be the lattice of the set WW of elements partially ordered by [=, with meet A and join V, maximal element T, minimal element F.

Define the measure function m on M, m:WW --> [0,1] such that for T,F and every P,Q e WW

1-5-9 m(T) = 1, m(F) = 0
1-5-10 m(P) <= m(Q) iff P [= Q
1-5-11 m(P A Q) + m(P V Q) = m(P) + m(Q) additivity

Then M is a metric lattice, hence modular [62].

In a metric lattice the distance d between P and Q is defined as [63]: for every P,Q in WW:

1-5-12 d(P,Q) = m(P V Q) - m(P A Q)

A valuation function v is defined in terms of the measure m; v is used for basic truth-value assignments to the connectives of L, in terms of meet A and join V of the lattice M. It is intended that 'and' (&) corresponds to meet 'A', 'or' (|) corresponds to join 'V'.

Define the valuation function v on W: v:W --> [0,1] such that for every P,Q in WW, s,t in W, p in P, q in Q,

1-5-13 v(s) = m(P) iff s is in P
1-5-14 v(s & t) = m(P A Q) iff s is in P and t is in Q
1-5-15 v(s | t) = m(P V Q) iff s is in P or t is in Q

An implication operator is suggested by analogy with SC, where p => q is true iff p&q has the same value as p. The counterpart to this condition for UL is that P A Q = P. Thus an implication operator is defined in terms of a distance function between elements of M, so that the closer P is to P A Q, the more strongly does p imply q. This implication operator turns out to have the same form as the one for FL (1-3-15) [64].

1-5-16 v(p => q) = 1 - d(P, P A Q) = 1 - d(Q, P V Q)

The complement of P is defined in terms of the distance between P and F. The closer P is to F, the closer its complement -P is to T, and conversely. This definition for complementation turns out to have the same form as the one for FL (1-3-12) [65].

1-5-17 v(-p) = 1 - d(P, F)

The truth-value assignments for UL can be derived from the basic truth-value assignments 1-5-13 to 1-5-17, under the assumption that the lattice M is modular.

1-5-18 Truth-value assignments for UL: For every p,q in W,

UL1 0 <= v(p) <= 1
UL2 v(p & q) <= min[v(p), v(q)]
UL3 v(p | q) >= max[v(p), v(q)]
UL4 v(p => q) = 1 - v(p) + v(p & q)
UL5 v(-p) = 1 - v(p)

1.5.2.2 Inference for UL

As a generalization of FL, UL could have an inference rule which is a generalization of Fuzzy Modus Ponens. Thus UL Modus Ponens would have the form

From v(p) = i and v(p => q) = j infer something about v(q).

Since v(p => q) = 1 - v(p) + v(p&q), j = 1 - i + v(p&q), so v((p&q) = j + i - 1. But v(p&q) <= v(q), so j + i - 1 <= v(q), which thus has a lower bound.

Thus an inference rule UL Modus Ponens is

1-5-19 From v(p) = i and v(p => q) = j infer v(q) >= j+i -1

For the fuzzy modus ponens 1-4-16, no inference is warranted when both v(p) and v(p =) q) are less than .5. For UL modus ponens the case is not quite so definite; because of the inequality in 1-5-19, v(q) is guaranteed to be nonzero only when v(p) + v(p => q) >1. Thus when both v(p) and v(p => q) are less than .5, inference is not unwarranted, but it is pointless, since nothing can be concluded about the value v(q) in that case.

1.5.3 Discussion

Among the valid formulas of UL are [66]

1-5-20 v(p|q) + v(p&q) = v(p) + v(q)

Substituting -p for q in 1-5-20, it follows (UL5) that

1-5-21 v(p|-p) + v(p&-p) = 1

By 1-5-21 UL is not exclusional, just as for FL (1-4-17).

By 1-5-18, UL2-3, UL is not truth-functional, since the truth-value of a compound statement is not uniquely determined by the values of its constituents. The value of p&q can range between 0 and min[v(p), v(q)]; even though both v(p) and v(q) are known, still v(p&q), v(p|q), and hence v(p => q) cannot be determined.

1.6 Summary and discussion

Sentential Calculus (SC), Probability Theory (PC), and Fuzzy Logic (FL) are apparently rather different systems. SC has only two truth values, while PC and FL are multi-valued. PC is defined on crisp sets, while FL is defined on fuzzy sets. SC is exclusional and truth-functional, PC is exclusional but not truth-functional, and FL is truth-functional but not exclusional.

But by representing the three systems in terms of UL, their similarities can be seen. If the lattice in which UL is interpreted be restricted such that 1-6-1 can be derived:

1-6-1 $v(p \mid -p) = 1$ excluded middle

Then by 1-5-21 we have

1-6-2 $v(p \,\&\, -p) = 0$ non-contradiction

The result is a logic of unconditional probability PL [67].

If the lattice in which UL is interpreted be restricted such that 1-6-3 can be derived:

1-6-3 $v(p \mid q) = \max[\, v(p), v(q) \,]$ |-truth-functionality

Then 1-6-4 can be derived [appendix, theorem T6]:

1-6-4 $v(p \,\&\, q) = \min[\, v(p), v(q) \,]$ &-truth-functionality

the result is a fuzzy logic FL (see 1-4-13,14)

If the lattice in which UL is interpreted be restricted such that both 1-6-1 and 1-6-3 can be derived, the resulting logic can have only 0 and 1 for truth-values [appendix 2, theorem T11], and the result is SC.

In section 4 differences between these three systems are shown to correspond to differences in the kind of lattice in which UL is interpreted. But before that, section 2 contains a review of the literature, indicating a need for such an analysis; section 3 proposes the method to be used.

2 INEXACT INFERENCE, METHODS AND SELECTION

In this chapter a number of approaches to inexact inference are surveyed. Each paper illustrates one or more of the following points:

- o Most approaches are chosen on an ad-hoc basis.
- o Most approaches do not distinguish between different types of inexactness.
- o Some approaches do not distinguish between conditionals and inference rules.
- o Terminology is not standardized.

2.1 Some methods for inexact inference

2.1.1 Gaines & Shaw [68] discuss the need for approximate reasoning in rule processing applications, and the use of different logics for this purpose. They do not explicitly distinguish between different types of inexactness, nor do they define the various terms used for inexactness.

They give examples of "fairly vague but highly applicable rules" [69] such as those from a cement kiln operator's manual; these were implemented in a process control expert system using fuzzy logic.

While "our scientific training has taught us to regard imprecision as undesirable.." [70], expert system pioneers discovered that it is "..a necessary component of any system adequate to encode human knowledge." [71]. A major weakness of past information processing systems was their inability to cope with the ordinarily vague expression of information.

Imprecision arises not only through lack of information, but can also be intrinsic to situations: "A key aspect of executive action is planning under uncertainty and normal language provides a means for imprecision to be clearly and exactly expressed." [72]. This kind of expression can capture the actual meaning required far more accurately than if expressed with more precision than warranted by the situation.

Gaines' Standard Uncertainty Logic [73] is discussed as a means of managing both imprecision and non-truth-functionality (uncertainty due to lack of information). The need for such a logic "..is obvious in retrospect. Most of the distinctions we make in everyday life do not have hard boundaries." [74]. The constraints on such distinctions can be computed, but a single-valued result cannot be computed.

Using fuzzy logic allows single values to be calculated by imposing truth-functionality. "The common assumption of independence of events in probabilistic situations is an alternate way of forcing truth-functionality. Both these approaches are expedients whose utility depends on the problem domain." [75].

Standard Uncertainty Logic is an exception to the usual ad-hoc approach; as a generalization of both probabilistic and fuzzy logics, it has features common to both. However, although Gaines distinguishes between different logics of inexactness, he does not explicitly distinguish between different types of inexactness. This is apparent in the quotes above: the terms "vague", "imprecise", and "uncertain" are used more or less synonymously.

2.1.2 Mamdani and Efstathiou [76] discuss the limitations of existing logics for automated inference systems, including rule-based expert systems. They discuss the ad-hoc nature of such schemes, but do not explicitly distinguish between different types of inexactness.

They argue that engineering is more advanced than theory in this field, because the usual ad-hoc selection of inference procedures ignores the fact that these represent logics.

In two-valued logic, each logical operator (connective) has a unique truth table; in multivalued logic a given operator, such as implication, can be interpreted with a number of different truth tables [77]. The connectives thus become topic-dependent, since their choice depends upon the meaning of intermediate values. The power of the system for automated reasoning is reduced, because intervention of the user is necessary to interpret the meanings in terms of specific applications [78].

This dilemma between flexibility and mechanizability is exemplified by Zadeh's PRUF [79] language for meaning representation, where even the simpler translation rules are context dependent [80]. They conclude "Uncertainty is multi-faceted and fuzzy reasoning is likely to be able to handle its different forms better than Bayesian reasoning, even if it does so in a highly topic-dependent way." [81]

Mamdani and Efstathiou concur that most approaches are chosen on an ad-hoc basis. Their statement that "Uncertainty is multi-faceted.." expresses an implicit recognition that there may be different kinds of inexactness. They use the term "uncertainty" to generically denote the many facets of inexactness.

2.1.3 Boose [82] uses two different ad-hoc methods for managing inexactness, does not recognize different types of inexactness, and uses the term "certainty" in two senses.

His rulebase prototyping Expertise Transfer System uses both conclusion rule certainty factors and intermediate rule entailment strengths [83].

Conclusion rule certainty factors are based on EMYCIN certainty factors, ranging from -1 (certainly false) to +1 (certainly true), with a belief threshold of .2, below which a conclusion is not believed to be true [84].

Other definitions of uncertainty were considered, such as Bayesian probabilities, but the EMYCIN method was chosen because although the Bayesian method is theoretically more satisfactory, human experts found the EMYCIN uncertainty calculus more intuitively obvious, and moreover, most current systems use it [85].

Different methods have been used to generate EMYCIN type certainty factors; Boose chose a method which multiplies together three factors derived from 1) rulebase structure (number of rules involved), 2) rule content (certainty), and 3) rule importance (rated by expert) [86]. No rationale is given for choosing this approach. "Although no formal studies have yet been carried out, experts who use the system feel that its behaviour is reasonable." [87].

Intermediate rule entailment strengths are derived from empirically generated truth tables; these strengths represent "the certainty of belief ..[in].. rules' conclusions." [88] When multivalued rating scales are used, an algorithm maps "..ratings onto truth strengths" [89]. The resulting truth tables are truncated so that only values above an arbitrary threshold are used.

Although strengths calculated from the full truth table are "theoretically more sound than those based on .." the truncated table, the latter were used in practice because "experts seem to be overwhelmed with this much information, and prefer the .." threshold version [90]. This rather ad-hoc approach has been applied in prototyping over 70 different rulebases, from a wide variety of domains [91].

The ad-hoc nature of Boose's approach to inexactness management is apparent. He acknowledges different logics of inexactness, in having considered a Baysian approach, but there is no apparent recognition of different types of inexactness. He uses the term "certainty" (meaning "degree of certainty") as generic for inexactness, and he also speaks of "certainty of belief". Does this mean the certainty that someone believes something, or that someone believes that something is certain?

2.1.4 Whalen & Schott [92] discuss the ad-hoc nature of a number of inexactness management schemes, and distinguish between different locii, but not different types of inexactness; nor do they characterize the differences between the various terms they use for inexactness.

They categorize and analyze eight applications in terms of their types, and also in terms of the nature of inexactness management used. They discuss the types of "fuzziness in production rules" and the nature of "uncertainty in the data" [93].

The Whalen & Schott Business Decision Support system [94] is an off-line control system; rather than updating the database directly, it returns fuzzy ratings of alternatives to the user, who then updates the database. Antecedents and consequents are expressed with relational linguistic phrases ("greaterthan", "equalto") and linguistic constants ("low", "medium", "high").

This system employs the Lukasiewicz implication operator, which requires strong evidence to rule out any possibility. They say it is "most suitable for representing isolated chunks of knowledge rather than well-structured algorithms", but that its major disadvantage is the extreme vagueness of results in many cases. Thus their system sorts output in order of vagueness [95].

Sanchez' [96] Medical Diagnosis system establishes a fuzzy subset of possible diagnoses based on a fuzzy subset of possible symptoms; it is not really a production system, since it uses a single exhaustive rule for matching symptoms with diagnoses, rather than dividing knowledge into chunks. The system provides certainty measures using fuzzy sets. Input is a fuzzy set of discrete symptoms; there is no explicit database.

The Sanchez system uses the Brouwer/Godel implication operator, which supports inverse fuzzy relations and does not suffer from the same vagueness as the Lukasiewicz operator. The single rule is derived by composing a patient/diagnosis matrix and a patient/symptom matrix into a symptom/diagnosis matrix. The major disadvantage of this operator is that it is discontinuous for small changes in value of antecedent and consequent when they are close in value [97].

Weiss & Donnell [98], in their Room Temperature Control system, input a crisp control strategy to be evaluated. This is converted into fuzzy linguistic variables for internal representation. The fuzzy truth value of the implication operator is computed, and a performance score for the control strategy is determined as the minimum of its performance under all production rules. Automatic control algorithms convert this into a crisp evaluation of the proposed control strategy. They use the Kleene/Diens operator, which gives results tending toward "indeterminate" (= .5) [99].

