FUZZY LOGIC AND THE RESOLUTION PRINCIPLE

Richard C. T. Lee

National Institutes of Health
Bethesda, Maryland 20014

Abstract

Problem-solving systems using two-valued logic suffer from one drawback, they cannot handle fuzzy, or uncertain, information. In this paper, the author recommends the use of fuzzy logic, which is based on the concept of fuzzy sets and first order predicate calculus. It is proved that, in fuzzy logic, a set of clauses is unsatisfiable iff it is unsatisfiable in two-valued logic. It is also shown that if the most unreliable clause of a set of clauses has a truth-value a>0.5, then all the logical consequences obtained by repeatedly applying the resolution principle has truth-value never smaller than a. Implications of these results for applying fuzzy logic to problem-solving *are* discussed.

Descriptive terms

fuzzy logic, fuzzy sets, first order predicate calculus, problem-solving systems, question-answering systems, resolution principle, uncertainty, completeness theorems.

1. Introduction

In recent years, we have witnessed great progress in techniques of designing problem-solving and question-answering systems (9, 23, 25, 19). The new approach may be called an axiomatic approach, where facts and information are stored as axioms and the original problem to be solved, or the original question to be answered, is stated as a theorem to be proved.

This axiomatic approach suffers from one drawback. That is, since we store information as axioms, we are assuming that the information is absolutely correct and no uncertainty is involved. But, everyone knows that the real world is full of uncertainty and any machine that is going to work in this real world, such as a robot designed for exploring the surface of the moon, should be able to make intelligent decisions in a "fuzzy" environment.

Obviously, it would be nice if one could combine probability theory with symbolic logic. But we do not seem to know how to do this. (This was pointed out in (17)). In this paper, we propose that instead of two-valued logic, one may use fuzzy logic (26, 27) which is a special kind of many-valued logic. The proposal to use many-valued logic in problem-solving systems is nothing new, Green proposed this in 1969 (9). It should be emphasized that the author by no means claims that the problem of uncertainty is well taken care of if fuzzy logic is used. In fact, the main contribution of this paper is to show many interesting properties of fuzzy logic which make it suitable for incorporation into problem-solving systems.

Despite the author's effort to make this paper self-contained, it may still be too difficult for many naive readers. The reader at least has to have some background in elementary symbolic logic and is encouraged to read (20), (23), and (19) for an introduction to mechanical theorem-proving and its applications to problem-solving systems.

2. Fuzzy Logic

Fuzzy logic is based on the concepts of fuzzy sets (3, 5, 6, 15, 26, 27) and symbolic logic. In (15) and (13), the discussion of fuzzy logic was limited to propositional calculus.

We may view fuzzy logic as a special kind of many-valued logic (21, 1, 2). In fuzzy logic, the truth-value of a formula, instead of assuming two values, (0 and 1), can assume any value in the interval [0,1] and is used to indicate the degree of truth represented by the formula.

For example, let P(x) represent "x is a large number." Then the truth-value of $P(10^6)$ and P(10") are certainly 1 and 0 respectively. As for P(125), the truth-value of it may be some value between 0 and 1, say 0.25.

We shall assume that well formed formulas (formulas for short) are defined to be exactly the same as those in two-valued logic. Letting T(S) denote the truth-value of a formula S, the evaluation procedure for a formula in fuzzy logic can be described as follows:

(1) $T(S)=T(A)$ if $S=A$ and A is a ground atomic formula.

(2) $T(S)=1-T(R)$ if $S=\sim R$.

(3) $T(S)=\min[T(S_1),T(S_2)]$ if $S=S_1 \& S_2$.

(4) $T(S)=\max[T(S_1),T(S_2)]$ if $S=S_1 v S_2$

(5) $T(S)=\inf[T(B(x))|x \in D]$ if $S=(x)B$ and D is the domain of x.

(6) $T(S)=\sup[T(B(x))|x \in D]$ if $S=(Ex)B$ and D is the domain of x.

