Consistency Checking for Fuzzy Expert Systems

K. S. Leung and Y. T. So
The Chinese University of Hong Kong

ABSTRACT

Inconsistency frequently exists in a rule-based expert system. Detecting the existence of inconsistency in a fuzzy rule-based environment is difficult and may be different from that of traditional rule-based systems. An affinity measure, which is based on the similarity measure, is introduced to determine the likeness of two fuzzy terms. By using the affinity measure, the techniques for consistency checking in a non-fuzzy environment can be easily applied to a fuzzy environment. A consistency checker (CCFE) is implemented to detect possible inconsistency in a mixed fuzzy and non-fuzzy environment.

KEYWORDS: ***Expert system, fuzzy theory, inconsistency, possibility distribution***

1. INTRODUCTION

In the development of a rule-based expert system, the knowledge engineering process is an iterative one. The knowledge base needs to be refined many times. When new rules are added or the existing rules are altered, inconsistency in the knowledge base frequently occurs.

Although Hayes–Roth [1] pointed out that a missing key feature of rule-based systems is "a suitable verification methodology or a technique for testing the consistency of completeness of a rule set," many software tools such as TERIRESIAS [2], ONCOCIN's rule-checker [3], INSPECTOR [4], and ESC [5] have been developed to identify inconsistencies in non-fuzzy knowledge bases. The consistency checking tools help both the domain experts and knowledge engineers to build the expert systems more easily, accurately, and quickly.

However, checking inconsistency in a fuzzy rule-based environment [6–8] is not as simple as that in a non-fuzzy environment and may also be different from that of traditional rule-based systems. Let "height" be a

Address correspondence to K. S. Leung, Department of Computer Science, The Chinese University of Hong Kong, Shatin, N. T., Hong Kong.

Received April 30, 1991; accepted March 12, 1993.

fuzzy object having three fuzzy terms (tall, medium, short) as its possible values. If the user enters the following two rules:

Rule 1: IF height is tall THEN C

Rule 2: IF height is short THEN C

Does redundancy occur?

Consider another case:

Rule 3: IF A THEN height is more-or-less tall

Rule 4: IF A THEN height is not very tall

Does conflict occur?

Thus, it would be better if we could apply a quantitative measure on the two fuzzy terms (e.g., tall, not short) in order to determine to what extent these two fuzzy terms mean the same thing. An "affinity" measure based on the similarity measure [6] is created to determine the degree of matching between two fuzzy expressions. Using the affinity measure, the techniques for consistency check in a non-fuzzy environment can be transported to a fuzzy environment.

The following section summarizes different kinds of inconsistencies that exist in a rule-based environment. Section 3 introduces the similarity and affinity measures. Section 4 describes a consistency checker, CCFE, implemented for fuzzy rule-based expert systems. Its results in detecting inconsistency in a fuzzy environment are illustrated by an example and discussed in the fifth section.

2. INCONSISTENCY IN RULE-BASED SYSTEMS

There are several kinds of inconsistencies in a non-fuzzy rule-based environment [9–12]. The summary is given as follows:

Let us consider the rules:

Rule 1: IF $A1$ THEN $B1$ ($CF1$)

Rule 2: IF $A2$ THEN $B2$ ($CF2$)

where $A1$ and $A2$ may be any combination of propositions, $B1$ and $B2$ are single propositions, and $CF1$ and $CF2$ are the certainty factors, ranging from 0 to 1, of rule 1 and rule 2 respectively.

2.1. Redundant Rules

- Two rules succeed in the same situation and have the same results.
  i.e., i, $A1 = A2 \,\&\, B1 = B2 \,\&\, \mathrm{sign}(CF1) = \mathrm{sign}(CF2)$
  or ii, $A1 = A2 \,\&\, B1 \neq B2 \,\&\, \mathrm{sign}(CF1) \neq \mathrm{sign}(CF2)$
- The two rules may cause the same information to be counted twice, leading to erroneous increases in the certainty factor of the conclusion.

- Condition ii above may not be redundant if the expert wants to confirm one value (e.g., $B1$ with $CF1 > 0$) and at the same time disconfirm another value (e.g., $B2$ with $CF2 < 0$) of the conclusion.