Two other operators are analyzed, both of which are used in automatic control systems [100]. The Zadeh operator generalizes the IF-THEN-ELSE construct, while the Mamdani operator is a generalization of the biconditional rather than of material implication [101].

The Mamdani operator requires strong evidence to support a conclusion, and thus is popular in fuzzy control systems requiring crisp output even though the rules are fuzzy. It is most suitable when an exhaustive set of conditions is known [102].

Whalen and Schott attribute differences in selection of operators partly to the preferences and arbitrary choices of system designers. They conclude that "As ..experience.. increases, our ability to match the characteristics of the reasoning system to the characteristics of the problem should improve." [103]

The preferences and arbitrary choices leading to selection of an approach, noted by Whalen & Schott, are indicative of the ad-hoc nature of this selection. Their distinction between "fuzziness in rules" and "uncertainty in data" is an implicit recognition of different types of inexactness, but their explicit reference is to the locus of inexactness, not to the type. They refer variously to "fuzziness", "uncertainty", and "vagueness", without characterizing the differences between them.

2.1.5 The Dempster-Shafer (D-S) theory of evidence [104] is a generalization of Bayesian inference, applicable to sets of mutually exclusive and exhaustive hypotheses (called "frames of discernment" - hereafter "frames"). It allows accumulation of evidence for and against the hypotheses in a frame. Degrees of belief for sentences (subsets of the frame, representing disjunctions) and their complements are computed from basic probability assignments. Several functions are derived from the degree of belief in a sentence A, Bel(A): doubt in A, Dou(A) = Bel(-A); the upper probability of A, P*(A) = 1 - Dou(A); the belief interval [Bel(A), P*(A)] gives a measure of uncertainty. In its original form D-S theory is computationally inefficient, since it requires calculations for all subsets of the frame [105].

Shafer says of the theory that although he can provide "..no conclusive a priori argument, ...it does seem to reflect the pooling of evidence." [106]. Gordon & Shortliffe say of the theory "It is based on intuition of how evidence should combine, ...and not on any formal underlying theory." [107].

Barnett [108] shows that the exponential-time requirements of the D-S model can be reduced to simple polynomial time if the theory is applied only to simple hypotheses and their negations, and if the hypotheses are ordered as a vector. However, when degree of belief and the other measures have been computed for an application, a decision must still be made as to which measure to actually use:

> ...should the estimate provided by Bel or the one provided by P* be used? Perhaps something in between. But what? No one has a good answer to this question.
>
> Thus, the difference between the theories is that the Bayesian approach suppresses ignorance up front while the other must deal with it after the evidence is in. This suggests one benefit of the Dempster-Shafer approach: surely it must be right to let the evidence narrow down the possibilities first, then apply some ad hoc method afterward. [109]

Moreover, since Barnett's approach rules out disjunctions (intermediate subsets of the frame), hierarchical relationships between hypotheses cannot be handled.

Gordon and Shortliffe [110] propose a modification to the D-S theory which overcomes the latter limitation of Barnett's approach. Instead of doing away with all intermediate subsets of the frame, they restrict the allowable subsets to those which form a strict hierarchy (a tree), since this kind of structure is characteristic of many problem-solving situations. But when subsets are restricted to a tree, complements of some nodes do not occur as nodes in the tree, so belief which would be assigned to a missing node is assigned instead to the smallest set in the tree which includes the missing node as a subset [111].

But in this case, since the complement of a proposition may not be in the tree, the notion of belief interval is lost, ie, uncertainty cannot always be calculated [112]. As for the issue of how to use the belief measure after the scheme has been applied, Gordon & Shortliffe say "There is not likely to be a 'correct' approach to this problem because the nature of the actions based on evidence varies so greatly from one domain to another." [113]

Dong and Wong [114] extend the D-S theory to rule-based inference, allowing ignorance as well as evidence to affect confidence in conclusions. Evidences for antecedents of a rule are propagated with evidence for the rule, to become the evidence for the hypothesis (consequent) of the rule. They discuss applications to algorithmic "rules" (procedures) as well as to conditional rules.

Most current rule-based applications of Shafer's theory are limited to combining two independent sets of evidence for a given consequent. Two problems which prevent extensive use of the method in rule-based systems are combining uncertainties in antecedents and combining uncertainties from different rules with the same consequent [115].

A special case of their method gives a procedure for propagating conditional probabilities through rule networks. They compare this with Applebaum & Ruspini's [116] method for propagating upper and lower probability limits, noting that the limits disappear after the probability of the conclusion is computed, which is equivalent to treating the ignorance as equal to 0.

They conclude that their proposed methods are "..just ideas and should not be considered as theories." "Much more research is needed to substantiate the ideas.." [117].

It is not clear from their paper why Dong & Wong have chosen to extend Shafer's theory rather than some other approach; by default, then, their approach appears to be ad-hoc. They seem to admit as much in characterizing their work as "..just ideas..not..theories.". They do not distinguish between different types of inexactness. They use the terms "uncertainty" and "ignorance" generically and with apparent synonymy.

2.1.6 Gaines [118] combines fuzzy and probabilistic approaches to the induction of rules from Personal Construct Psychology repertory grid data. An information-theoretic measure of "surprise" is used to compute the significance (uncertainty reduction) of each rule, so that rules can be ranked by significance.

Each rule receives a probability as well as a fuzzy truth-value, and trade-offs between probabilities and truth-values are evaluated by significance analysis. A rule with lower truth-value can be more significant than another with higher truth-value [119].

The method is implemented in software called ENTAIL II, and the induced rules can be loaded directly into a standard expert system shell. This software is used by Boose [120] (section 2.1.3). The truth-value of the rule codes into the truth-value representation used by the shell. (No mention is made of whether or how the probabilities are encoded).

With this approach Gaines comes close to an implicit distinction between different types of inexactness. This is reflected in his use of the terms "surprise", "significance", and "uncertainty" for the probabilistic parts of the system, as distinguished from the term "fuzzy truth value" for the fuzzy part.

2.1.7 Summary

Various methods for inexact inference in rule processing applications have been used, but there is no clear rationale for choosing between them. They include: probability theory, Shafer belief functions, Zadeh's meaning representation language PRUF, EMYCIN uncertainty calculus, rating scales, and various fuzzy logical implication operators such as the Lukasiewicz, Brouwer/Godel, Kleene/Diens, Zadeh, and Mamdani operators.

Most methods have been chosen on an ad-hoc basis. None of them explicitly distinguishes between different types of inexactness. There is a diversity of terms for inexactness, including "certainty of belief", "fuzziness", "ignorance", "imprecise", "significance", "surprise", "uncertain", and "vague".

2.2 Approaches to selection of inexact inference methods

2.2.1 Cohen & Grinberg [121] exemplify a heuristic approach to a theory of inexactness. They argue that the propagation of various kinds of evidential weights over an inference net is domain-specific, and therefore should be handled as part of the rulebase, i.e., heuristically.

They discuss two kinds of uncertainty, that characteristic of rules and that arising from noisy data [122]. Current systems propagate uncertainty factors or "degrees of belief" over inference nets with combining functions, which yield numerical degrees of belief for conclusions.

They see several problems with this approach. A single numerical value tells nothing about precision of result; it cannot distinguish between disbelief and lack of evidence, since it combines evidence for and against a conclusion. The meaning of the numeric value is not clear, so explanations of results are limited to a recitation of the inferences leading to a conclusion [123].

Moreover, when Bayesian measures are used, statistical evidence is required in addition to degrees of belief, and subjective expert judgement is usually used instead, thereby vitiating the validity of this approach. Finally, numerical representation is inadequate to discriminate between different kinds of evidence, so different kinds of evidence cannot be reasoned about differently [124].

Analyzing an example from the domain of portfolio management, they find that reasoning about uncertainty is characterized by two main approaches: 1) attempts to explain it away, mainly by trading specificity for certainty, and 2) by accepting hypotheses as sufficiently certain. These approaches require much specialized domain knowledge; it is necessary to explicitly state reliability of evidence, and to clearly distinguish different kinds of evidence [125].

Cohen & Grinberg advocate a paradigm of "endorsements", or audit trail of the kinds of inferences which have taken place. Endorsements are of different types, but they can be ranked, although additional domain-specific knowledge is necessary to do so [126].

Endorsements are propagated over inferences by heuristics, so a set of meta-rules is needed. But such rules are idiosyncratic for each domain, and interactions between endorsements complicate the picture. Endorsements have an operational semantics; their meaning is determined by how a system reasons with them. Thus each domain has a characteristic set of endorsements and set of methods defining what they mean [127].

Discussion: heuristic approach

The heuristic approach (Cohen & Grinberg [128]) acommodates domain-dependent aspects of inexactness management, but the issue of selecting an implication operator is ignored.

Moreover, this approach compounds the problem of knowledge aquisition, which is the most expensive and time-consuming aspect of rule processing applications due to the difficulty of translating an expert's implicit knowledge and reasoning processes into explicit rules [129], [130].

Translating nuanced knowledge about inexactness, and meta-reasoning about reasoning would be even more difficult. Cohen & Grinberg say that reasoning about uncertainty requires much specialized domain knowledge, and a set of rules, idiosyncratic for each domain, is needed to propagate endorsements over inference nets; the situation is complicated by interactions between endorsements.

The heuristic approach is focused on domain-dependence, to accomodate idiosyncrasies of human inexact reasoning. Common properties of reasoning over different domains are ignored; empirical evaluation of performance would be necessary to validate a purely heuristic system.

Since Cohen & Grinberg do not discuss the nature of inference in detail, by default they do not distinguish between its different aspects. (section 1.2)

They use various terms for inexactness, such as "uncertainty" in rules, "noisy" data, "uncertainty factors", and "degrees of belief". These terms are not clarified.

2.2.2 Whalen & Schott [131] exemplify an empirical approach to a theory of inexactness. They compare the performance of 11 implication operators for a Financial Ratio Analysis rulebase, to determine which operator performs best in terms of correctness and precision [132].

The operators were evaluated on how closely their outputs matched the judgements of a human expert, for the simulated financial data. Two "experiments" were conducted; first the operators were compared for a no-problem scenario and two problem scenarios. In the first scenario all performed adequately, although three of them gave "mild false positive" results; in the second scenario all performed adequately; in the third scenario only four performed adequately, and three of these were the ones which gave mild false results in the first scenario [133].

In the second experiment, the four successful operators were subjected to a sensitivity analysis for the third scenario (which most strongly distinguished their performance from the others). Three of the four operators displayed significant sensitivity to the changes, and they had very similar response patterns. Curiously, these three operators are the same ones which were only marginally adequate in the first scenario! [134]

The paper concludes with a summary of these results (without mentioning the curiousity noted above); the results are neither analyzed nor explained.

Discussion: empirical approach

The empirical approach (Whalen & Schott [135]) addresses the necessity, in the absence of theoretical justification, to compare the performance of specific implication operators to determine the optimal one.

In Whalen & Schott's study, the only successful implication operators for their third scenario were also the only ones which were marginally unsuccessful for their first scenario. Thus the choice of the best operator for a situation can depend upon features of that specific situation, and a seperate empirical comparison would have to be done not only for all available operators, but also for all types of situations for a given application domain.

The empirical approach is focussed on finding the best implication operator for an application. Heuristic aspects of inexact reasoning are ignored, as are mathematical relationships between various operators.

Whalen & Schott do concisely characterize some aspects of inference. They distinguish between an implication operator, which "..constructs the implication relation 'IF...THEN' from the antecedent and consequent propositions", and "..ways of reasoning from implications..". They describe logics as being characterized by the definitions for their connectives, especially the implication operator, which they say constitutes the most significant difference among the proposed logics [136].

2.2.3 Prade [137] exemplifies the analytical approach to a theory of inexactness. He develops a unified mathematical framework for presenting most of the existing measures of inexactness. He distinguishes between approximate reasoning (inferences with uncertain or imprecise premises) and plausible reasoning (patterns of reasoning which yield uncertain conclusions even when premises are certain).

A proposition is uncertain if its truth or falsity cannot be definitely established; a proposition is imprecise if the value of its contents is not sufficiently determined with respect to a given scale. Thus the truth of a proposition may be certain while its contents are imprecise, and conversely [138].

Prade discusses various theories of inexact inference. In probability theory, which assumes the law of non-contradiction, the probabilities of X and not X always sum to 1. When a conditional probability is certain (= 1), the probability of the condition (antecedent) is always less or equal to the the probability of the hypothesis (consequent).

When there is ignorance about the truth or falsity of a proposition, it may seem natural to treat the probabilities of both as = .5, But when there are more than two alternatives, ignorance is difficult to represent, since some propositions will paradoxically be more probable than others, even though ignorance about them is equally distributed [139].

Prade mentions Shafer's [140] solution to this problem: to introduce a belief function, whereby the credibility of a proposition is the weight of evidence in favour of it, and the plausibility = 1 - the credibility. Since opposing claims can both be plausible although neither is credible, lack of belief can be distinguished from belief [141].

When propositions form a nested entailment structure, credibility and plausibility functions are truth-functional, and reduce to necessity and possibility measures, respectively; these measures allow possibility to be a matter of degree, unlike standard modal logic. The uncertainty of a proposition is represented by a pair of numbers (its possibility and necessity measures), rather than by a single number as in probability theory [142].

Prade compares imprecision with fuzziness and specificity. Zadeh's [143] fuzzy sets are induced by vague predicates, while Yager's [144] specificity measure estimates the precision of information rather than its fuzziness.