Note that if D is a finite set, then (5) and (6) become

(5)' $T(S)=T(B(a_1)\& \ldots \& B(a_n))$

if $S=(x)B$ and x assumes values $a_1,\ldots,a_n$.

and (6)' $T(S)=T(B(a_1)\vee \dots \vee B(a_n))$
if $S=(Ex)B$ and x assumes values $a_1,\dots,a_n$.

The reader should note that two-valued logic is a special case of fuzzy logic; all the rules stated above *are* applicable in two-valued logic.

Example 1

Consider S=(PvQ)&-R.

Assume $T(P)=0.1$, $T(Q)=0.7$ and $T(R)=0.6$.

Then $T(S)=\min[\max[T(P),T(Q)], 1-T(R)]$
$=\min[\max[0.1, 0.7], 1-0.6]$
$=\min[0.7, 0.4]$
$=0.4$.

Example 2

Consider $S=(x)P(x)$.

Assume the domain of x is $\{a_1,a_2,a_3\}$ and $T(P(a_1))=0.1$, $T(P(a_2))=0.7$ and $T(P(a_3))=0.5$.

Then $T(S)=T(P(a_1)\&P(a_2)\&P(a_3))$
$=\min[P(a_1),P(a_2),P(a_3)]$
$=\min[0.1, 0.7, 0.5]$
$=0.1$.

Example 3

Consider $S=(x)(Ey)P(x,y)$.

Assume $D=\{a_1,a_2\}$ and
$T(P(a_1,a_1))=0.1$, $T(P(a_1,a_2))=0.3$
$T(P(a_2,a_1))=0.2$ and $T(P(a_2,a_2))=0.7$.

Note that $T((Ey)P(x,y))=T(P(x,a_1)\vee P(x,a_2))$.

Therefore, $T((x)(Ey)P(x,y))$
$= T((P(a_1,a_1)\vee P(a_1,a_2))\&(P(a_2,a_1)\vee P(a_2,a_2)))$
$= \min[\max[P(a_1,a_1),P(a_1,a_2)],$
$\max[P(a_2,a_1),P(a_2,a_2)]]$
$= \min[\max[0.1,0.3], \max[0.2,0.7]]$
$= \min[0.3,0.7]$
$= 0.3$.

If two-valued logic is used in a problen.-solving system, one stores a statement if and only if the truth-value of it is 1. (If the truth-value of a statement A is 0, one simply stores -A). In fuzzy logic, obviously we should store a statement A, instead of -A, iff the truth-value of A is greater than or equal to that of -A. That is, we store A iff

$T(A)\geq 1-T(A)$.

In this case, $T(A)\geq 0.5$.

We can use this concept to define "satisfiability" in fuzzy logic. An interpretation I is said to satisy a formula S if $T(S)\geq 0.5$ under I. An interpretation I is said to falsify S if $T(S)\leq 0.5$ under I. (According to this definition, Tf $T(S)=0.5$ under I, then I both satisfies and falsifies S.) A formula is said to be unsatisfiable if it is falsified by every interpretation of it. Again, it should be easy to note that these definitions are compatible with those in two-valued logic. However, in fuzzy logic, "not satisfying" is different from "falsifying" and "not falsifying" is different from "satisfying."

In applying logic to problem-solving systems, we often have to prove the unsatisfiability of a formula (19). We shall devote the next section to this subject.

3. Satisfiability in Fuzzy Logic

In (13), the following theorem was proved.

Theorem I

A ground formula S is unsatisfiable in fuzzy logic if and only if it is unsatifiable in two-valued logic.

Regretfully, the above theorem has only been proved for the ground case. In this section, we shall prove a theorem similar to the above one except that the formula does not have to be a ground formula.

We shall assume that formulas contain no existential quantifiers. Variables that are existentially quantified are all replaced by Skolem functions (7). Due to the existence of distributive and De Morgan's laws, we can also assume that the formula is in the form of a conjunction of clauses where *every* variable is universally quantified.

Since an interpretation can be defined over any domain, it seems that we would have to consider an infinite number of domains. Fortunately, so far as unsatisfiability is concerned, we only have to consider one particular domain, namely, the Herbrand universe (10, 17).