2.2. Conflicting Rules (Contradiction)

- Two rules succeed in the same situation but with conflicting results.
  i.e. i, $A1 = A2 \,\&\, B1 = B2 \,\&\, \mathrm{sign}(CF1) \neq \mathrm{sign}(CF2)$
  or ii, $A1 = A2 \,\&\, B1 \neq B2 \,\&\, \mathrm{sign}(CF1) = \mathrm{sign}(CF2)$
- It is a common occurrence in rule sets. However, it may cause no problems (inconsistency) because the expert may want to conclude different values with different certainty factors.
- For example, the following are the rules extracted from a medical expert system ESROM [8]:

```
(Rule m17a                          (Rule m18
IF (((mdcs is anycd) OR             IF (((mdcs is anycd) OR
        (mdcs is gmd))                      (mdcs is gmd))
AND  (gestation >= 30)              AND        (gestation >= 30)
AND  (gestation <= 34))             AND        (gestation <= 34)
THEN management is delivery         THEN management is observation
Certainty is 0.7                    Certainty is 0.3
```

Management is a single-valued object with expected values (delivery, observation). The two rules listed above have the same antecedent part but with conflicting conclusions. However, they are set on purpose by the expert and by no means a mistake.

2.3. Subsumed Rules

- Two rules have the same result, but one contains additional restrictions on the situation in which it will succeed. Whenever the more restrictive rule succeeds, the less restrictive rule also succeeds, resulting in redundancy.
  i.e., $(A1 \subset A2 \text{ or } A2 \subset A1) \,\&\, B1 = B2 \,\&\, \mathrm{sign}(CF1) = \mathrm{sign}(CF2)$
  or $(A1 \subset A2 \text{ or } A2 \subset A1) \,\&\, B1 \neq B2 \,\&\, \mathrm{sign}(CF2) \neq \mathrm{sign}(CF1)$
- e.g., in ESROM

```
(Rule io1                      (Rule ii10
IF (diagnosis is unrupt)       IF ((diagnosis is unrupt)
THEN cx is uninf)              AND (ctg is reactive))
Certainty is 0.8               THEN cx is uninf)
                               Certainty is 0.95
```

Whenever rule ii10 is triggered, rule io1 will also be triggered. Thus, there is redundancy.

- However, the knowledge engineer may want to write these kinds of rules so that the more restrictive rules will add weight to the conclusions. Thus, the experts should be warned and requested to clarify their meanings. In the above example, if the expert's intention is to give more weight to "cx is uninf" when "ctg is reactive" is also true, then the certainty factor of rule ii10 should be changed to 0.75. If both "diagnosis is unrupt" and "ctg is reactive" are true, the certainty of conclusion "cx is uninf" becomes 0.95 after both rules are fired and the evidence combination calculation is applied [6].

2.4. Sub-contradiction Rules

It is a special case of contradiction. Two rules have different results, but one contains additional restrictions on the situation in which it will succeed. Whenever the more restrictive rule succeeds, the less restrictive rule also succeeds:

i.e., $(A1 \subset A2 \text{ or } A2 \subset A1) \,\&\, B1 \neq B2 \,\&\, \text{sign}(CF1) = \text{sign}(CF2)$

or $(A1 \subset A2 \text{ or } A2 \subset A1) \,\&\, B1 = B2 \,\&\, \text{sign}(CF2) \neq \text{sign}(CF1)$

2.5. Unnecessary IF Conditions

- Two rules have the same conclusion, an IF condition in one rule is in conflict with an IF condition in the other rule, and all other IF conditions in the two rules are equivalent.
  e.g., $(A1 = p \wedge q), (A2 = p \wedge \neg q), B1 = B2 \,\&\, CF1 = CF2$
- The example described above actually indicates that only one rule is necessary. The second IF conditions (q and $\neg q$) of both rules are unnecessary. Although it will not cause any error in consultation, it would be better to integrate the two rules into one: IF p THEN $B1(CF1)$
- e.g., $(A1 = p \wedge q), (A2 = \neg q), B1 = B2 \,\&\, CF1 = CF2$
  The second IF condition in the first rule is unnecessary, and the two rules could be combined into: IF $p \vee \neg q$ THEN $B1(CF1)$
- If $CF1 \neq CF2$, no unnecessary IF condition occurs because the expert may want to conclude a value with different certainty factors in different situations, e.g., in ESROM

```
(Rule m20
IF (((wcc > 10) AND (wcc < 15))
  OR ((crp > 20) AND (crp < 40)
       and (gestation >= 32)))
THEN management is delivery)
Certainty is 0.8
```

```
(Rule m21
IF (((wcc > 10) AND (wcc < 15))
  OR ((crp > 20) AND (crp < 40)
       and (gestation < 32)))
THEN management is delivery)
Certainty is 0.5
```

The condition (gestation >= 32) in rule m20 is conflicting with the condition (gestation < 32) in rule m21. However, the expert wants to assign different certainty factors to different situations.