Fuzzy sets are distinguished from possibility distributions, which measure the possibility that a proposition takes on a given value from among mutually exclusive values. Fuzzy set membership is non-exclusive, i.e., there can be overlap [145].

Prade shows that frequency histograms, usually rendered as probabilities, can also be interpreted as possibility distributions. The uncertainty of fuzzy events is measured by modelling vaguely described events with fuzzy sets, for which the laws of non-contradiction and excluded middle do not hold [146].

Prade treats degree of truth as a measure of conformity between a proposition and evidence for it; thus "..degree of truth..is relative to our state of knowledge." [147]. When the evidence is precise or the proposition is non-vague, the degree of truth measure is truth-functional. He shows that the MYCIN certainty factor can be viewed as a degree of truth in this sense [148].

He discusses two kinds of approximate reasoning: premises may be weighted or they may be imprecise or fuzzy. Premise weights can be interpreted as degrees of uncertainty (based on measures of probability, possibility, or necessity), or as degrees of truth (based on multivalued logic) [149].

Inference rules, such as modus ponens, are extended for both interpretations. Most fuzzy implication operators are special cases of a triangular norm or co-norm. It turns out that the patterns of the respective cases coincide. "Thus, from a practical point of view, the [two approaches] coincide; moreover, the differences which are introduced by the different implication functions or by the different measures of uncertainty which are used, are only nuances easy to compare, since the ordering of the main triangular norms induces an ordering on the [implication] operators" [150].

Discussion: analytical approach

The analytical approach has the merit of determining abstract relationships between different implication operators. This allows comparison of their behaviours in a general way; e.g. the Lukasiewicz operator leads to the least lower bound and greatest upper bound, and thus "..is the safest choice." [151].

However, Domain-dependent effects of inexactness on human reasoning are not reflected in this mathematical analysis; neither heuristic meta-reasoning nor context specific differences are taken into account.

The analytical approach is focused on systemic abstraction, to establish a general theoretical framework for inexact reasoning. Context-dependent differences in performance of operators are not considered, nor are the idiosyncracies of heuristic reasoning.

The main points of Prade's paper concern "the modelling of uncertainty and imprecision, and discussing various kinds of approximate and plausible reasoning schemes." [152]. In this case, many aspects of inference are rigorously analyzed, and terminology is carefully defined throughout.

2.2.4 Gaines [153] exemplifies another analytical approach to a theory of inexactness. He discusses Zadeh's development of fuzzy set theory as an approach to the formal mathematical treatment of imprecision. Zadeh chose Lukasiewicz multi-valued logic for the foundations of his system; in this logic the laws of non-contradiction and excluded middle are absent [154].

Gaines develops a weaker logic which is an axiomatized form of Lukasiewicz' infinitely-valued logic, but which is not truth-functional. This Standard Uncertainty Logic generically includes both probability logic and fuzzy logic as special cases. If the requirement of truth-functionality is added to the axioms, the result is Zadeh's fuzzy logic, while if either the law of excluded middle or the law of non-contradiction is added to the axioms, the result is a standard probability logic [155].

Gaines says this does not imply that the differences between the two logics are trivial, but rather that the differences do not lie primarily in the connectives used. "It is rather to the forms of the applications themselves that we should look for the interesting differences between the fuzzy and probability logics." [156].

Discussion: Gaines

The remarks concerning the analytical approach to a theory of inexactness in the section on Prade (2.2.3) apply equally to Gaines. Various aspects of inference are discussed in some detail, and Standard Uncertainty Logic is presented both model-theoretically and proof-theoretically.

2.2.5 Cheeseman [157] exemplifies the normative Baysian approach to a theory of inexactness. He believes that probability theory is the only way to represent and reason about inexact knowledge. He says "..knowledge is fundamentally an update process..", and that this is the fundamental viewpoint of Baysian inference. Conditional probability statements encode relevance between conditions and hypotheses, and thus allow updating of knowledge. Probability logic thus exhibits "non-monotonicity" [158].

Cheeseman concedes to logic a subservient role as an assistant to "internal" deductions, and says "Bayes' theorem is the modus ponens of probability theory.. When all probabilities are 0 or 1, probability reduces to logic." [159].

That probability is not truth-functional is a "major difference between probability and logic and means that it is usually necessary to make independence assumptions. Fortunately, it is a property of the world we live in that most events 704are 704 independent of other events.." [160].

Some other expressions of the normative Bayesian approach:

"..a rule is either true or not true..". "..if these [simplifying] alternatives are evaluated in scientific terms... then their success depends on their ability to approximate probabilistic calculations such as those invoking Bayes' rule" (McLeish [161]).

"..to optimize performance, any other approach either is equivalent to probability or else is inferior to it" (Rendell [162]).

Discussion: normative Bayesian approach

The normative Baysian approach (Cheeseman [163]) offers an established, mathematically sound method for some types of inexact reasoning. It enables belief updating through representation of context dependencies .

However, Pearl says the Baysian approach is "procedurally sterile" [164]; probabilities, interpreted as degrees of belief, can be calculated, but how to use them is not determined by a Baysian approach.

Moreover, as a normative prescription for how an ideally rational agent should reason, the Baysian approach contravenes the heuristic spirit of rule-based expert system applications. This approach, rather than simulating the way experts do think, would prescribe to them how they should think.

There is some controversy [165] over how widely applicable Baysian inference is to various types of inexactness.

The normative Baysian approach emphasizes standardization, that is, externally imposing uniformity on various types of applications, rather than abstracting a unity of structure which might underly the diversity of applications. This amounts to assuming at least one of the following:

There is a uniform structure common to all types of applications, and the form of this structure is appropriate for Baysian treatment. For instance, Cheeseman says [166] "fortunately most events in the real world ARE independent".

No matter how different various applications are, Baysian methods are sufficiently flexible to accomodate them all.

Cheeseman does not distinguish between inference rules and conditionals: "Bayes' theorem is the modus ponens (logical implication) of probability theory..", and sees logic as a special case of probability theory: " When all probabilities are 0 or 1, probability reduces to logic." [167]. He also speaks of using "..Bayesian inference (inductive and deductive).." [168]. However, Dubois and Prade object to these usages, since "Bayesian inference is NEVER deductive in our opinion. It is abductive since it always goes from observations to causes, and updates beliefs about causes on the basis of observations." [169].

2.2.6 Pearl [170] exemplifies a moderate Bayesian approach to a theory of inexactness. He examines "...the operational difference between the logical statement A --> B and its probabilistic counterpart P(B|A) = p." The logical statement constitutes "..a vey attractive, modular unit of computation, while the probabilistic statement ... is computationally sterile".

On the other hand, he contrasts "..the apparent inability of standard logic to express context-dependent information.." with probabilistic reasoning, "..a powerful language for expressing context-dependent beliefs". Pearl forsees the convergence of logical and probabilistic paradigms, through new semantic-preserving representations of the logical notions of dependence and relevance, and through new logics embodying the probabilistic notion of context dependence in deductive inference rules.

2.3 Summary

Each of the various approaches to selecting inexact inference methods is incomplete in some respect.

The heuristic approach is focused on domain-dependence, to accomodate idiosyncrasies of human reasoning. Common properties of reasoning over different domains are ignored; empirical evaluation of performance would be necessary to validate a purely heuristic system.

The empirical approach is focused on finding the best inference method for an application. Heuristic aspects of inexact reasoning are ignored, as are mathematical relationships between various operators.

The analytical approach is focused on systemic abstraction, to establish a general theoretical framework for inexact reasoning. Context-dependent differences in performance of inference methods are not considered, nor are the idiosyncracies of heuristic reasoning.

The normative Baysian approach focuses on standardization, that is, externally imposed uniformity on various types of application, ignoring their heuristic idiosyncrasies, rather than abstracting a unity of structure which might underly the diversity of applications.

the moderate Baysian approach allows for a complementary combination of probability theory and logic, thus at least implicitly addressing the issue of analytic comparison; it is focussed primarily on heuristics, in its recognition of context-dependence. However, it is silent on the issue of empirical validation.

3. RATIONALE AND METHOD

Section 3.1 contains a rationale for this study, and section 3.2 proposes the method employed in section 4.

3.1 Rationale

Analysis of the literature survey (section 2) indicates that a comparison of fuzzy and probabilistic inference is needed because:

3.1.1 There is a recognized need for inexact inference in rule processing applications. (but:)

3.1.2 There are no commonly accepted criteria for selecting an appropriate type of inexact inference for an application. (in part because:)

3.1.3 Investigators approach the issue from different theoretical frameworks. (and hence:)

3.1.4 Some terminology is used with different meanings by different investigators.

3.1.1 Need for inexact inference

Gaines & Shaw say "imprecision...is a necessary component of any system adequate to encode human knowledge." [171]. "Most of the distinctions we make in everyday life do not have hard boundaries." [172].

Whalen & Schott document the "..fuzziness in [expert's] rules.." and the nature of "..uncertainty in the data" [173]. Cohen & Grinberg distinguish "two types of uncertainty..One [arising] from noisy data..the other..associated with inference rules.." [174]. Prade [175] offers a fairly exhaustive categorization of different types of inexactness found in expert system applications.

3.1.2 Lack of criteria for applicability

Gaines & Shaw say fuzzy logic and probabilistic approaches are both "..expedients whose utility depends on the problem domain." [176]. Mamdani and Efstathiou mention the usual ad-hoc selection of inference procedures and the topic-dependence of multi-valued logical connectives [177].

Boose chose EMYCIN uncertainty factors primarily because "..most current systems use it." [178], but he gives no rationale for choosing his particular method of generating uncertainty factors from among the several methods available. Whalen & Schott attribute the different choices of implication operators "..partly to the preferences and arbitrary choices of system designers." [179].

3.1.3 Different approaches to inexact inference

The heuristic approach (Cohen & Grinberg [180]) focuses on the domain-dependent aspects of inexact inference, but does not distinguish different types of inference.

The empirical approach (Whalen & Schott [181]) addresses the necessity, in the absence of theoretical justification, to compare the performance of specific implication operators to determine the optimal one.

The analytical approach takes various forms; Prade [182] has an algebraic orientation, using T-norms; Garbolino [183] uses a modified Bayesian approach, while Gaines [184]) employs a lattice-theoretic approach.

The Baysian approach (Cheeseman [185]) relies upon presuppositions concerning the nature of the application (eg, independence of events).

3.1.4 Terminological variations

The various approaches to inexact inference are expressed with a corresponding diversity of terminology. Terms are used variously for different kinds of inexactness, and for different aspects of inference.

Terms variously express inexactness in data, in propositions about data, and in reasoning about data.

Data can be approximate, fuzzy, imprecise, missing, noisy, or uncertain [186]. Propositions can be ambiguous, credible, fuzzy, imprecise, inexact, plausable, possible, probable, uncertain or vague [187]. Reasoning can be approximate, credible, fuzzy, inexact, plausable, probable, or uncertain [188].

Aspects of inference include propositions, inference rule(s), and the logic used. Propositions can use connectives, such as "implication", to express logical relationships between other propositions; inference rules, such as modus ponens, specify conditions warranting the derivation of a proposition from others.

A sentence of the form **IF P THEN Q** is called an implication [189], a conditional [190], or a (production) rule [191], and is variously translated as [192]

P is a (sufficient) condition for Q
Q is a (necessary) condition of P
P implies Q
P entails Q
P causes Q
from P infer Q

Here are three terms and six translations for one form.

Moreover, the term **Modus Ponens** refers both to a particular sentence form and to an inference rule applying to sentences [193]:

[P & (P => Q)] => Q   sentence form

From **P** and **P => Q** infer **Q**   inference rule

The term **P** is called premise, antecedent, condition, cause, or evidence [194].

The term **Q** is called consequent, conclusion, hypothesis, or effect [195].

Baysians [196] tend to refer to the formal language for probabilistic reasoning as a calculus or a theory, contrasting it with logic; others [197] refer to it as probability logic.

3.2 Method

Fuzzy logic is compared with probability logic by interpreting a formal language L in a lattice M, and giving a basic truth-value assignment for the connectives of L in terms of lattice operations of meet and join; the lattice M is thus a model for L. The differences between probability logic and fuzzy logic are then shown to correspond to different models for L, that is, to different kinds of lattice.

This method also allows comparison of probability and fuzzy logics with Standard Uncertainty Logic (UL) and with the Sentential calculus (SC), which correspond to other kinds of lattice.

By presenting these comparisons in terms of a uniform representation, the mathematical theory of lattices, the confusions noted in section 3.1 are avoided: the comparisons are formally presented within an established theoretical framework with a precisely defined terminology.

Moreover, this context for comparison is neutral with respect to the issue at hand; there is no bias in favour of one or the other alternative logic, since lattice theory per se is just an abstract mathematical structure.

This method of comparison differs from Gaines' approach [198] in that he defines UL in terms of a lattice, then shows that probability and fuzzy logics are special cases of UL, by adding one or another condition to UL. Here various conditions are imposed upon the lattice (model), constraining its underlying structure, and then the various logics are derived from the ensuing properties of the model.