It is assumed here that the reader is familiar with the definition of Herbrand universe.

Given an interpretation I of a set S of clauses over an arbitrary domain D, we can always construct an interpretation I' of S over the Herbrand universe of S that preserves some interesting properties of I. We shall call this interpretation I' the H-interpretation of S with respect to I (over domain D).

Before giving the definition of the H-interpretation, note that the interpretation I maps every constant occurring in S to some element in D. In case no constant occurs in S, then the basic element a which initiates the construction of the Herbrand universe is assumed to be mapped to any element in D. Thus, *every* element in the Herbrand universe of S is assumed to be mapped by I to some element in the domain D.

Let I be an interpretation of a set S of clauses over a domain D. Let H denote the Herbrand universe of S. The H-interpretation I' of S (over the Herbrand universe of S) with respect to I is defined as follows:

(1) I' maps all constants occurring in S to themselves.

(2) Let $h_1, h_2 ... h_n$ be elements of H. Let f be an n-place function symbol (n>0) occurring in S. In I^1, f is assigned to be a function which maps $\{h_1,h_2,....,h_n\}$ (an element in H^n) to $f(h_1,.h_2.....h_n)$ (an element in H).

(3) Let h,,...,h be elements of H. Let P be an n-place (n>0) predicate symbol occurring in S. Let *every* element h. be mapped to some d_i in D. If $P(d_1,...,d_n)$ is assigned a truth-value t by I, then P(h, ,...,h) is also assigned the truth-value t.

Example 4

Consider the following clause:

P(a,f(x)).

The interpretation I is defined as follows:

D = {1,2}.

Assignment of constants: a → 1

Assignment of functions: f(1) → 2

f(2) → 1

Assignment of predicates:

T(P(1,1))=0.57 T(P(I,2))-0.7

T(P(2,1))=0.47 T(P(2,2))=0.36

The Herbrand universe of S is H={a,f(a), f^2(a),...}

The H-interpretation I' of S is constructed as follows:

Assignment of constants: a -> a

Assignment of functions: f(a) -> f(a)

f^2(a) -> f^2(a)

Assignment of predicates:

T(P(a,f(a)))=T(P(I,f(I)))=T(P(I,2))=0.7

T(P(a,f^2(a)))=T(P(I,f^2(I)))=T(P(I,I))=0.57

T(P(a,f^3(a)))=T(P(I,f^3(I)))=T(P(I,2))=0.7

In this example, it is easy to see that I satisfies S. It is interesting to note that the H-interpretation I' (over the Herbrand universe of S) with respect to I also satisfies S. In fact, we can prove the following lemma.

Lemma 1

In fuzzy logic, if an interpretation I of a set S of clauses over a domain D does not falsify S, then the H-interpretation I' oT S (over the Herbrand universe of S) does not falsify S either.

Proof: Assume I' does falsify S. Then there must exist at least one clause C in S such that T(C)< 0.5 under I'. Let x, ,...,x be the variables occurring in C. Then there exist $h_1,...h_n$ in the Herbrand universe of S such that T(C')<0.5 where C is the ground clause obtained from C by replacing every x. with h.. Let every h. be mapped to some d_i in D by the interpretation I. According to the definition of the H-interpretation of S, if C" is the ground clause obtained from C by replacing *every* x. with d. then T(C")<0.5 also. This means that I falsifies S which is impossible. Q.E.D.

Using the above lemma, we can prove the following important theorem.

Theorem 2

A set of clauses is unsatisfiable in fuzzy logic if and only if it is falsified by every interpretation I over the Herbrand universe of S.

Proof;

(a) (→)

This part of the proof is trivial. By definition, S is unsatisfiable if and only if S is falsified by all interpretations over any domain, including the Herbrand universe of S.

(b) (←)

Assume S is not falsified by an interpretation I over some domain D. According to Lemma 1, there exists an interpretation I' over the Herbrand universe of S such that I' does not falsify S either. This contradicts the assumption. Q.E.D.