2.6. Circular Rules

A set of rules is circular if the chaining of these rules in the set forms a cycle.

i.e., $A1 = B2$ & $A2 = B2$ & $\text{sign}(CF1) = \text{sign}(CF2)$

2.7. Self-referring Rules

The condition and conclusion clauses of a rule refer to the same subject.

e.g., IF $A = 0$ THEN $A = 1$ (for A is not a multi-valued object)

2.8. Inconsistent IF–Clause

The clauses in the condition are contradictory to each other.

e.g., IF $A = 1$ and $A = 2$ THEN B (for A is not a multi-valued object)

In 2.7 and 2.8, if A is a multi-valued object, it cannot be concluded that inconsistency occurs because a multi-valued object can have more than one value at anytime

The above discussion only mentions superficial inconsistency between two rules. However, inconsistency may arise after a sequence of inferring steps, e.g.,

Rule 1: IF A THEN B ($CF1 > 0$)
Rule 2: IF B THEN C ($CF2 > 0$)
Rule 3: IF C THEN D ($CF3 > 0$)
Rule 4: IF A THEN D ($CF4 > 0$)

Redundancy occurs between rule set (1–3) and rule 4.

Moreover, the detection of circular-rule chains is affected by the threshold. The certainty factors may cause a circular chain of rules to be "broken" if the certainty factor of conclusion falls below the threshold (e.g., 0.2).

e.g.,
Rule 1: IF A THEN B ($CF = 0.4$)
Rule 2: IF B THEN C ($CF = 0.7$)
Rule 3: IF C THEN D ($CF = 0.7$)
Rule 4: IF D THEN A ($CF = 0.8$)

As $(0.4)(0.7)(0.7) = 0.19 < 0.2$, the circular-rule chain is broken.

3. CONSISTENCY CHECKS IN A FUZZY ENVIRONMENT

First, let us consider the possibility, necessity, and similarity measures in possibility theory, and then the affinity measure of two fuzzy terms will be introduced. Let the fuzzy terms be represented by a list of 11 real numbers that are grades of membership of the points on an imaginary psychological continuum with an interval scale. The fuzzy terms "tall," "medium," and "short" are represented as follows:

tall:	0	0	0	0	0	0	.0280	.1176	.3277	.6561	1
medium:	0	0	0	.0588	.328	1	.328	.0588	0	0	0
short:	1	.6561	.3277	.1176	.0280	0	0	0	0	0	0

Hedges like "very" or "more-or-less" may be applied to the above fuzzy terms to form fuzzy expressions such as "very tall" and "more-or-less short." A square and a square root functions are the fuzzy concepts handling functions for the hedges "very" and "more-or-less" respectively [13]. The modified representations are shown:

very tall:	0	0	0	0	0	0	.0008	.0138	.1074	.4305	1
more-or-less short:	1	.8100	.5725	.3429	.1673	0	0	0	0	0	0

The membership values described above can be viewed as a possibility distribution [14]. But this distribution has only an indicative or subjective meaning and may not be interpreted directly. Figures 1–3 depict the curves of the above fuzzy terms/expressions. Note that the curves are normalized and convex.

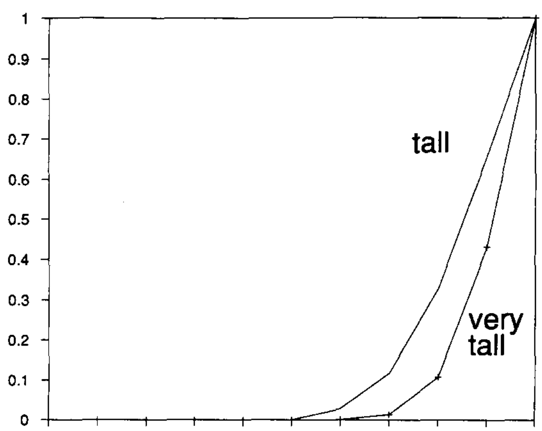

Figure 1. Possibility distribution for the concepts "tall" and "very tall."

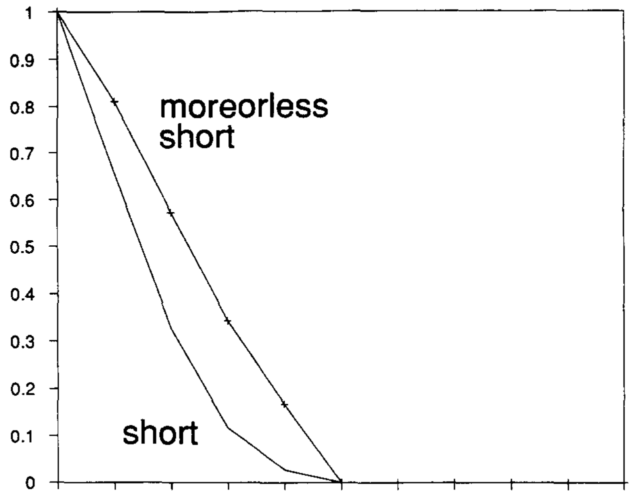

Figure 2. Possibility distribution for the concepts "short" and "more-or-less" short.