4 A LATTICE-THEORETIC COMPARISON OF SYSTEMS OF LOGIC

Overview:

Fuzzy and Probabilistic inference are compared by deriving them from alternative interpretations of a formal language. The interpretations are in lattices, and it is shown that fuzzy inference is derivable from an interpretation in a chain lattice, while probabilistic inference is derivable from an interpretation in a Boolean lattice. In either case, the basic truth-value assignments are the same, so the essential differences between the two types of logic can be represented by differences in the structure of the underlying interpretation.

Moreover, it is shown first that if the formal language is interpreted in a lattice which is modular, but not necessarily either a chain or Boolean, Standard Uncertainty Logic inference [199] is derivable; and finally, it is shown that if the interpretation is in a lattice which is both a chain and Boolean, the Sentential Calculus inference Modus Ponens is derivable.

In this chapter the sequence of results is presented without accompanying proofs, which are given in appendix 2.

Summary:

1 Present a formal language L.
2 Interpret L in a metric lattice M.
3 Assign truth-values for L in terms of a measure on M.

4 Derive truth-value assignments for UL.
5 Suppose M is a chain; derive value assignments for FL.
6 Suppose M is Boolean; derive value assignments for PL.
7 Suppose M is a Boolean chain; derive value assignments for SC.

4.0 Notation

As noted in section 1, and as noticed in section 2, the object language and metalanguage should be distinguished, to avoid confusion. Theorems for the object language (the interpreted formal language) are proven using the metalanguage, in this case abbreviated English. The following abbreviations are used:

'A ==> B' means 'if A then B'
'&&' means 'and'
'A && B ==> C' means '(A && B) ==> C'
'A iff B' means '(A ==> B) && (B ==> A)'

Note that '==>' and '&&' are abbreviations in the metalanguage, while '=>' and '&' are connectives in the object language.

Labels for expressions in definitions and theorems correspond to labels for the same expressions in proofs in the appendix, eg a1, a2, etcet.

4.1 The formal language L

A formal language L is presented, which can admit of various interpretations.

Let L = < S, -, &, |, =>, (,) >, where

S is a set of constants
- is a unary connective
&,|,=> are binary connective
(,) are punctuation symbols

Under the intended interpretations, the constants of S will be sentences, '-' will mean 'not', '&' will mean 'and', '|' will mean 'or', '=>' will mean 'implies', and the parentheses will serve as grouping symbols.

The grammatically valid, or well-formed, formulas (wffs) of L are all expressions built up by finitely many applications of rules 4-1-1 and 4-1-2:

4-1-1 Any constant in S is a wff of L.

4-1-2 Let p,q be wffs of L. Then p, (-p), (p & q), (p | q), (p => q), are wffs of L.

4-1-3 No other formulas are wffs.

Let W be the set of all wffs of L.

Under the intended interpretations, W will be a finite set of sentences, because the results are to be applied to expert system rulebases of the VRP sort, which always contain finitely many sentences.

4.2 Interpret L in a metric lattice M.

An equivalence relation on the set of wffs of L is used to partition the set into equivalence classes, which become elements in a metric lattice. To ensure the interpretations of L in the lattice are as intended, the equivalence relation is further constrained in terms of L connectives. (This approach is adapted from Donnellan [200], where however it is formulated axiomatically).

The actual order of exposition is the following:

2.1 Define an equivalence relation E on the set W.
2.2 Specify constraints on E in terms of L connectives.
2.3 Partition W into a set WW of equivalence classes.
2.4 Define a partial order [= on WW.
2.5 Show the E-classes in WW generate a lattice M for [=.
2.6 Define measure and distance functions on M.

4.2.1 Equivalence relation E

In the intended interpretations, individual wffs of L cannot be used as lattice elements, because the partial order of the lattice requires a previous definition of equality, but individual wffs are distinct objects [201].

Moreover, in applying the interpretations to VRP type inference, it is the truth-value of sentences, not their content, which is of interest. Thus the lattice elements should be equivalence classes of sentences, where members of the same class are intended to have the same value.

The equivalence relation serves as the definition of equality for the partial order of the lattice, and to establish a connection between individual wffs of L and their equivalence classes, a quasi-order (which does not presuppose equality) is used [202]. Thus an equivalence relation is defined in terms of a quasi-order Q on the set W of wffs of L.

Define the binary relation Q on W:

for every x, y, z in W,

a1 xQx reflexive
a2 xQy && yQz ==> xQz transitive

Then Q is a quasi-order on W [203].

Theorem T1: Let E be a binary relation on W such that

for every x, y in W,

a3 xEy iff xQy && yQx

Then E is an equivalence relation: for every x,y,z in W,

xEx reflexive
xEy ==> yEx symmetric
xEy && yEz ==> xEz transitive

4.2.2 Constraints on Q and E in terms of connectives of L:

L is to be interpreted in a metric lattice. To ensure that the connectives behave as intended, further constraints are placed on Q and E in terms of connectives of L: For every x,y,z in W

a4 (x&y)Qx, (x&y)Qy, zQx && zQy ==> zQ(x&y) glb of x,y
a5 xQ(x|y), yQ(x|y), xQz && yQz ==> (x|y)Qz lub of x,y
a6 x&y E x iff xQy, x|y E y iff xQy consistency
a7 x&x E x, x|x E x idempotent
a8 x&y E y&x, x|y E y|x commutative
a9 x&(y&z) E (x&y)&z, x|(y|z) E (x|y)|z associative
a10 x&(x|y) E x, x|(x&y) E x absorptive

Constraints a4 and a5 guarantee the interpretation to be in a lattice, and constraints a6 - a10 ensure the complete algebraic characterization of the lattice.

4.2.3 Partition W into a set of equivalence classes

Let E partition W into classes X, Y, Z, ... of equivalent elements: For every x in X, y in Y,

a11 xEy iff X = Y

Let WW = { X, Y, Z, ... }.

The alternate notation [x] = X is also used.

4.2.4 Define a partial order [= on WW.

A partial order [= is defined on WW, and it is shown that the E classes in WW form a lattice M under [=.

Theorem T2: define the binary relation [= on WW:

a12 for every x in X and y in Y, X [= Y in WW iff xQy in W

Then X [= Y is a partial order on WW:

X [= X (reflexive)
X [= Y && Y [= X ==> X = Y (antisymmetric)
X [= Y && Y [= Z ==> X [= Z (transitive)

4.2.5 The lattice M formed by WW under [=.

The set WW of E equivalence classes partially ordered by [= forms a lattice M with meet A and join V:

a13 For every x,y in W, [x]A[y] = [x&y], [x]V[y] = [x|y]

a14 For every X,Y in WW, there exist P,Q in WW such that

i P = XAY = [x&y], P [= X {4-1-2, a4, a11-a13} glb of X,Y
ii Q = XVY = [x|y], Y [= Q {4-1-2, a5, a11-a13} lub of X,Y

By a12-a14, M is a lattice [204]. Then the complete algebraic characterization of M [205] follows from a6 - a11 and a13: For every X, Y, Z in WW,

a15 X A X = X, X V X = X idempotent

a16 X A Y = Y A X, X V Y = Y V X commutative

a17 X A (Y A Z) = (X A Y) A Z associative
X V (Y V Z) = (X V Y) V Z

a18 X A (X V Y) = X V (X A Y) = X absorptive

a19 X [= Y iff XAY = X consistency
X [= Y iff XVY = Y

Since there are only a finite number of wffs of L, W, and hence WW, is finite. Consequently WW has a least element and a greatest element: for every X in WW there exist F, T such that [206]

a20 F [= X [= T

a21 i) F = FAX, ii) X = FVX; F is least element
iii) T = TVX, iv) X = TAX T is greatest element

4.2.6 Define measure and distance functions on lattice M.

A measure function m is defined on the lattice M, which accordingly is modular [207]; this allows definition of implication and complementation in terms of meet and join, even though M is not complemented. Implication and complementation are defined in terms of the distance between elements.

Define the measure function m on M, m:WW --> [0,1] such that for T,F and every X,Y in WW

a22 m(T) = 1, m(F) = 0

a23 m(X) <= m(Y) iff X [= Y

a24 m(X A Y) + m(X V Y) = m(X) + m(Y) additivity

Then M is a metric lattice, hence modular [208].

For a metric lattice the distance d between X and Y is defined as [209]: for every X,Y in WW:

a25 d(X,Y) = m(X V Y) - m(X A Y)

4.3 Assign truth-values for L in terms of the measure m.

A valuation function v is defined in terms of the measure m; v is used for basic truth-value assignments to the connectives of L, in terms of meet (A) and join (V) of the lattice M. It is intended that 'and' (&) corresponds to meet 'A', 'or (|) corresponds to join 'V'.

Implication '=>' and complementation '-' are given value assignments in terms of the distance function d, for motivation as explained below.

Define the valuation function v on W: v:W --> [0,1] such that for every X,Y in WW, s,t in W, x in X, y in Y,

a26 v(s) = m(X) iff s is in X
a27 v(s & t) = m(X A Y) iff s is in X and t is in Y
a28 v(s | t) = m(X V Y) iff s is in X or t is in Y
a29 v(x => y) = 1 - d(X, X A Y) = 1 - d(Y, X V Y)
a30 v(-x) = 1 - d(X, F)

The definition for implication (a29) is motivated by noticing [210] that in the Sentential Calculus (SC), when v(x => y) = 1, v(x) = v(x & y) and v(y) = v(x | y):

v(x)	v(y)	v(x => y)	v(x & y)	v(x | y)
1	1	1	1	1
1	0	0	0	1
0	1	1	0	1
0	0	1	0	0

Accordingly, v(x => y) is defined in terms of the distance between [x] and [x&y] (also between [y] and [x|y]; this part of the definition is stated without proof, since it is not used - its consequences are derived from the first part of the definition, for theorem T4).

The definition for complementation (a30) follows the same form; in SC, when v(-x) = 1, v(x) = 0:

Accordingly, v(-x) is defined in terms of the distance between [x] and F.

4.4 Derive truth-value assignments for UL

It is shown that the truth-value assignments for UL, as presented in 1-5-18, can be derived from the basic truth-value assignments a26 - a30, under the assumption that the lattice M is modular.

Truth-value assignments for UL: For every x,y in W,

UL1 0 <= v(x) <= 1
UL2 v(x & y) <= min[v(x), v(y)]
UL3 v(x | y) >= max[v(x), v(y)]
UL4 v(x => y) = 1 - v(x) + v(x & y)
UL5 v(-x) = 1 - v(x)

Theorem T3: For every x,y in W, x in X, y in Y,

i v(x & y) <= min[v(x), v(y)] (UL2)
ii v(x | y) >= max[v(x), v(y)] (UL3)

Theorem T4: For every x,y in W, x in X, y in Y,

i v(x => y) = 1 - v(x) + v(x & y) (UL4)
ii v(x => y) = 1 + v(y) - v(x | y)

Theorem T5: For every x in X,

v(-x) = 1 - v(x) (UL5)

4.5 Let M be a chain; derive truth-value assignments for FL.

It is shown that the truth-value assignments for FL, as presented in 1-4-11 to 1-4-15, can be derived from the basic truth-value assignments a26 - a30, under the assumption that the lattice M is a chain.

Truth-value assignments for FL: For every x,y in W,

FL1 0 <= v(x) <= 1
FL2 v(x & y) = min[v(p), v(q)]
FL3 v(x | y) = max[v(p), v(q)]
FL4 v(x => y) = 1 - v(x) + v(x & y)
FL5 v(-x) = 1 - v(x)

FL1, FL4, and FL5 have been proved in section 4; they correspond to 1-4-11, 1-4-15, and 1-4-12.

Theorem T6: If M is a chain, then for every x, y in W,

i v(x & y) = min[v(x), v(y)] (FL2)
ii v(x | y) = max[v(x), v(y)] (FL3)

FL2 and FL3 correspond to 1-4-13 and 1-4-14.

4.6 Let M be Boolean; derive truth-value assignments for PL.

It is shown that the truth-value assignments for PL, as presented in Rescher [211], can be derived from the basic truth-value assignments a26 - a30, under the assumption that the lattice M is Boolean, ie, is complemented and distributive. (The PL conditional is denoted by "=)").

Truth-value assignments for PL: For every x,y in W,

PL1 0 <= v(x) <= 1
PL2 v(x | -x) = 1
PL3 v(x | y) = v(x) + v(y), if v(x & y) = 0
PL4 v(x) = v(y), if v(x & -y) = 0 and v(-x & y) = 0
PL5 v(x =) y) = v(-x | y)

Theorem T7: If M is Boolean, then for every x in W,

i v(x & -x) = 0
ii v(x | -x) = 1 (PL2)

Theorem T8: for every x,y in W,

if v(x & y) = 0 then v(x | y) = v(x) + v(y) (PL3)

Theorem T9: If M is Boolean, then for every x,y in W,

if v(x & -y) = 0 and v(-x & y) = 0 then v(x) = v(y) (PL4)

Theorem T10: If M is Boolean, then for every x,y in W,

v(x =) y) = v(-x | y) = v(x => y)

4.7 Let M be a boolean chain; derive truth-value assignments for SC

It is shown that the truth-value assignments for SC, as presented in 1.2.2, can be derived from the basic truth-value assignments a26 - a30, under the assumption that the lattice M is a Boolean chain.