It can be easily seen that so far as satisfiability is concerned, the only relevant information is whether a ground atom is assigned a truth-value greater than, equal to or smaller than 0.5. For example, a ground clause is satisfied by an interpretation if the truth-value of at least one literal of the clause is greater than or equal to 0.5. Therefore, in the sequel, instead of giving the exact truth-value of a ground atom in an interpretation, we shall merely compare it with 0.5.

Furthermore, since the domain is fixed to be the Herbrand universe, we only have to consider the truth-value of those ground atoms whose arguments are elements of the Herbrand universe. Let A be the set of ground atomic formulas of the form $P(h_1,\ldots,h_n)$ for all n-place predicate symbols occurring in S, where every h. is an element of the Herbrand universe of S. The set A is called the atom set of S.

We still have not discussed how to determine whether a set S of clauses is unsatisfiable or not. Obviously, we can exhaustively construct all possible interpretations. If a partial interpretation is found to falsify a clause, we may stop constructing that particular interpretation. Let us enumerate atomic formulas in the atom set of S by

A A A

Note that each different assignment of truth-value (with respect to 0.5) to each A. corresponds to a distinct branch of the infinite binary tree shown in Fig. 1. Since the path from the root of the tree T to *every* node N of the tree

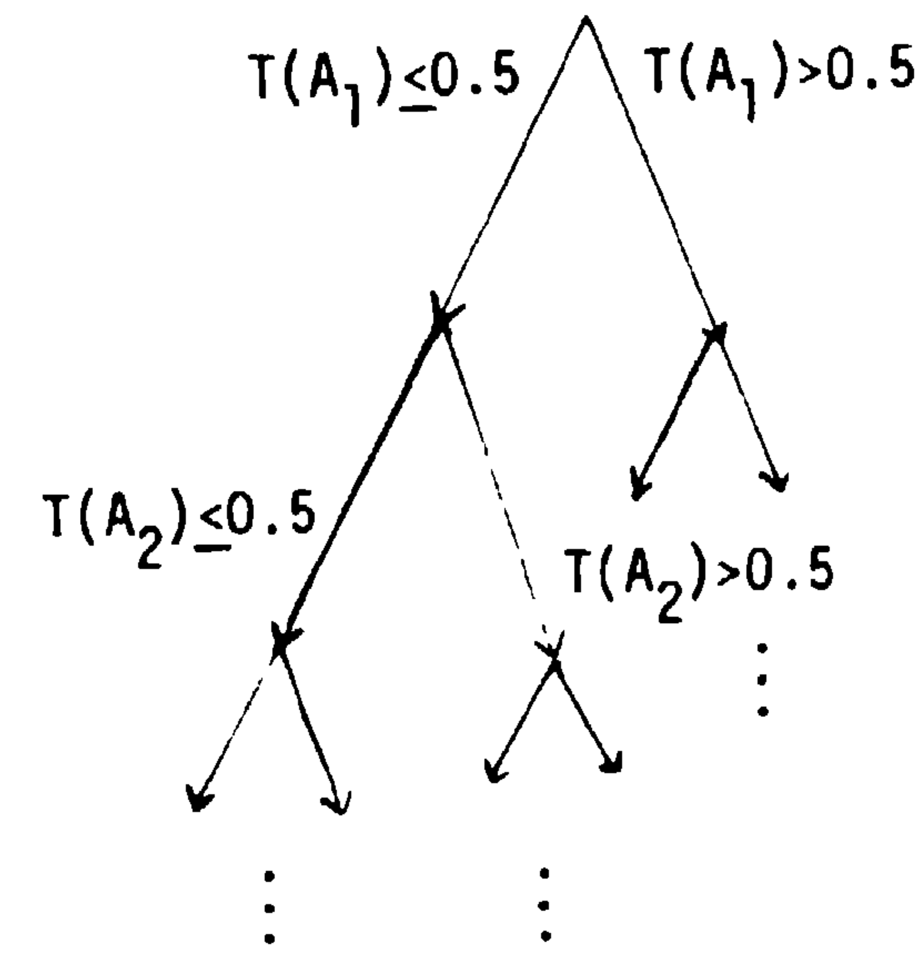

Figure 1

represents a partial interpretation I of S, we can terminate the tree at node N if I falsifies a clause. In such a case, N is called a failure node. If S is unsatisfiable, then *every* branch of the semantic tree in Fig. 1 is terminated by a failure node and vice versa. We shall call the semantic tree T in Fig. 1 a closed semantic tree if every branch of T is terminated by a failure node.