3.1 Possibility and Necessity

The formulae of the possibility (P) and necessity (N) measures between a fuzzy datum and a fuzzy pattern are given as follows [15]:

$$P(F|F') = \max(\min(\mu_F(w), \mu_{F'}(w)))$$
$$N(F|F') = 1 - P(\sim F|F')$$

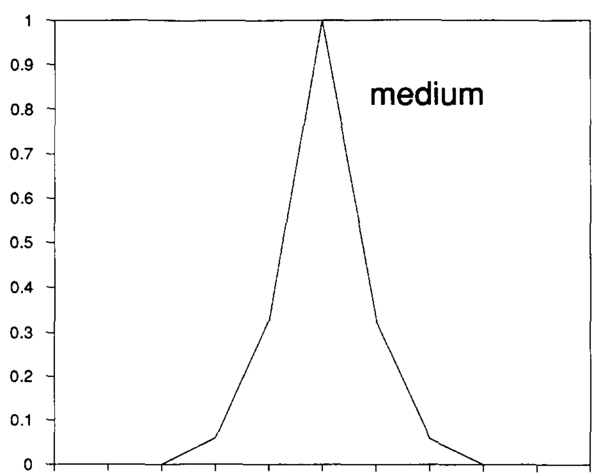

Figure 3. Possibility distribution for the concept "medium."

where

$P(F|F')$ is the possibility of the fuzzy datum F' given the fuzzy pattern F, and
$N(F|F')$ is the necessity of the fuzzy datum F' given the fuzzy pattern F.
$\mu(w)$ is the membership function of w in the universe of discourse, and
$\sim F$ is the complement of F.

The possibility measures the degree of overlapping between the datum and the pattern; or in other words, it measures the non-emptiness of $F \cap F'$.

For example:

$P(\text{tall}|\text{very tall}) = 1$

$P(\text{short}|\text{more-or-less short}) = 1$

$P(\text{tall}|\text{medium}) = 0.028$

$P(\text{short}|\text{tall}) = 0$

Possibility is an optimistic measure: it only means something is possible but does not guarantee it will happen. On the other hand, necessity is a drastic measure that is equal to the impossibility between two opposite events.

For example:

$N(\text{tall}|\text{very tall}) = 0.6561$

$N(\text{short}|\text{more-or-less short}) = 0.4275$

$N(\text{tall}|\text{medium}) = 0$

$N(\text{short}|\text{tall}) = 0$

Under normal circumstances, the necessity has the following properties:

IF $N(F|F') > 0.5$ THEN

F' has a more concentrated or narrower distribution than F.

e.g., tall and very tall.

IF $N(F|F') < 0.5$ THEN

F' has a more dilated or broader distribution than F.

e.g., short and more-or-less short,

tall and medium.

IF $N(F|F') = 0.5$ THEN

F has the same distribution as F.

Also from the definition of both possibility and necessary, it can be shown that:

i, $P(A|B) = P(B|A)$ ....(1)
ii, $P(A|B \cup C) = \max(P(A|B), P(A|C))$ ....(2)
iii, $N(A|B \cup C) = \min(N(A|B), N(A|C))$ ....(3)
iv, $N(A|A) >= 0.5$ ....(4)

3.2. Similarity

The similarity M is calculated by the following algorithm [6]:

IF $N(F|F') \geq 0.5$
THEN $M = P(F|F')$
ELSE $M = (N(F|F') + 0.5)*P(F|F')$

where * denotes multiplication.

That means when necessity is greater than or equal to 0.5, the similarity between pattern and datum is saturated and is forced to equal the possibility. If the similarity is not saturated (necessity is less than 0.5), the similarity should depend on both possibility and necessity.

The similarity measure is proven to be a useful tool to determine the similarity between the fuzzy data in a fact base and the fuzzy patterns in the premise part of a rule [8]. However, there is one shortcoming in similarity; it is not commutative, i.e, $M(p|q) \neq M(q|p)$.

For example,

$P(\text{not medium}|\text{tall}) = 1$ and $P(\text{medium}|\text{tall}) = 0.0588$
$\Rightarrow N(\text{not medium}|\text{tall}) = 1 - 0.0588 = 0.9412$

Therefore, $M(\text{not medium}|\text{tall}) = 1$. But

$P(\text{tall}|\text{not medium}) = 1$ and $P(\text{not tall}|\text{not medium}) = 1$
$\Rightarrow N(\text{tall}|\text{not medium}) = 1 - 1 = 0$

Therefore, $M(\text{tall}|\text{not medium}) = 0.5$.

Hence, the order of rule checking will influence the value of similarity. Owing to this limitation, the affinity (A) measure is introduced.