Truth-value assignments for SC: For every x,y in W, and v --> {0,1},

SC1 v(-x) = 1 iff v(x) = 0
v(-x) = 0 iff v(x) = 1

SC2 v(x & y) = 1 iff v(x) = 1 and v(y) = 1
v(x & y) = 0 iff v(x) = 0 or v(y) = 0

SC3 v(x | y) = 1 iff v(x) = 1 or v(y) = 1
v(x | y) = 0 iff v(x) = 0 and v(y) = 0

SC4 v(x => y) = 1 iff v(-x) = 1 or v(y) = 1
v(x => y) = 0 iff v(-x) = 0 and v(y) = 0

For theorems T11 through T15, assume M is a Boolean chain.

Theorem T11: for every x in W,

v(x) = 0 or v(x) = 1

Theorem T12: for every x in W,

v(-x) = 1 iff v(x) = 0 (SC1)
v(-x) = 0 iff v(x) = 1

SC1 corresponds to 1-2-4.

Theorem T13: for every x,y in W,

v(x & y) = 1 iff v(x) = 1 and v(y) = 1 (SC2)
v(x & y) = 0 iff v(x) = 0 or v(y) = 0

SC2 corresponds to 1-2-5.

Theorem T14: for every x,y in W,

v(x | y) = 1 iff v(x) = 1 or v(y) = 1 (SC3)
v(x | y) = 0 iff v(x) = 0 and v(y) = 0

SC3 corresponds to 1-2-6.

Theorem T15: for every x,y in W,

i v(x => y) = 1 iff v(-x) = 1 or v(y) = 1 (SC4)
ii v(x => y) = 0 iff v(-x) = 0 and v(y) = 0

SC4 corresponds to 1-2-8.

5 IMPLEMENTATION CONSIDERATIONS

It is shown how the results of section 4 can be implemented as inference procedures for processing VRP rulebases. Subsections are:

5.1 VRP multivalued inference procedure.

In section 1.1 a simplified account of the VRP version 1 inference procedure was presented, as an example of a common kind of rule processing. In this section some details of VRP multivalued inference are presented, as background for example implementations of the various systems of logic of section 4.

5.1.1 VRP evidential weights and coweights

During processing, VRP maintains two values for each sentence, a weight and a coweight. The standard interpretation is that a weight represents a degree of affirmation, while a coweight represents a degree of rejection, of the sentence. In addition, each rule has a strength, which limits its potential contribution to the weight/coweight of the consequent. Other interpretations may be more appropriate, depending on the application; the developer must decide what these factors represent and what their values mean, for a given application [212].

5.1.2 Confidence and uncertainty

The weight and coweight of a sentence range beyween 0 and 100. During VRP processing, it is required that

5-1 weight + coweight <= 100

The uncertainty of a sentence is defined as:

5-2 uncertainty = 100 - weight - coweight

Intuitively, the uncertainty of a sentence during processing represents the maximum total amount of affirming and refuting evidence which can still be accumulated for the sentence during subsequent processing.

The confidence for a sentence is defined as:

5-3 confidence = weight - coweight

The confidence for a sentence represents how much the affirming evidence outweighs (or falls short of) the refuting evidence. By convention, a confidence factor of +100 represents total affirmation, while a confidence factor of -100 represents total refutation [213].

5.1.3 Combination and propagation of weights

This section describes how VRP combines and propagates weights during a consultation; the description is simplified by ignoring coweights.

The inequality of 5-9 can be used by a function which reports results of rule processing, which could display something like

The weight of "q" is at least n   [$n = s + wt(p) - 100$]

and/or

The weight of "NOT q" is at most m [$m = 200 - s - wt(p)$]

Note that in this implementation VRP uncertainty is always 0: by UL5 (section 4.4),

UL5 $v(-p) + v(p) = 1$

so by UL5, 5-4 and 5-5,

5-12 $wt(p) + cwt(p) = 100$

Thus by 5-2 and 5-12, VRP uncertainty = 100 - 100. However, UL uncertainty is represented by inequalities 5-9 and 5-11; since $wt(q) <= 100$, $s + wt(p) - 100 <= wt(q) <= 100$. Thus UL uncertainty Uul can be interpreted as

$Uul = 100 - (s + wt(p) - 100) = 200 - s - wt(p)$

Intuitively: for a given premise weight, the greater the rule strength the more certain the conclusion; for a given rule strength, the greater the premise weight the more certain the conclusion.

By 5-3, VRP confidence is still computable, eg:

wt(p)	cwt(p)	confidence in p
100	0	100
75	25	50
50	50	0
25	75	-50
0	100	-100

5.2.2.2 Combination and propagation of (co)weights

When more than one VRP rule has the same consequent, say

R1: IF p1 THEN (s1) q

R2: IF p2 THEN (s2) q

this represents a single rule with a disjunctive antecedent

IF p1 OR p2 THEN (s) q

provided that the rule strengths are the same: s1 = s2 = s. (VRP does not support an explicit disjunction operator OR). VRP's method of combining weights for the consequent in such a case is to sum them (5.1.3). But for UL,

v(p1 | p2) >= max[v(p1), v(p2)].   [by UL3 in 4.4]

Thus, when s is the same for both rules, a VRP simulation of UL assigns at least the max of the antecedent weights of the two rules to the weight of their combined antecedents.

Then by 5-9 and UL3,

5-13 wt(q) >= s + max[wt(p1), wt(p2)] - 100.

The inequality ">=" in 5-13 can be used by a function which reports results of rule processing, which could display something like

The weight of "q" is at least n

Where n = s + max[wt(p1), wt(p2)] - 100.

5.3 Implementing Fuzzy Logic (FL)

The inference rule Fuzzy Modus Ponens (1-4-16) and the VRP representations of 5-4 to 5-6 give a VRP FL inference procedure ("wf(q)" denotes resulting weight of consequent):

5-14 when s < 100, wf(q) = wt(p) + s - 100;
when s = 100, wf(q) >= wt(p)

Both VRP and FL uncertainties are 0, except when s = 100; then FL uncertainty Ufl is

Ufl = 100 - wt(p) = cwt(p)   [by 5-2, 5-10, 5-12, 5-14]

By 5-3, VRP confidence is still computable (see 5.2.2.1).

5.4 Implementing Probability Logic (PL)

A PL Modus Ponens can be derived from

v(p =) q) = v(-p|q) >= max(v(-p),v(q)) [PL5 4-6, UL3 4-4]

Case 1: v(q) = max(v(-p), v(q)), v(p =) q) >= v(q)
Case 2: v(-p) = max(v(-p),v(q)), v(q) <= v(-p) <= v(p =) q)

In both cases, v(q) <= v(p =) q) = v(p => q) [T10, p.83] so ("wp(q)" denotes resulting weight of consequent):

5-15 wp(q) <= s [5-4, 5-6]

But also the UL inference procedure applies, since PL is a special case, so

5-16 s + wt(p) - 100 <= wp(q) <= s [5-9, 5-15]

Since wp(q) can have a value between s + wt(p) - 100 and s, PL uncertainty Upl can be represented by

s - (s + wt(p) - 100) = 100 - wt(p) = cwt(p) [5-16, 5-10].

5.5 Summary and discussion

A comparison of results of inference for PL and FL under the proposed VRP implementation is given in the table:

s	wp(q)	Upl	wf(q)	Ufl
100	wt(p) to 100	cwt(p)	wt(p) to 100	cwt(p)
<100	wt(p)+s-100 to s	cwt(p)	wt(p)+s-100	0

When rule strength = 100 there is no difference in the results; when rule strength is less than 100, wp(q) is between wf(q) and s:

wf(q) <= wp(q) <= s, s < 100

Although FL and PL exhibit different properties vis-a-vis truth-functionality and exclusionality, the results of their inference procedures are not that different. For rules regarded as axiomatic (strength = 100), the results are the same; for other rules the only differences are that FL assigns a specific value to a consequent, hence has no uncertainty, while PL assigns a range of possible values, between the FL value and the rule strength.

These results echo Prade's finding that "..from a practical point of view the [two approaches] coincide..", and that there is an ordering on the implication operators [section 2.2.3].

From a computational point of view the PL inference procedure would be marginally faster than that for FL, since the latter requires a comparison of s with 100. PL also offers more flexibility in choosing an inference policy: a conservative policy could be implemented by using the lower limit of the range of wp(q) as the value for wp(q); a liberal policy could use the upper limit (rule strength); or a moderate policy could use an intermediate point in the range.

On the other hand, if a conservative policy were chosen for FL for the case s = 100, there would be no need to compare s with 100, so PL would have no computational advantages over FL, which would then also be conceptually simpler. In light of Boose's experience that domain experts prefer fewer intervals for rating inexactness [section 2.1.3], in a practical application it may be just as well to use FL.

REFERENCES AND NOTES

Section 1

1. VRP User Guide.
2. ESE, VPEXPERT.
3. Personal experience as developer of sundry VRP rulebases.
4. In actuality, a confidence factor is computed from the weight and coweight, which are both initialized to 0 (see section 5).
5. Kleene, p. 33.
6. Kleene, op. cit., pp. 11 - 33.
7. Dempster, p. 72.
8. Adapted from Enderton, pp. 17-20.
9. Kleene, op. cit., p. 12.
10. Adapted from Enderton, op. cit., p. 31.
11. Kleene, op. cit., p. 70.
12. Pedersen, pp. 36-40.
13. Bandler & Kohout, p. 95.
14. Mendelson, p. 40.
15. Kleene, op. cit., p. 34.
16. Easily verified by truth tables.
17. Kleene, op. cit., p. 3.
18. Rescher, p. 21.
19. Documented in section 2.1.
20. Leblanc, vii-viii.
21. See, eg, Suppes.
22. A non-standard approach, but used here for accord with subsequent development; adapted from Pal, et al.
23. Suppes, op. cit.
24. Ibid.
25. Hoel & Jessen, pp. 48-54.
26. Adapted from Breiman, p. 6.

27. Morgan, pp. 95-6.
28. Breiman, op. cit., p. 127.
29. Pearl, p. 100.
30. Ibid.
31. Rescher, op. cit., pp. 184-5.
32. Pearl, op. cit., p. 100.
33. Dubois & Prade, p. 74.
34. Ibid.
35. Dong & Wong, p. 564.
36. Zadeh ('65).
37. Gaines & Shaw ('87), p. 7.
38. Pal et al, p. 40.
39. Ibid, p. 42.
40. LeBlanc, pp. 9-13, 76-8.
41. Ibid, pp. 76-8.
42. Adapted from Ibid.
43. Pal et al, p. 43.
44. Haack, p. 74.
45. Pal et al, op. cit., p. 43.
46. Zadeh, pp. 257-8.
47. Gaines ('76), pp. 641-2.
48. Gaines ('83), p. 125.
49. Ibid, p. 120-3.
50. Donnellan, pp. 43-52.
51. Birkhoff, p. 1.
52. Ibid, p. 16.
53. Abbott, p. 109.
54. Donnellan, op. cit., pp. 38, 69.
55. Ibid, p. 101.
56. Ibid, p. 38, 69.
57. Ibid.
58. Ibid, p. 202.
59. Ibid, p. 224.
60. Birkhoff, op. cit., pp. 74,76.

61. Gaines ('83), op. cit., pp. 121-2.
62. Birkhoff, op. cit., p. 76.
63. Ibid.
64. Gaines ('83), op. cit., p. 122.
65. Ibid.
66. Follows from a24 - a28 in appendix 2.
67. Gaines ('83), op. cit., p. 125.

Section 2

68. Gaines & Shaw ('85).
69. Ibid, p. 8.
70. Ibid.
71. Ibid.
72. Ibid, p. 9.
73. Gaines ('83).
74. Gaines & Shaw ('85), p. l2.
75. Ibid, p.l2.
76. Mamdani & Efstathiou ('84).
77. Ibid, p. 218.
78. Ibid, p. 219.
79. Ibid, p. 214.
80. Ibid, p. 225.
81. Ibid, p. 226.
82. Boose ('85).
83. Ibid, p. l58.
84. Ibid.
85. Ibid, p. l59.
86. Ibid, p. l60.
87. Ibid, p. l62.
88. Ibid, p. l82.
89. Ibid, p. l85.
90. Ibid, p. 200.
91. Ibid, p. ll5.

92. Whalen & Schott ('83).
93. Ibid, p.59.
94. Ibid, p. 58.
95. Ibid, pp. 64-65.
96. Ibid, p. 59.
97. Ibid, pp. 65-66.
98. Ibid, p. 59.
99. Ibid, p. 66.
100. Ibid, p. 68.
101. Ibid.
102. Ibid, p. 69.
103. Ibid, p. 70.
104. Shafer ('76).
105. Barnett, p. 871.
106. Gordon & Shortliffe, p. 330.
107. Ibid, p. 331.
108. Barnett, ('81).
109. Ibid, pp. 874-5.
110. Gordon & Shortliffe, op. cit.
111. Ibid, p. 348.
112. Ibid, p. 349.
113. Ibid, p. 351.
114. Dong & Wong, pp. 551-565.
115. Ibid, p. 565.
116. Ibid, p. 564.
117. Ibid, p. 565.
118. Gaines ('85).
119. Ibid, pp. 8-9.
120. Boose ('85), op. cit.
121. Cohen & Grinberg.
122. Ibid, p. 17.
123. Ibid, p. 18.
124. Ibid, p. 19.
125. Ibid, p. 21.