Although corresponding to *every* ordering of the elements in the atom set of S, there is a closed semantic tree for S if and only if S is unsatisfiable, we still cannot use this approach until we know that the closed semantic tree is also a finite one. The following theorem essentially shows us that it is indeed the case.

Theorem 3

A set S of clauses is unsatisfiable in fuzzy logic if and only if corresponding to every ordering of the elements~of the atom set of S, the semantic tree of S is a finite closed semantic tree.

Proof:

(a) (→)

Suppose S is unsatisfiable. Then no interpretation over the Herbrand universe of S satisfies S. Therefore *every* possible assignment of trutn-values to $A_1 A_2,\ldots,$ will falsify at least one clause of S. Hence *every* branch of the semantic tree is terminated by a failure node. Assume there are infinitely many nodes in the tree, then the initial node would be a node with infinitely many descendants. Moreover, if any node has infinitely many descendants, then at least one of its descendants has infinitely many descendants. Therefore there is a non-terminating branch in the semantic tree which is impossible. Thus the closed semantic tree must be finite.

(b) (←)

This part of the proof is trivial and will be omitted.

From Theorem 3, we can easily obtain Theorem 4 as follows:

Theorem 4

A set S of clauses in fuzzy logic is unsatisfiable if and only if there is a finite unsatisfiable (in fuzzy logic) set S' of ground instances of S over the Herbrand universe of S.

Since two-valued logic is a special case of fuzzy logic, the above theorem is true for two-valued logic. A proof of the foregoing theorem in two-valued logic can be found in (4).

Using Theorems 1 and 4, we can now prove the following theorem which is the main contribution of this section.

Tneorem 5

A set $ of clauses is unsatisfiable in fuzzy logic if and only if it is unsatisfiable in two-valued logic.

Proof:

(a) (→)

Since S is unsatisfiable in fuzzy logic, S must be falsified by every interpretation of S. In particular, S must be falsified by all the interpretations in which the truth-value is either 1 or 0. Thus S must be unsatisfiable in two-valued logic.

(b) (<-)

Since S is unsatisfiable in two-valued logic, there must exist a finite unsatisfiable (in two-valued logic) set S' of ground instances in S over the Herbrand universe of S. According to Theorem I, S' must be also unsatisfiable in fuzzy logic. From Theorem 4, we conclude that S is unsatisfiable in fuzzy logic.

4. The Concept of Logical Consequence in Fuzzy Logic

Given a formula F, we shall define a formula G to be a logical consequence of F if and only if F&-G is unsatisfiable. If F&-G is unsatisfiable, then T(F&-G)^0.5 under all interpretations. If we further require that T(F)>0.5, then T(-G) must not be larger than 0.5. But T(-G)=1-T(G). Therefore T(G)>0.5 in all interpretations in which T(F)>0.5. This means that if the degree of truth of F exceeds 0.5, the degree of truth of all logical consequences of F is never smaller than 0.5. Note that our definition of logical consequence in fuzzy logic is compatible with that in two-valued logic. In two-valued logic, T(G)=1 whenever T(F)=1.

Using the above definition and Theorem 5, we can establish the following lemma.

Lemma 2

A formula G is a logical consequence of a formula F in fuzzy logic if and only if G is a logical consequence of G in two-valued logic.

equenc

The proof of this lemma is omitted.

In two-valued logic, there is a *very* good inference rule, called the resolution principle (20), which has been proved complete for deducing logical consequences (12,24,22). In the following, we shall show that the resolution principle is also complete in fuzzy logic. It is assumed here that the reader is familiar with the resolution principle.