3.3. Affinity

The Affinity of two simple propositions p and q is defined as follows:

$$A(p,q) = M(p \wedge q | p \vee q)$$

where $\mu_{p \wedge q}(w) = \min(\mu_p(w), \mu_q(w))$,

$\mu_{p \vee q}(w) = \max(\mu_p(w), \mu_q(w))$ and

$\mu(w)$ is the membership function of w in the universe of discourse.

$A(p, q)$ measures the similarity of $p \vee q$ given $p \wedge q$. That is, to what extent does the whole part of both p and q ($p \vee q$) match the shared part of them ($p \wedge q$). The following are some properties of the affinity measures:

i, If p and q are identical, $A(p, q)$ will be equal to 1 (because $p \vee q = p \wedge q$).

ii, $A(p, q)$ is commutative, i.e., $A(p, q) = A(q, p)$.
Because $p \wedge q = q \wedge p$ and $p \vee q = q \vee p$.

iii, $A(p, q) = \min(M(p|q), M(q|p))$ ....(5)

Proof of assertion (iii):

Consider the possibility first

$$\begin{aligned} & P(p \wedge q | p \vee q) \\ &= \max\big(\min(\mu_{p \wedge q}(w), \mu_{p \vee q}(w))\big) \\ &= \max(\mu_{p \wedge q}(w)) \\ &= \max\big(\min(\mu_p(w), \mu_q(w))\big) \\ &= P(p|q) \\ &= P(q|p) && \text{[by eqn (1)]} \end{aligned}$$

....(6)

and

$$\begin{aligned} & N(p \wedge q | p \vee q) \\ &= \min(N(p \wedge q | p), N(p \wedge q | q)) && \text{[by eqn (3)]} \end{aligned}$$

Now consider

$$\begin{aligned} & N(p \wedge q | p) \\ &= 1 - P(\sim (p \wedge q) | p) \\ &= 1 - P(p | \sim (p \wedge q)) && \text{[by eqn (1)]} \\ &= 1 - P(p | \sim p \vee \sim q) \\ &= 1 - \max(P(p | \sim p), P(p | \sim q)) && \text{[by eqn (2)]} \\ &= 1 - \max(P(\sim p | p), P(\sim q | p)) && \text{[by eqn (1)]} \\ &= \min(N(p|p), N(q|p)) \end{aligned}$$

Similarly,

$$\begin{aligned} & N(p \wedge q | q) \\ &= \min(N(q|q), N(p|q)) \end{aligned}$$

So that

$$N(p \wedge q|p \vee q)$$
$$= \min(N(p|p), N(p|q), N(q|p), N(q|q)) \quad \text{....(7)}$$

By eqn (4), it can be concluded that $N(p|p)$ and $N(q|q)$ are both greater than or equal to 0.5.

Now, we consider all four cases:

case 1:

If both $N(p|q)$ and $N(q|p) >= 0.5$,
then $M(p|q) = P(p|q) = P(q|p) = M(q|p) =$
$\min(M(p|q), M(q|p))$ [by eqn (6)]
and $N(p \wedge q|p \vee q >= 0.5$ [by eqn (7)]

Thus $A(p, q) = M(p \wedge q|p \vee q) = P(p \wedge q|p \vee q) = P(p|q) = P(q|p) = \min(M(p|q), M(q|p))$ [by eqn (6)]

Thus, the assertion (eqn (5)) holds.

case 2:

If $N(p|q) < 0.5$ and $N(q|p) > 0.5$,
then $N(p \wedge q|p \vee q) = N(p|q) < 0.5$
and $A(p, q) = (N(p|q) + 0.5)*P(p|q) = M(p|q)$ (minimum of $M(p|q)$ and $M(q|p)$)

case 3:

If $N(q|p) < 0.5$ and $N(p|q) > 0.5$,
then $N(p \wedge q|p \vee q) = N(q|p) < 0.5$
and $A(p, q) = (N(q|p) + 0.5)*P(q|p) = M(q|p)$ (minimum of $M(p|q)$ and $M(q|p)$)

case 4:

If both $N(p|q)$ and $N(q|p) < 0.5$,

then $N(p \wedge q|p \vee q) = \min(N(p|q), N(q|p))$

So $A(p, q)$

$= M(p \wedge q|p \vee q)$

$= (N(p \wedge q|p \vee q) + 0.5)*P(p \wedge q|p \vee q)$

$= (\min(N(p|q), N(q|p)) + 0.5)*P(p|q)$ [by eqns (6) & (7)]

$= \min(M(p|q), M(q|p))$

Therefore, $A(p, q) = \min(M(p|q), M(q|p))$ holds for all cases. ■

Implementing $A(p,q)$ using our definition (ie., $A(p,q) = M(p \wedge q | p \vee q)$) will result in two comparisons, two additions, and one multiplication less than that of using $A(p,q) = \min(M(p|q), M(q|p))$ each time. Hence the former is actually used for implementation in our system.