126. Ibid.
127. Ibid, p. 22.
128. See 111.
129. Shaw, pp. 1-2.
130. Benfer & Furbee, pp. 22-3.
131. Whalen & Schott ('85).
132. Ibid, pp. 331-3.
133. Ibid, p. 340.
134. Ibid, p. 344.
135. see 121.
136. Ibid, pp. 328-9.
137. Prade ('85).
138. Ibid, pp. 260-261.
139. Ibid, p. 261.
140. Ibid.
141. Ibid.
142. Ibid, p. 262.
143. Ibid.
144. Ibid.
145. Ibid, p. 263.
146. Ibid, pp. 263-4.
147. Ibid, pp. 264-6.
148. Ibid, p. 268.
149. Ibid, p. 269.
150. Ibid, p. 270.
151. Ibid, p. 272.
152. Ibid, p. 260.
153. Gaines ('83), op. cit.
154. Ibid, p. 124.
155. Ibid, pp. 124-5.
156. Ibid, p. 125.
157. Cheesman, pp. 58-66, pp. 129-142.
158. Ibid, p. 59.
159. Ibid, p. 60.

160. Ibid, p. 131.
161. McLeish, p. 93.
162. Rendell, p. 103.
163. Cheesman, op. cit.
164. Pearl, op. cit., p. 99.
165. See "taking issue/forum", Computational Intelligence.
166. Cheesman, op. cit.
167. Ibid, p. 60.
168. Ibid, p. 158.
169. Dubois & Prade, pp. 75-6.
170. Pearl, op. cit., p. 199.

Section 3

171. Gaines & Shaw ('85), p. 8.
172. Ibid.
173. Whalen & Schott ('83), pp. 60-63.
174. Cohen & Grinberg, p. 17.
175. Prade ('85).
176. Gaines & Shaw ('85), op. cit.
177. Mamdani & Efstathiou, p. 213.
178. Boose, p. 1.
179. Whalen & Schott ('83), op. cit., P. 70.
180. Cohen & Grinberg ('83), op. cit., p. 17.
181. Whalen & Schott ('83), op. cit., pp. 60-63.
182. Prade ('85), op. cit.
183. Garbolino, pp. 709-715.
184. Gaines ('83), op. cit.
185. Cheeseman, op. cit.
186. See 163, 164.
187. See 162, 172.
188. See 5, 172, 174.
189. See 5.
190. See 31.

191. See 120.
192. See 5.
193. Ibid.
194. Ibid.
195. Ibid.
196. See 147.
197. See 31.
198. Gaines ('83), op. cit.

Section 4.

199. As in section 1.5.2.2
200. Donnellan, op. cit., pp. 271-6.
201. Ibid, p. 271.
202. Ibid.
203. Ibid, p. 264.
204. Birkhoff, op. cit., p. 16.
205. Ibid, p. 18.
206. Donnellan, op. cit., pp. 53-5.
207. Birkhoff, op. cit., p. 76.
208. Ibid.
209. Ibid.
210. Gaines ('83), op. cit., p. 122.
211. Rescher, op. cit., p. 185.

Section 5

212. Akin & Reuter, Refor Rulebase Guide, p. 6.
213. Ibid, pp. 6-7.
214. Ibid, Edatopi Rulebase Notes, p. 1.
215. Ibid.
216. Ibid, Rulebase Guide, p. 1.
217. VRP User Guide, pp. 51 - 54.
218. Confirmed by Yimin Shen, VRP II compiler author, personal communication, 3-23-90.
219. Ibid.

Proofs

220. Donnellan, op. cit., p. 75.
221. Birkhoff, op. cit., p. 10.
222. Donnellan, op. cit., p. 224.
223. Ibid.

BIBLIOGRAPHY

Abbott, James C. **Sets, Lattices, and Boolean Algebras.** Allyn and Bacon, Inc. Boston; l969.

Akin, G. & Reuter, L. **Forest Regeneration Assistant User Guide.** Autometrics Associates Ltd. Victoria BC; l988.

Barnett, J. A. "Computational methods for a mathematical theory of evidence" in Proceedings 7th International Joint **Conference on Artificial Intelligence.** Vancouver, BC; 1981.

Birkhoff, Garret. **Lattice Theory.** American Mathematical Society Publications Volume XXV. Providence, RI; l964.

Boose, J.H. **Personal Construct Psychology and the Transfer of Human Expertise.** Artificial Intelligence Center, Advanced Technology Applications Division, Boeing Computer Services. Report BCS-62010-1, Jan l985.

Breiman, Leo. **Probability and Stochastic Processes.** Houghton Mifflin Company. Boston; l969.

Cheeseman, Peter. "An inquiry into computer understanding." in **Computational Intelligence.** National Research Coucil Canada. Vol 4, No. 1; l988.

Cohen, P.R. & Grinberg, M.R. "A theory of heuristic reasoning about uncertainty." In **The AI Magazine,** Summer l983.

Dong, W. & Wong, F. "Propagation of evidence in rule based systems." In **International Journal of Man-Machine Studies** (l987) 26.

Donnellan, Thomas. Lattice Theory. Pergamon Press. Oxford; l963.

Dubois, D. & Prade, H. "Comments on An inquiry into computer understanding." in **Computational Intelligence** National Research Council Canada. Vol 4, Number 1; l988.

Dukarm, J. & Akin, G. **VRP User Guide.** Autometrics Associates Ltd. Victoria BC; l988.

Enderton, Herbert B. **A Mathematical Introduction to Logic.** Academic Press, Inc. New York; l972.

Gaines, B.R. (83). "Precise past - fuzzy future." In **International Journal of Man-Machine Studies** (l983) l9.

Gaines, B.R. & Shaw, M.L.G. "Foundations of expert systems." in Kacprzyk, J. & Yager, R.R., Eds. **Management Decision Support Systems Using Fuzzy Sets and Possibility Theory.** ISR Interdisciplinary Systems Research Verlag TUV Rheinland; l985.

Garbolino, Paolo. "A comparison of some rules for probabilistic reasoning." in **International Journal of Man-Machine Studies.** Academic Press Limited. Vol 27; l987.

Gordon, J. & Shortliffe, E. H. "A method for managing evidential reasoning in a hierarchical hypothesis space." in **Artificial Intelligence.** Elsevier Science Publishers B. V. (North Holland); Vol 26, 1985.

IBM Corp. **ESE Concepts and Facilities**. Report Y6067; l988.

Kleene, Stephen Cole. **Mathematical Logic.** John Wiley & Sons, Inc. New York; l967.

Leblanc, Hugues. **Statistical and Inductive Probabilities.** Prentice-Hall, Inc. Englewood Cliffs, NJ.; l962.

Mamdani, E. & Efstathiou, J. "An analysis of formal logics as inference mechanisms in expert systems." **International Journal of Man-Machine Studies** Academic Press Ltd. l984 21.

McLeish, D.L. "In praise of Bayes." in **Computational Intelligence.** National Research Council Canada. Vol 4. Number 1; l988.

Mendelson, Elliott. **Introduction to Mathematical Logic.** D. Van Nostrand Company, Inc. New Jersey; l964.

Morgan, Charles G. "Probability theory versus procedural pessimism." in **Computational Intelligence.** op cit.

Pal, Sankar K. & Majumder, Dwijesh K. Dutta. **Fuzzy Mathematical Approach to Pattern Recognition.** John Wiley & Sons. New York; l986.

Pearl, Judea. "On Logic and probability." in **Computational Intelligence.** op cit.

Pederson, K. "Well-structured Knowledge Bases, Part III," in **AI Expert Magazine.** Nov. l989.

Prade, H. "A computational approach to approximate and plausible reasoning with applications to expert systems." In **IEEE Transactions on Pattern Analysis and Machine Intelligence.** Vol. PAMI-7. no. 3. May l985.

Rendell, Larry. "Reply to An inquiry into computer understanding." in **Computational Intelligence.** op cit.

Rescher, Nicholas. **Many-valued Logic.** McGraw-Hill Book Company. New York; l969.

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

Suppes, P. **Axiomatic Set Theory.** Van Nostrand Co. Princeton; l960.

VP Expert User Guide. Paperback Software Inc. San Francisco; l988.

Whalen, T. & Schott, B. "Issues in fuzzy production systems." In **International Journal of Man-Machine Studies** (l983) l9.

Whalen, T. & Schott, B. "Alternative logics for approximate reasoning in expert systems: a comparative study." In **International Journal of Man-Machine Studies** (l985) 22.

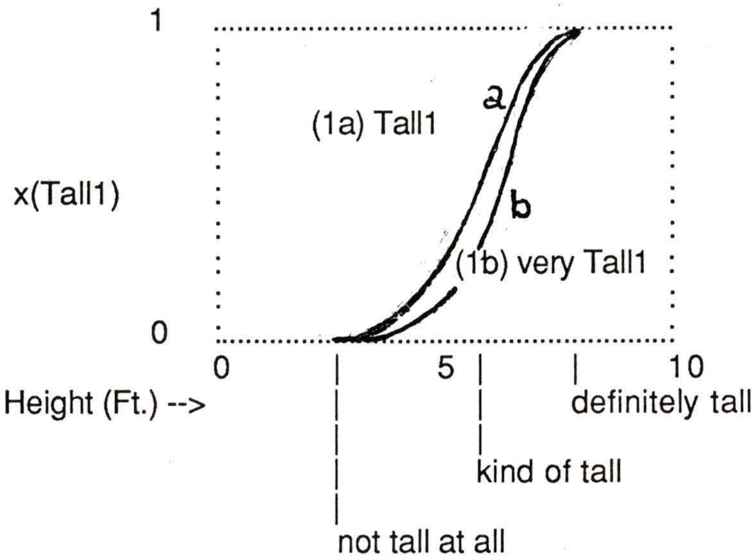
Zadeh, L. "A Computational Approach to Fuzzy Quantifiers in Natural Languages." In **Computation & Mathematics with Applications.** l983 Vol 9, no. 1.

Zadeh, L. "The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems." **In Fuzzy Sets and Systems.** 11. North-Holland; l983.

Appendix 1

Figures

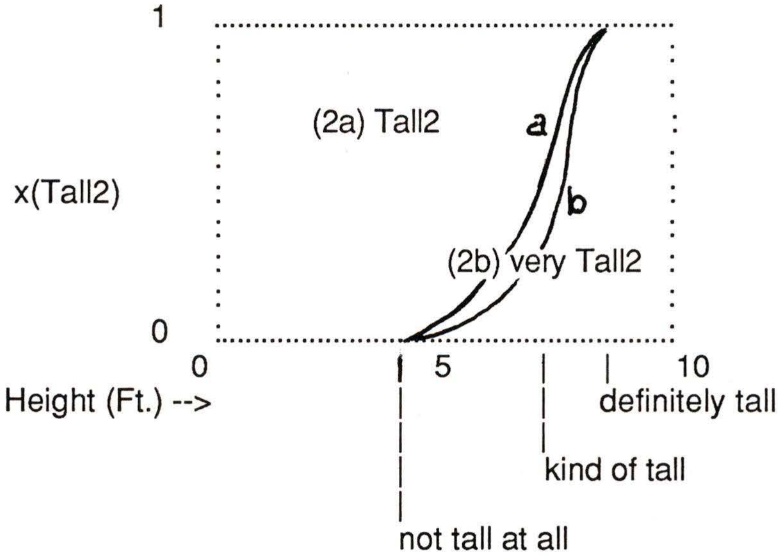
Figure 1



x(Tall1) = degree of membership of x in the fuzzy set Tall1

Tall1 represents one person's construal of tallness

Figure 2



x(Tall2) = degree of membership of x in the fuzzy set Tall2

Tall2 represents the average of a small sample of people's construals of tallness

Figure 3

v(p)

v(q)	0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1
0	1	.9	.8	.7	.6	.5	.4	.3	.2	.1	0
.1	1	1	.9	.8	.7	.6	.5	.4	.3	.2	.1
.2	1	1	1	.9	.8	.7	.6	.5	.4	.3	.2
.3	1	1	1	1	.9	.8	.7	.6	.5	.4	.3
.4	1	1	1	1	1	.9	.8	.7	.6	.5	.4
.5	1	1	1	1	1	1	.9	.8	.7	.6	.5
.6	1	1	1	1	1	1	1	.9	.8	.7	.6
.7	1	1	1	1	1	1	1	1	.9	.8	.7
.8	1	1	1	1	1	1	1	1	1	.9	.8
.9	1	1	1	1	1	1	1	1	1	1	.9
1	1	1	1	1	1	1	1	1	1	1	1

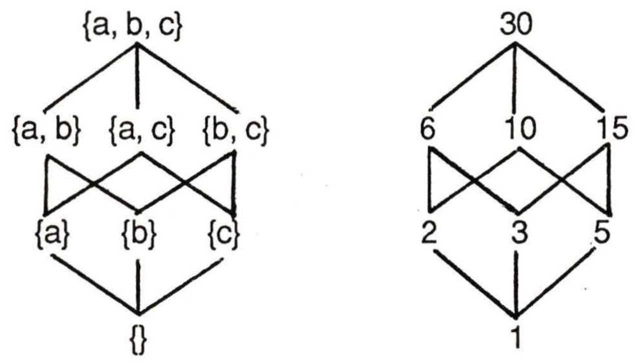
Truth-table for Lukasiewicz implication operator