Let S be a set of clauses. The resolution of S, denoted R(S), is the set consisting of members of S together with all the resolvents of the pairs of members of S. The n-th resolution of S, denoted by $R^n(S)$, is defined for n^0 as follows:

$$R^0(S)=S \text{ and } R^{n+1}(S)=R(R^n(S)).$$

The completeness theorem in (12) can now be stated as follows:

Theorem 6

In two-valued logic, if a clause C is a logical consequence of a set S of clauses, then for some n≥0, there exists a clause $C' \epsilon R^n(S)$ such that C is a logical consequence of C'.

Given a set S of clauses, let us define a logical consequence C (C is assumed to be a clause) to be a prime logical consequence of S if there exists no other logical consequence C of S (C is also a clause) such that C is also a logical consequence of C'. Theorem 6 can now be stated as Theorem 7.

Theorem 7

In two-valued logic, if a clause C is a prime logical consequence of a set S of clauses, then for some n>0, $C \epsilon R^n(S)$

A similar theorem in the context of fuzzy logic will now be proved. This is accomplished by first proving the following lemma.

Lemma 3

Given a set S of clauses, a clause C is a prime logical consequence of S in fuzzy logic if and only if C is a prime logical consequence of S in two-valued logic.

Proof:

(a) (→)

Since C is a logical consequence of S in fuzzy logic, by Lemma 2, C is a logical consequence of S in two-valued logic. Assume C is not a "prime" logical consequence of S. Then there is a logical consequence C of S such that C is a logical consequence of C in two-valued logic. Applying Lemma 2 again, we will be able to conclude that the relationship governing C, C and S also holds in fuzzy logic. Thus C is not a "prime" logical consequence of S in fuzzy logic, which contradicts the assumption. Hence C must be a prime logical consequence of S in two-valued logic.

(b) (-)

This part of the proof is similar to the proof of part (a) above.

Using Theorem 7 and Lemma 3, we can deduce the completeness theorem of the resolution principle in fuzzy logic.

Theorem 8

In fuzzy logic, if a clause C is a prime logical consequence of a set S of clauses, then for some $n>0$, $C \subset R^n(S)$.

A fundamental property of probability theory is that "if G is a consequence of F, then P(G)>P(F)." Beautiful discussions on this property can be found in (8) and (14). We have a similar relationship in two-valued logic. That is, in two-valued logic, if G is a logical consequence of F, then T(G)>T(F). However, we cannot establish this in fuzzy logic. Consider the following example.

Example 5

Consider C_1: -PvQ

and C_2: P.

Q is a resolvent (thereby a logical consequence) of C, and C_2.

Let T(P)=0.3 and T(Q)=0.2.

We have $T(C_1\&C_2)=T((-PvQ)\&P)$

= nnn[max[T(-P),T(Q)],T(P)]

= min[max[0.7,0.2],0.3]

= min[0.7,0.3]

= 0.3.

Thus $T(Q)$^$T(C_1\&C_2)$.

The above example shows that the truth-value of $C_1\&C_2$ is not necessarily smaller than or equal to that of a logical consequence of C_1 and C_2. But, if $T(C_1\&C_2)>0.5$, we have a different situation.

Lemma 4

Let C_1 and C_2 be two clauses. Let $R(C_1,C_2)$ denote any resolvent of C_1 and C_2. If $T(C_1\&C_2)>0.5$, then $T(R(C_1,C_2))>T(C_1\&C_2)$.

Proof:

Without losing generality, we can represent C, and C_2 as follows:

C_1: PvL_1

and C_2: $-PvL_2$

where L_1 and L_2 are two disjunctions of literals $R(C_1,C_2)=L_1vL_2$.

Let $T(C_1\&C_2)=\min[T(C_1),T(C_2)]=a$.

Again, without losing generality, we can assume that

$T(C_1)=\max[T(P),T(L_1)]=a$ (i)

and $T(C_2)=\max[T(-P),T(L_2)>a$. (2)

From (1), we conclude that T(L-j) can be either equal to or smaller than a.