Among the above properties, the commutative one is the most important because it implies that the order of rule-checking has no influence on the value of the affinity.

In order to investigate the suitability of affinity in consistency checks, we compute all the results of the affinity for different values of p and q on a fuzzy object such as "height." Hedges like "very" and "more-or-less" and logical negation "not" are added to the possible fuzzy terms of the object. So the values of p and q could be

[not][very|more-or-less]{tall|medium|short}

where the choice inside the [ ] is optional. (Although the expressions like "more-or-less medium" or "very medium" are not very meaningful, we include them to complete the investigation.)

The results of the affinity of any two fuzzy expressions p and q can be divided into four groups.

(1) $A(p,q) = 1$
e.g., A(tall, tall) = 1
A(very medium, medium) = 1

(2) $A(p,q) > 0.5$
e.g., A(tall, very tall) = 0.9305
A(more-or-less short, short) = 0.9275
A(more-or-less medium, very medium) = 0.9273

(3) $A(p,q) = 0.5$, $P(p \wedge q | p \vee q) = 1$ and $N(p \wedge q | p \vee q) = 0$
e.g., A(tall, not medium) = 0.5
A(not short, medium) = 0.5
A(not very tall, more-or-less short) = 0.5

(4) $A(p,q) < 0.5$, $P(p \wedge q | p \vee q) < 1$ and $N(p \wedge q | p \vee q) = 0$
e.g., A(tall, not tall) = 0.1720
A(more-or-less tall, medium) = 0.0837
A(short, tall) = 0

The affinities of groups (1) and (2) are both greater than 0.5. This means that their degrees of matching are greater than their differences. Therefore, we conclude that the two fuzzy concepts being compared are the same or similar (to some extent). The affinities in group (4) are less than 0.5, we say that the two terms are different. The affinities in group (3) equal to 0.5 which means that the truth and falsity of the likeness between two fuzzy terms cannot be indicated by the affinity measure. As a conclusion, if the affinity between two fuzzy terms is greater than 0.5, they can be treated as equal. For case (3), we could swing to either side depending on the implementation.

The definition of affinity could be extended to detect the likeness of the antecedent part of two rules.

Let *P1* and *P2* be the antecedent parts of two rules that contain multiple propositions connected by AND operators, i.e.,

$$P1 = p_{11} \wedge p_{12} \wedge p_{13} \wedge \cdots \wedge p_{1k}$$
$$P2 = p_{21} \wedge p_{22} \wedge p_{23} \wedge \cdots \wedge p_{2k}$$
$$\text{Then } A(P1, P2) = \prod_{k=1}^{n} A(p_{1k}, p_{2k})$$

If p_{1k} and p_{2k} are non-fuzzy and $p_{1k} = p_{2k}$, then $A(p_{1k}, p_{2k}) = 1$; otherwise, $A(p_{1k}, p_{2k}) = 0$. The affinity of an antecedence is equal to the product of the affinity of its propositions.

4. A CONSISTENCY CHECKER IN FUZZY ENVIRONMENT

Using affinity, we could detect inconsistency in a fuzzy environment. The detection methods can be the same as those used in a nonfuzzy environment except that we use affinity to measure the degree of matching of two fuzzy expressions.

i.e., $(P1 = P2)$ and $(P1 \neq P2)$ could be replaced by $A(P1|P2) \geq 0.5$ and $A(P1|P2) < 0.5$ in a fuzzy environment respectively.

A consistency checker in a fuzzy environment (CCFE) has been implemented to check the rule bases built from a fuzzy expert system shell Z-III that allows any mix of fuzzy and non-fuzzy terms in the rules [8]. CCFE is implemented using Turbo C 2.0 in a microcomputer environment. Affinity measure is employed to determine the degree of matching of two fuzzy propositions. At the first stage of the development of CCFE, we set the following assumptions:

(a) Only one proposition is allowed in the consequence part of each rule.

(b) Disconfirmation of a consequent proposition is represented by a negative certainty factor to a rule.

i.e., For the rule:

IF height is tall
THEN weight is not light
certainty is 0.8

can be represented as follows:

IF height is tall
THEN weight is light
certainty is -0.8

(c) Single-valued objects are assumed.
(d) CCFE only searches for the following most common inconsistencies:
(i) Redundancy
Let:
rule 5: IF *P1* THEN *C1* (*CF1*)
rule 6: IF *P2* THEN *C2* (*CF2*)
where *P1* and *P2* may be any combination of propositions, and CF stands for the certainty factor of a rule.