Figure 4

v(q) \ v(p)	0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1
0	0	0	0	0	0	0	0	0	0	0	0
.1	1	1	.5	.3	.3	.2	.2	.1	.1	.1	.1
.2	1	1	1	.7	.5	.4	.3	.3	.3	.2	.2
.3	1	1	1	1	.8	.6	.5	.4	.4	.3	.3
.4	1	1	1	1	1	.8	.7	.6	.5	.4	.4
.5	1	1	1	1	1	1	.8	.7	.6	.6	.5
.6	1	1	1	1	1	1	1	.9	.8	.7	.6
.7	1	1	1	1	1	1	1	1	.9	.8	.7
.8	1	1	1	1	1	1	1	1	1	.9	.8
.9	1	1	1	1	1	1	1	1	1	1	.9
1	1	1	1	1	1	1	1	1	1	1	1

Truth-table for Gaines 43 implication operator

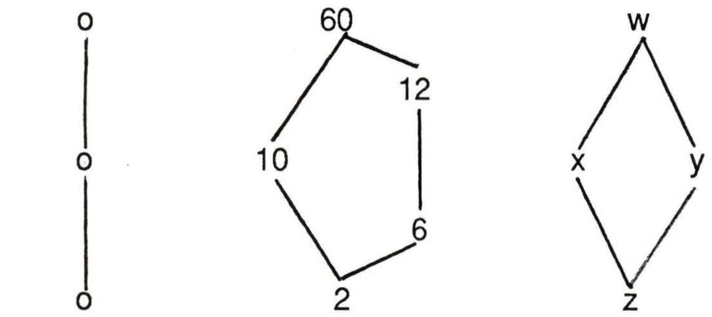
Figure 5



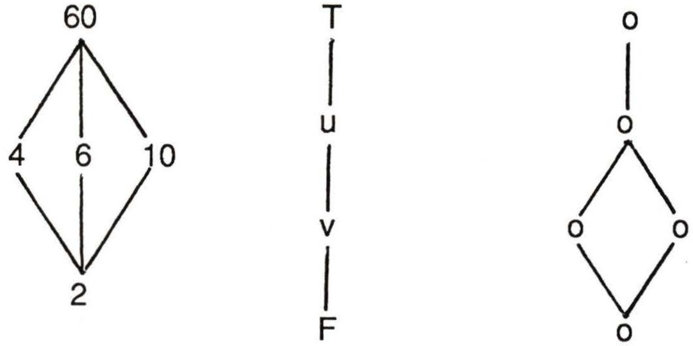
(a): [= means "Subset of"

(b): [= means "divisor of"

Figure 6



6a 6b 6c



6d 6e 6f

Appendix 2

Proofs

PROOFS

Overview:

Section numbers and theorem and formula labels correspond with those in section 4.

In the expression pairs a14 - a18, each expression is the dual of the other member of the pair, ie, one expression can be obtained from the other by exchanging V and A throughout. In proving theorems about dual expressions, it is only necessary to prove a theorem for one of the expressions; proof for the dual follows by duality [220]. Some of the other proofs could be shortened by appealing to duality.

0 Notation (abbreviations in the metalanguage)

'A ==> B' means 'if A then B'
'&&' means 'and'
'A && B ==> C' means '(A && B) ==> C'
'A iff B' means '(A ==> B) && (B ==> A)'

1 The formal language L

Let L = < S, -, &, |, =>, (,) >, where

S is a set of constants
- is a unary connective
&,|,=> are binary connective
(,) are punctuation symbols

The wffs of L are all expressions built up by finitely many applications of rules 1.1 and 1.2:

1.1 Any constant in S is a wff of L.

1.2 Let p,q be wffs of L. Then p, (-p), (p & q), (p | q), (p => q), are wffs of L.

Let W be the set of all wffs of L.

2.1 Equivalence relation E

Define the binary relation Q on W:

for every x, y, z in W,

a1 xQx reflexive
a2 xQy && yQz ==> xQz transitive

Then Q is a quasi-order on W

Theorem t1: Let E be a binary relation on W such that

for every x, y in W,

a3 xEy iff xQy && yQx

Then E is an equivalence relation: for every x,y,z in W,

xEx reflexive
xEy ==> yEx symmetric
xEy && yEz ==> xEz transitive

Proof: for every x,y,z in W,

1	xQx && xQx	{ a1, assumption }	
2	xEx	{ 1, a3 }	(reflexive)
3	xEy	{ assumption }	
4	xQy && yQx	{ 3, a3 }	
5	yQx && xQy	{ 4 }	
6	yEx	{ 5, a3 }	
7	xEy ==> yEx	{ 3, 6 }	(symmetric)
8	xEy && yEz	{ assumption }	
9	xQy && yQz	{ 8, a3 }	
10	xQz	{ 9, a2 }	
11	yQx && zQy	{ 8, a3 }	
12	zQy && yQx	{ 11 }	
13	zQx	{ 12, a2 }	
14	xEz	{ 10, 13, a3 }	
15	xEy && yEz ==> xEz	{ 8, 14 }	(transitive)

2.2 Constraints on Q and E in terms of connectives of L:

a4 (x&y)Qx, (x&y)Qy, zQx && zQy ==> zQ(x&y) glb of x,y
a5 xQ(x|y), yQ(x|y), xQz && yQz ==> (x|y)Qz lub of x,y

a6	x&y E x iff xQy,	x|y E y iff xQy	consistency
a7	x&x E x,	x|x E x	idempotent
a8	x&y E y&x,	x|y E y|x	commutative
a9	x&(y&z) E (x&y)&z,	x|(y|z) E (x|y)|z	associative
a10	x&(x|y) E x,	x|(x&y) E x	absorptive

2.3 Partition W into a set WW of equivalence classes

Let E partition W into classes X, Y, Z, ... of equivalent elements: For every x in X, y in Y,

a11 xEy iff X = Y

Let WW = { X, Y, Z, ... }.

The alternate notation [x] = X will also be used.

2.4 Define a partial order [= on WW.

Theorem T2: define the binary relation [= on WW:

a12 for every x in X and y in Y, X [= Y in WW iff xQy in W

Then X [= Y is a partial order on WW:

X [= X (reflexive)
X [= Y && Y [= X ==> X = Y (antisymmetric)
X [= Y && Y [= Z ==> X [= Z (transitive)

Proof:

1 X [= X { a1, a12 } (reflexive)
2 X [= Y && Y [= X { assumption }
3 xQy && yQx { 2, a12 }
4 xEy { 3, a3 }
5 X = Y { 4, a11 }
6 X [= Y && Y [= X ==> X = Y { 2, 5 } (antisymmetric)
7 X [= Y && Y [= Z { assumption }
8 xQy && yQz { 7, a12 }
9 xQz { 8, a2 }
10 X [= Z { 9, a12 }
11 X [= Y && Y [= Z ==> X [= Z { 7, 10 } (transitive)

2.5 The lattice M generated by WW under [=.

The set WW of E equivalence classes partially ordered by [= generates a lattice M with meet A and join V:

a13 For every x,y in W, [x]A[y] = [x&y], [x]V[y] = [x|y]

a14 For every X,Y in WW, there exist P,Q in WW such that

i P = XAY = [x&y], P [= X; glb of X,Y
ii Q = XVY = [x|y], Y [= Q; lub of X,Y

Proof: let X, Y be any classes in WW. Then

1 x, y are in W {a11}
2 x&y is in W {1, 1.2}
3 (x&y)Qx {2, a4}
4 [x&y] [= [x] {3, a12}
5 X A Y = [x&y] {a11, a13}
6 X A Y [= X {4,5} i done
7 x|y is in W {1, 1.2}
8 yQ(x|y) {7, a5}
9 [y] [= [x|y] {3, a12}
10 X V Y = [x|y] {a11, a13}
11 Y [= X V Y {9,10} ii done

By a12-a14, M is a lattice. Then the complete algebraic characterization of M follows from a6 - a11 and a13: For every X, Y, Z in WW,

a15 X A X = X, X V X = X idempotent

Proof:

1 [x] A [x] = [x&x] {a13}
2 (x&x)Ex {a7}
3 [x&x] = X {2, a11}
4 X A X = X {1, 3}
5 X V X = X {1-4 by duality} done

a16 X A Y = Y A X, X V Y = Y V X commutative

Proof:

1 [x] A [y] = [x&y] {a13}
2 (x&y)E(y&x) {a8}
3 [x&y] = [y&x] {2, a11}
4 [x] A [y] = [y&x] {1, 3}
5 X A Y = Y A X {4, a13}
6 X V Y = Y V X {1-5 by duality} done

a17 X A (Y A Z) = (X A Y) A Z associative
X V (Y V Z) = (X V Y) V Z

Proof:

1 [x] A ([y] A [z]) = [x&(y&z)] {1.2, a13}
2 (x&(y&z))E((x&y)&z) {a9}
3 [x&(y&z)] = [(x&y)&z] {2, a11}
4 [x] A ([y] A [z]) = [(x&y)&z] {1, 3}
5 X A (Y A Z) = (X A Y) A Z {4, a13}
6 X V (Y V Z) = (X V Y) V Z {1-5 by duality} done

a18 X A (X V Y) = X V (X A Y) = X absorptive

Proof:

1 [x] A ([x] V [y]) = [x&(x|y)] {1.2, a13}
2 = [x] {1, a10, a11}
3 = [x|(x&y)] {2, a10, a11}
4 = [x] V ([x] A [y]) {3, a13}
5 X A (X V Y) = X V (X A Y) = X {4, a13} done

a19 X [= Y iff XAY = X consistency
X [= Y iff XVY = Y

Proof:

1 X [= Y {assumption}
2 xQy {1, a12}
3 (x&y)Ex {2, a6}
4 [x&y] = [x] {3, a11}
5 X A Y = X {4, a13}
6 X A Y = X {assumption}
7 [x&y] = [x] {6, a13}
8 (x&y)Ex {7, a11}
9 xQy {8, a6}
10 X [= Y {9, a12}
11 X [= Y iff X A Y = X {1, 5, 6, 10}
12 X [= Y iff X V Y = Y {1-11 by duality} done

Since there are only a finite number of wffs of L, W, and hence WW, is finite. Consequently WW has a least element and a greatest element: for every X in WW there exist F, T such that

a20 F [= X [= T

a21 i) F = FAX, ii) X = FVX; F is least element
iii) T = TVX, iv) X = TAX T is greatest element

Proof:

1 F [= X {a20}
2 F A X = F {1, a19} i done
3 F V X = X {1, a19} ii done
4 X [= T {a20}
5 X V T = T {4, a19} iii done
3 X A T = X {4, a19} iv done

2.6 Define measure and distance functions on lattice M.

Define the measure function m on M, m:WW --> [0,1]
such that for T,F and every X,Y in WW

a22 m(T) = 1, m(F) = 0
a23 m(X) <= m(Y) iff X [= Y
a24 m(X A Y) + m(X V Y) = m(X) + m(Y) additivity

Then M is a metric lattice, hence modular.

Define the quasimetric distance d between X and Y, for every X,Y in WW:

a25 d(X,Y) = m(X V Y) - m(X A Y)

3 Assign truth-values for L in terms of the measure m.

Define the valuation function v on W: v:W --> [0,1] such that for every X,Y in WW, s,t in W, x in X, y in Y,

a26 v(s) = m(X) iff s is in X
a27 v(s & t) = m(X A Y) iff s is in X and t is in Y
a28 v(s | t) = m(X V Y) iff s is in X or t is in Y
a29 v(x => y) = 1 - d(X, X A Y)
a30 v(-x) = 1 - d(X, F)

4 Derive truth-value assignments for UL

Truth-value assignments for UL: For every x,y in W,

UL1 0 <= v(x) <= 1
UL2 v(x & y) <= min[v(p), v(q)]
UL3 v(x | y) >= max[v(p), v(q)]
UL4 v(x => y) = 1 - v(x) + v(x & y)
UL5 v(-x) = 1 - v(x)

Theorem T3: For every x,y in W, x in X, y in Y,

i v(x & y) <= min[v(p), v(q)] (UL2)
ii v(x | y) >= max[v(p), v(q)] (UL3)

Proof

1 (X A Y) A Y = X A (Y A Y) { a17 }
2 (X A Y) A Y = X A Y { 1, a15 }
3 X A Y [= Y { 2, a19 }
4 m(X A Y) <= m(Y) { 3, a23 }
5 v(x & y) <= v(y) { 4, a26, a27 }
6 (Y A X) A X = Y A X { 2, subst }
7 X A Y [= X { 6, a16, a19 }
8 m(X A Y) <= m(X) { 7, a23 }
9 v(x & y) <= v(x) { 8, a26, a27 }
10 v(x & y) <= min [v(x), v(y)] { 5, 9 } i done
11 (X V Y) V Y = X V (Y V Y) { a17 }
12 (X V Y) V Y = X V Y { 11, a15 }
13 Y [= X V Y { 12, a16, a19 }
14 m(Y) <= m(X V Y) { 13, a23 }
15 v(y) <= v(x | y) { 14, a26, a28 }
16 (Y V X) V X = Y V X { 12, subst }
17 X [= X V Y { 16, a16, a19 }
18 m(X) <= m(X V Y) { 17, a23 }
19 v(x) <= v(x | y) { 18, a26, a28 }
20 v(x | y) >= max [v(x), v(y)] { 15, 19 } ii done

Theorem T4: For every x,y in W, x in X, y in Y,

i v(x => y) = 1 - v(x) + v(x & y) (UL4)
ii v(x => y) = 1 + v(y) - v(x | y)

Proof: v(x => y) =

1 1 - m(X V (X A Y)) + m(X A (X A Y)) { a25, a29 }
2 1 - m(X) + m((X A X) A Y) { 1, a18, a17 }
3 1 - m(X) + m(X A Y) { 2, a15 }
4 1 - v(x) + v(x & y) { 3, a26, a27 } i done
5 1 + m(Y) - m(X V Y) { 3, a24 }
6 1 + v(x) - v(x | y) { 5, a26, a28 } ii done

Theorem T5: For every x in X,

v(-x) = 1 - v(x) (UL5)

Proof

1 v(-x) = 1 - m(X V F) + m(X A F) { a25, a30 }
2 = 1 - m(X) + m(F) { 1, a21 }
3 = 1 - m(X) { 2, a22 }
4 v(-x) = 1 - v(x) { 1, 3, a26 } done

5 Let M be a chain; derive truth-value assignments for FL.

Truth-value assignments for FL: For every x,y in W,

FL1 0 <= v(x) <= 1
FL2 v(x & y) = min[v(p), v(q)]
FL3 v(x | y) = max[v(p), v(q)]
FL4 v(x => y) = 1 - v(x) + v(x & y)
FL5 v(-x) = 1 - v(x)

FL1, FL4, and FL5 have been proved in section 4.