Case 1: Assume $T(L_1)=a$. Then $T(R(C_1,C_2))$

$= T(L_1vL_2)$

$= \max[T(L_1),T(L_2)]\geq a$.

But $T(C_1\&C_2)=a$. Therefore, $T(R(C_1,C_2))\geq T(C_1\&C_2)$.

Case 2: Assume $T(L_1)<a$.

From (1), T(P) must be equal to a. But a>0.5. Therefore

$T(-P)=1-T(P)<0.5<a$.

Consequently, from (2), $T(L_2)>$ $T(R(C_1C_2))$

$= T(L_1 v L_2)$

$\equiv \max[T(L_1),T(L_2)]>a.$

Therefore $T(R(C_1,C_2))>T(C_1\&C_2)$. Q.E.D

Using the definition of $R^n(S)$, we can extend the result of Lemma 4 to the following theorem.

Theorem 9

Let S be a set of clauses and $T(S)>0.5$. Let C^n denote any clause in the set $R^n(S)$. Then for all $n\to 0$, $T(C^n)>T(S)$.

Theorem 9 is an interesting theorem. It shows that if *every* clause in S is something more than a "half-truth" and the most unreliable clause has truth-value a, then we *are* guaranteed that all the logical consequences obtained by repeatedly applying the resolution principle will have truth-value at least equal to a. We shall discuss the significance of this theorem in the next section.

5. Conclusions

Many readers may find the "min-max" principle in fuzzy logic disturbing. That is, they may find it hard to accept

$$T(A\&B)=\min[T(A),T(B)]$$

and $T(AvB)=\max[T(A),T(B)]$.

To answer this criticism, we would like to point out that two-valued logic uses exactly the same evaluation procedure. By rejecting the evaluation procedure of fuzzy logic, one would simultaneously reject that of two-valued logic. We do not believe that fuzzy logic is adequate to describe the complex world we *are* living in (it would be simply too naive to think that the world can be described by a set of mathematical rules). However, it would not be difficult to see that it is an improvement of two-valued logic; at least it gives us a way to handle fuzzy information.

The estimation of the truth-value of a statement might be a serious problem to many readers. It was pointed out by Zadeh (26,27) that this has to be subjective. In fact, many readers will find it difficult to believe that "no objective definition of probabilities in terms of actual or possible observations, or possible properties of this world, is admissible' (a famous quote from Sir Harold Jeffreys in (11)). Thus if the information is supplied by a human being, he might as well supply the truth-value of it also. If the information is supplied by another machine, say a pattern recognizer, then a measure of uncertainty is often associated with it. For example, a pattern recognizer may decide that an incoming object is a meteor, not an airplane, with probability 0.9. Then this information may be assigned a truth-value of 0.9 since probability often reflects the degree of truth.

Perhaps the most useful theorem proved in this paper is Theorem 9. Suppose we have two sets of clauses, S_1 and S_2, representing two sets of information. From S_1, we can deduce a logical consequence which suggests us to use Highway 70S. From $S_?$, we can deduce a logical consequence which suggests to use Highway 495. We might not know the exact degrees of correctness of these suggestions. But we still can make intelligent decisions based on Theorem 9. For example, let $T(S_1)=0.6$ and $T(S_2)=0.9$. According to Theorem 9, we know that the correctness of the second suggestion is at least 0.9 while that of the first one is only guaranteed to be not less than 0.6. This further tells us that the risk of taking the first action may be as high as 0.4 while the other risk cannot be higher than 0.1. To minimize the maximum possible risk, we should take the second suggestion, i.e., use Highway 495.

In fact, no one in this world uses two-valued logic to solve problems such as baking a cake, buying a used car or hiring an employee. After all, every competent housewife knows what it means to "generously grease the pan" or "vigorously beat the mixture." If we really want to build a machine with common sense (16), is it not a must that some kind of many-valued logic be used?