Redundancy may occur when:

$$A(P1|P2) \geq 0.5 \,\&\, A(C1|C2) \geq 0.5 \,\&\, \mathrm{sign}(CF1) = \mathrm{sign}(CF2)$$

$$\text{or } A(P1|P2) \geq 0.5 \,\&\, A(C1|C2) < 0.5 \,\&\, \mathrm{sign}(CF1) \neq \mathrm{sign}(CF2)$$

(ii) Contradiction
Contradiction may occur when:

$$A(P1|P2) \geq 0.5 \,\&\, A(C1|C2) \geq 0.5 \,\&\, \mathrm{sign}(CF1) \neq \mathrm{sign}(CF2)$$

$$\text{or } A(P1|P2) \geq 0.5 \,\&\, A(C1|C2) < 0.5 \,\&\, \mathrm{sign}(CF1) = \mathrm{sign}(CF2)$$

(iii) Subsumption
A special case of redundancy. The details have been described in section 2.
(iv) Sub-contradiction
A special case of contradiction. The details have been described in section 2.

A decision table is used for verification of the rule base. A decision table facilitates the testing of a set of rules for inconsistent conditions. Each column of a decision table represents the conditions of a rule. Building a decision table from the rules is relatively simple [5]. CCFE creates a decision table by parsing the rules stored in the rule base, entering each corresponding value in the appropriate row and column. The table is implemented using arrays of pointers. Thus, the storage for the table is dynamically allocated. Besides, a dynamically allocated one-dimensional array is created to store the certainty factor of each rule.

Table 1. A Sample Rule Set

```
(rule r1
 (if (height is tall
 then weight is heavy)
certainty is 0.7

(rule r2
 (if (height is tall)
 and (eating is very much)
 then weight is very heavy)
certainty is 0.65

(rule r3
 (if (eating is much)
 and (activity is little)
 then weight is moreorless heavy)
certainty is 0.9
```

The algorithm for creating the decision table and the array storing the certainty factors is as follows:

```
Read all the rules from a file into the rule base.
Count the total number of objects and rules in the rule base.
Allocate spaces for the table and the certainty factors.
Initialise all the cells in the table to UNOCCUPIED.

For all the rules in the rule base
    Parse the rule and enter the values to the corresponding row and
      column.
    Set that cell to OCCUPIED.
```

For instance, from the rules in Table 1, CCFE will create the corresponding decision table and the array storing certainty factors as shown in Figure 4.

	r1	r2	r3
height	tall	tall	
eating		very much	much
activity			little
weight	heavy	very heavy	moreorless heavy

Certainty factor	0.7	0.65	0.9

Figure 4. The sample decision table.

After constructing the decision table, all the rules are checked for inconsistencies. Affinity measure is employed to determine the likeliness between two expressions, either fuzzy or non-fuzzy. The following is the algorithm for consistency checking:

```
For i = 1 to no. of rules − 1
    For j = i + 1 to no. of rules
        case compare(antecedence(i), antecedence(j))

        EQUAL:
            if (same(consequence(i) , consequence(j)) then
                if (sign(CF(i)) = sign(CF(j))) then
                    report redundancy
                else
                    report contradiction
            else
                if (sign(CF(i)) = sign(CF(j))) then
                    report contradiction
                else
                    report redundancy
            endif
            continue

        SUBSET:
            if (same(consequence(i) , consequence(j)) then
                if (sign(CF(i)) = sign(CF(j))) then
                    report subsumption
                else
                    report sub-contradiction
            else
                if (sign(CF(i)) = sign(CF(j))) then
                    report sub-contradiction
                else
                    report subsumption
            endif
            continue

        DIFFERENT:
            continue
        endcase
    endfor
endfor
```

where *compare* is a function using affinity measure to determine whether two antecedences are the SAME, DIFFERENT, or one of the antecedence is a SUBSET of the other.

same is a Boolean function using affinity measure to determine whether two consequences are the same.

sign returns the sign of the certainty factor.

5. RESULTS AND DISCUSSIONS

CCFE is used to verify the consistency of fuzzy rule bases. The following rule base with fuzzy and non-fuzzy objects is verified by CCFE and the results are presented. (The example is an inconsistent testing case, so some rules may not make too much sense.)

Let

height is a fuzzy object with possible values (tall, medium, short).

weight is a fuzzy object with possible values (heavy, fit, light).

eating is a fuzzy object with possible values (much, balanced, little).

sleeping is a non-fuzzy single-valued object with possible values (enough, fair, bad).

sport is a non-fuzzy single-valued object with possible values (big-2, football, basketball).