Theorem T6: If M is a chain, then for every x, y in W,

i v(x & y) = min[v(x), v(y)] (FL2)
ii v(x | y) = max[v(x), v(y)] (FL3)

Proof:

a31 Suppose M is a chain, ie, for every x in X, y in Y, X,Y in WW, X [= Y or Y [= X [221]. Then

1 X [= Y { assumption }
2 X A Y = X { 1, a19 }
3 m(X) <= m(Y) { 1, a23 }
4 m(X) = min[m(X), m(Y)] { 3 }
5 m(X A Y) = min[m(X), m(Y)] { 2, 4 }
6 v(x & y) = min[v(x), v(y)] { 5, a26, a27 }
7 Y [= X { assumption }
8 X A Y = Y { 7, a16, a19 }
9 m(Y) <= m(X) { 7, a23 }
10 m(Y) = min[m(X), m(Y)] { 9 }
11 m(X A Y) = min[m(X), m(Y)] { 8, 10 }
12 v(x & y) = min[v(x), v(y)] { 11, a26, a27 }
13 i done { 1, 6, 7, 12, a31 }

14 X [= Y { assumption }
15 X V Y = Y { 14, a19 }
16 m(X) <= m(Y) { 14, a23 }
17 m(Y) = max[m(X), m(Y)] { 16 }
18 m(X V Y) = max[m(X), m(Y)] { 15, 17 }
19 v(x | y) = max[v(x), v(y)] { 18, a26, a28 }
20 Y [= X { assumption }
21 Y V X = X { 20, a19 }
22 m(Y) <= m(X) { 20, a23 }
23 m(X) = max[m(X), m(Y)] { 22 }
24 m(X V Y) = max[m(X), m(Y)] { 21, 23, a16 }
25 v(x | y) = max[v(x), v(y)] { 24, a26, a28 }
26 ii done { 14, 19, 20, 25, a31 }

6 Let M be Boolean; derive truth-value assignments for PL.

Truth-value assignments for PL: For every x,y in W,

PL1 0 <= v(x) <= 1
PL2 v(x | -x) = 1
PL3 v(x | y) = v(x) + v(y), if v(x & y) = 0
PL4 v(x) = v(y), if v(x & -y) = 0 and v(-x & y) = 0
PL5 v(x => y) = v(-x | y)

Theorem T7: If M is Boolean, then for every x in W,

i v(x & -x) = 0
ii v(x | -x) = 1 (PL2)

Suppose that M is Boolean, ie, that M is complemented:

a32 for every X in WW there is a Y in WW such that

X A Y = F, X V Y = T

and that M is distributive [222]:

a33 for every X, Y Z in WW,

X A (Y V Z) = (X A Y) V (X A Z)

L1: for all x in X, y in Y, if X A Y = F, then (-x)Ey

1 X A Y = F {assumption}
2 m(X A Y) = 0 { 1, a22, a32 }
3 m(X V Y) = 1 { 1, a22, a32 }
4 m(X) + m(Y) = 1 { 2, 3, a24 }
5 v(y) = 1 - v(x) { 4, a26 }
6 v(y) = v(-x) { 5, T5 }
7 v(-x) = m(Y) { 6, a26 }
8 -x is in Y { 7, a26 }
9 (-x)Ey { 8, a11 }
10 done { 1, 9 }

Define the equivalence class -X of complements to x:

a34 -X = Y iff X A Y = F { L1 }

Since M is distributive, the complement -X of X is unique [223].

Proof of T7:

1 m(X A -X) = 0 { a22, a34 }
2 v(x & -x) = 0 { 1, a27 } i done
3 v(x & -x) + v(x | -x) = v(x) + v(-x) { a24, a27, a28 }
4 v(x | -x) = v(x) + v(-x) { 2, 3 }
5 v(x | -x) = 1 { 4, T5 } ii done

Theorem T8: for every x,y in W,

if v(x & y) = 0 then v(x | y) = v(x) + v(y) (PL3)

Proof:

a35 Suppose that v(x & y) = 0. Then

1 v(x & y) + v(x | y) = v(x) + v(y) {a24, a26, a27, a28}
2 v(x | y) = v(x) + v(y) { 1, a35 } done

Theorem T9: If M is Boolean, then for every x,y in W,

if v(x & -y) = 0 and v(-x & y) = 0 then v(x) = v(y) (PL4)

Proof:

a36 Suppose that v(x & -y) = 0 and v(-x & y) = 0. Then

1 v(x | -y) <= 1 { definition of v }
2 v(x) + v(-y) <= 1 { 1, a36, T8 }
3 v(x) <= 1 - (1 - v(y)) { 2, T5 }
4 v(x) <= v(y) { 3 }
5 v(-x | y) <= 1 { dfn of v }
6 v(-x) + v(y) <= 1 { 5, a36, T8 }
7 v(y) <= 1 - (1 - v(x)) { 6, T5 }
8 v(y) <= v(x) { 7 }
9 v(x) = v(y) { 4, 8 } done

Theorem T10: If M is Boolean, then

v(x => y) = v(-x | y)

Lemma L2: v(x & y) + v(x | y) = v(x) + v(y) {a24, a27, a28}

Lemma L3: (X A Y) A (-X A Y) = F

Proof:

1 (X A Y) A (-X A Y) = ((X A Y) A -X) A Y {a17}
2 = (-X A (X A Y)) A Y {1, a16}
3 = ((-X A X) A Y) A Y {2, a17}
4 = (F A Y) A Y {3, a34}
5 = F A Y {4, a21}
6 = F {5, a21}

Lemma L4: m(Y) = m(X A Y) + m(-X A Y)

Proof:

1 m((X A Y) A (-X A Y)) = 0 {L3, a22}
2 m(Y) = m((X A Y) A (-X A Y)) + m(Y) {1}
3 = m((X A Y) A (-X A Y)) + m(Y A (X V -X)) {2,a34,a21}
4 = m((X A Y) A (-X A Y)) + m((Y A X) V (Y A -X)) {3,a33}
5 = m(X A Y) + m(-X A Y) {4,a24}
6 m(Y) = m(X A Y) + m(-X A Y) {2, 5} done

T10: v(-x|y) = v(x => y)

Proof:

1 v(-x&y) + v(-x|y) = v(-x) + v(y) {L2}
2 v(-x|y) = v(-x) + v(y) - v(-x&y) {1}
3 = v(-x) + v(x&y) + v(-x&y) - v(-x&y) {2,L4,a26,a27}
4 = 1 - v(x) + v(x&y) {3,T5}
5 v(-x|y) = v(x => y) {4,T4} done

7 Let M be a boolean chain; derive truth-value assignments for SC

Truth-value assignments for SC: For every x,y in W, and
v --> {0,1},

SC1 v(-x) = 1 iff v(x) = 0
v(-x) = 0 iff v(x) = 1

SC2 v(x & y) = 1 iff v(x) = 1 and v(y) = 1
v(x & y) = 0 iff v(x) = 0 or v(y) = 0

SC3 v(x | y) = 1 iff v(x) = 1 or v(y) = 1
v(x | y) = 0 iff v(x) = 0 and v(y) = 0

SC4 v(x => y) = 1 iff v(-x) = 1 or v(y) = 1
v(x => y) = 0 iff v(-x) = 0 and v(y) = 0

Theorem T11: If M is a Boolean chain, then
for every x in W,

v(x) = 0 or v(x) = 1

Proof:
a37 Suppose M is a Boolean chain. Then

1 v(x & y) = min[v(x), v(y)] { a37, T6 }
2 v(x | y) = max[v(x), v(y)] { a37, T6 }
3 v(x & -x) = 0 { a37, T7 }
4 v(x | -x) = 1 { a37, T7 }
5 min[v(x), v(-x)] = 0 { 1, 3 }
6 max[v(x), v(-x)] = 1 { 2, 4 }
7 v(x) = 0 or v(x) = 1 { 5, 6 } done

Theorem T12: If M is a Boolean chain, then for every x in W,

v(-x) = 1 iff v(x) = 0 (SC1)
v(-x) = 0 iff v(x) = 1

Proof: immediate by theorems T5, T11.

Theorem T13: If M is a Boolean chain, then for every x,y in W,

v(x & y) = 1 iff v(x) = 1 and v(y) = 1 (SC2)
v(x & y) = 0 iff v(x) = 0 or v(y) = 0

Proof: immediate by theorems T6, T11.

Theorem T14: If M is a Boolean chain, then for every x,y in W,

v(x | y) = 1 iff v(x) = 1 or v(y) = 1 (SC3)
v(x | y) = 0 iff v(x) = 0 and v(y) = 0

Proof: immediate by theorems T6, T11.

Theorem T15: If M is a Boolean chain, then for every x,y in W,

i v(x => y) = 1 iff v(-x) = 1 or v(y) = 1 (SC4)
ii v(x => y) = 0 iff v(-x) = 0 and v(y) = 0

Proof: Suppose M is a Boolean chain.

Part i

a38 Suppose v(x => y) = 1. Suppose also

n38 Not: v(-x) =1 or v(y) = 1. Then

(proof by contradiction):

1 v(-x) = 0 { n38, T11 }

2 v(y) = 0 { n38, T11 }

3 v(x) = 1 { 1, T5 }

4 v(x | y) = 0 { 2, a38, T4 }

5 v(x & y) = 1 { 3, a38, T4 }

6 max[v(x), v(y)] < min [v(x), v(y)] { 4, 5 }

7 v(-x) = 1 or v(y) = 1 { 6, n38 }

8 done i "if" { 7, a38 }

a39 Suppose v(-x) = 1. Then

12 v(x) = 0 { a39, T5 }

13 v(x & y) = 0 { 12, T13 }

14 v(x => y) = 1 { 12, 13, T4 }

a40 Suppose v(y) = 1. Then

15 v(x | y) = 1 { a40, T14 }

16 v(x => y) = 1 { a40, 15, T4 }

17 i done { a38, 11, a39, 14, a40, 16 }

Part ii:

a41 Suppose v(x => y) = 0. Then

1 v(x) = 1 + v(x & y) { a41, T4 }

2 0 <= v(x) <= 1 { definition of v }

3 1 + v(x & y) <= 1 { 1, 2 }

4 v(x & y) = 0 { 2, 3 }

5 v(x) = 1 { 1, 4 }

6 v(-x) = 0 { 5, T5 }

7 v(y) = 0 { 4, 5, T13 }

8 "only if" done { a41, 6, 7 }

a42 Suppose v(x) = 1 and v(y) = 0 Then

7 v(x & y) = 0 { a42, T13 }

8 v(x => y) = 0 { a42, 7, T4 }

9 "if" done { 8, T14 }

10 ii done { 6, 9 }

VITA

Surname: Akin   Given Names: Garth Stuart

Place of Birth: Riverside, CA, US. Date of Birth: 6-16-39

Educational Institutions Attended:

University of Victoria	1987 to 1990
University of Victoria	1979 to 1984
University of California, Berkeley	1965 to 1968
San Diego State College	1961 to 1964
Oakland City College	1960 to 1961
Riverside College	1957 to 1959

Degrees Awarded

B.A. San Diego State College 1964

Publications

Akin, G. and Reuter, L. Forest Regeneration Assistant User Guide, AAL Autometrics Associates Ltd. Victoria, B.C., 1988.

Dukarm, J. and Akin, G. VRP Expert System Developer's Guide, AAL Autometrics Associates Ltd. Victoria, B.C., 1988.

Akin, G. and Eng, D. Computer Science 100 Lab Manual, University of Victoria, Victoria, B.C., 1983.

Amidon, E. and Akin, G. "Algorithmic Selection of the Best Method for Compressing Map Data Strings." In Communications of the ACM, Dec. 1971.

Sharpnack, D. and Akin, G. "An Algorithm for Computing Slope and Aspect from Elevations." In Photogrammetric Engineering, March, 1969.

Amidon, E. and Akin, G. "Dynamic Programming to Determine Optimum Levels of Growing Stock." In Forest Science, V.14 No.3, 1968.