References

1. Ackerman.R. Introduction to Many Valued Logics, Dover, N.Y., 1967.

2. Chang, C.C. Algebraic Analysis of Many Valued Logics, Trans, of the Amer. Math. Soc. 88, 1958, pp.467-490.

3. Chang,C.L. Fuzzy Topological Spaces, Journal of hath. Anal, and Appl. 24, 1, 1968, pp. 182-190.

4. Chang,C.L. and Lee, R.C.T. Symbolic Logic and Mechanical Theorem Proving, unpublished manuscript, 1971.

5. Chang,S.K. On the Execution of Fuzzy Programs Using Finite State Machines (to appear in IEEE Trans, on Computers).

6. Chang,S.K. Fuzzy Programs, Proceedings of the Brooklyn Polytechnical Institute Symposium on Computers and Automata, Vol. XXI, 1971.

7. Davis,M. and Putnam,H. A Computing Procedure for Quantification Theory, JACM, 1960, pp. 201-205.

8. Gnedenko,B.V. The Theory of Probability, translated from Russian by B.R.Seckler, Chelsea, N.Y., 1962.

9. Green,C. Application of Theorem Proving to Problem Solving, Proc. of the First International Joint Artificial Intelligence Conf., Washington, D.C. 1969.

10. Herbrand,J. Recherches sur la Théoria de la Demonstration, Traveaux de la Société des Sciences de Varsoria, No. 33, 1930.

11. Jeffreys,H. Theory of Probability, Oxford University Press, Oxford, England, 1961.

12. Lee,R.C.T. A Completeness Theorem and a Computer Program for Finding Theorems Derivable from Axioms, Ph.D. Thesis, Dept. of Elect. Engr. and Computer Sciences, University of Calif. Berkeley, 1967.

13. Lee,R.C.T. and Chang,C.L. Some Properties of Fuzzy Logic, Division of Computer Research and Technology, National Inst. of Health, 1970.

14. Lukasiewicz,J. Logical Foundations of Probability Theory, in Jan Lukasiewicz, Selected Works, edited by L. Berkowski, North-Holland Publishing Company, Amsterdam, 1970, pp. 16-43.

15. Marinos,P.N. Fuzzy Logic and Its Applications to Switching Systems, IEEE Trans. on Computers, c-18, 4, 1969, pp. 343-348.

16. McCarthy,J. Programs with Common Sense, Mechanization of Thought Process, Vol. 1, Her Majesty's Stationary Office, London, 1961, pp. 75-84.

17. McCarthy,J. and Hayes,P.J. Some Philosophical Problems from the Standpoint of Artificial Intelligence, Machine Intelligence 4, (B.Meltzer and D.Michie, Eds.), American Elsevier, N.Y., 1968, pp. 463-502.

18. Nilsson,N.J. Learning Machines, McGraw-Hill, N.Y., 1965.

19. Nilsson,N.J. Problem Solving Methods in Artificial Intelligence, McGraw-Hill, N.Y., 1971.

20. Robinson,J.A. A Machine-oriented Logic Based on the Resolution Principle, JACM 12, 1965, pp. 23-41.

21. Rosser,J.B. and Turquette.A.R. Many-valued Logics, North-Holland, Amsterdam, 1952.

22. Slagle.J.R. Interpolation Theorems for Resolution in Lower Predicate Calculus, JACM, 17, 1970, pp. 535-542.

23. Slagle.J.R. Artificial Intelligence, the Heuristic Programming Approach, McGraw-Hill N.Y., 1971.

24. Slagle,J.R., Chang.CL. and Lee,R.C.T. Completeness Theorems for Semantic Resolution in Consequence Finding, Proc. of the First International Joint Artificial Intelligence Conf., Washington, D.C, 1969.

25. WaTdinger.R. and Lee,R.C.T. PROW: A Step Toward Automatic Program Writing, Proc. of the First International Joint Artificial Intelligence Conf., Washington, D.C, 1969.

26. Zadeh,L.A. Fuzzy Sets, Information and Control, 8, 1965, pp. 338-353.

27. Zadeh,L.A. Fuzzy Algorithms, Information and Control, 12, 1968, pp. 94-102