Rule Base:

(rule r1
If (height is tall)
then weight is heavy) Certainty is 0.7

(rule r2
if ((height is tall)
and (eating is much))
then weight is very heavy) Certainty is 0.75

(rule r3
if (height is tall)
then weight is balanced) Certainty is −0.5

(rule r4
If ((eating is much)
and (sport is big-2))
then weight is very heavy) Certainty is 0.6

(rule r5
if (eating is very much)
then weight is light) Certainty is 0.5

(rule r6
if (eating is very much)
and (sport is big-2)
and (sleeping is not bad)
then weight is very heavy) Certainty is −0.7

(rule r7
if (eating is much)
and (sport is big-2)
and sleeping is enough)
then weight is heavy) Certainty is 0.6

(rule r8
If (sleeping is not enough)
then weight is light) Certainty is 0.5

(rule r9
If (sleeping is bad)
then weight is light) Certainty is 0.4

Diagnosis:

Warning 1: Subsumption between rules r1 and r2
Warning 2: Redundancy between rules r1 and r3
Warning 3: Subsumption between rules r2 and r3
Warning 4: Sub-contradiction between rules r2 and r5
Warning 5: Sub-contradiction between rules r4 and r5
Warning 6: Sub-contradiction between rules r4 and r6
Warning 7: Subsumption between rules r4 and r7
Warning 8: Subsumption between rules r5 and r6
Warning 9: Sub-contradiction between rules r5 and r7
Warning 10: Contradiction between rules r6 and r7
Warning 11: Redundancy between rules r8 and r9
Total, 11 warnings.

From the results, we can see that affinity is useful to determine the degree of matching of two fuzzy expressions for consistency checking. The techniques used in a non-fuzzy environment can be easily transported to a fuzzy environment. However, in an environment full of mixtures of approximate and exact reasonings, there are no absolute definitions of consistency as discussed in sections 2 and 3. A fuzzy consistency checker such as the one implemented can only serve as a provider for warnings of possible inconsistency to aid better knowledge engineering. The levels of warnings can be adjusted by varying parameters such as the threshold affinity in the definition and levels of tracking.

6. CONCLUSION

Using affinity, it has been demonstrated that we can detect possible inconsistency in a fuzzy environment. The detection techniques are very similar to those used in a non-fuzzy environment except that we use affinity to measure the likeness of two fuzzy expressions, i.e., for two non-fuzzy expressions $A1$ and $A2$, $A1 = A2$ and $(A1 \neq A2)$ could be replaced by $A(A1, A2) \geq 0.5$ and $A(A1, A2) < 0.5$ respectively for fuzzy expressions $A1$ and $A2$ in a fuzzy environment. The commutative property of affinity makes the checking order of new and old rules totally independent. The affinity measure has been proven to be a good tool to measure the likeness of two fuzzy terms. The consistency checker CCFE has been successfully implemented for aiding knowledge engineering in a heterogeneous fuzzy environment with mixed fuzzy and non-fuzzy terms.

References

1. Hayes-Roth, F., Rule-based systems, *Communication of ACM*, 28(9), 921–932, 1985.
2. Shortliffe, E. H., *Computer-Based Medical Consultations: MYCIN*, Elsevier, New York, 1976.
3. Suwa, M., Scott, A. C., and Shortliffe, E. H., An approach to verifying completeness and consistency in a rule-based expert system, *AI Magazine*, 3(3), 16–21, 1982.
4. KES General Description Manual, Software Architecture and Engineering, Inc., Arlington, V.A., 33, 1983.
5. Cragun, B. J., and Steudel, H. J., A decision-table-based processor for checking completeness and consistency in rule-based expert systems, *Int. J. Man–Machine Studies*, 26, 633–648, 1987.

6. Leung, K. S., and Lam, W., Fuzzy concepts in expert systems, *IEEE Computer*, 21(9), 43–56, Sept. 1988.

7. Leung, K. S., and Lam, W., A fuzzy expert system shell using both exact and inexact reasoning, *J. Automated Reasoning*, 5(2), 207–233, 1989.

8. Leung, K. S., So, Y. T., Leung, A., and Wong, W. S. F., Z-III: A PC-based fuzzy expert system shell, *Proc. Int. Computer Symp.*, 657–662, 1990.

9. Beauvieux, A., and Dague, P., Interactive checking of knowledge base consistency, *Int. Computer Sci. Conf.* 567–574, 1988.

10. Buchannan, B. G., and Shortliffe, E. H., Knowledge Engineering, in *Rule-Based Expert Systems*, Addisley-Wesley Publishing Company, 1984.

11. Nguyen, T. A., Perkins, W. A., Laffey, T. J., and Pecora, D., Knowledge base verification, *AI Magazine*, 69–75, summer 1987.

12. Tsang, W. W., Wan, Y. S., Lim, E. L., and Hioe, W., A space searching method for checking the consistency and completeness of a rulebase, *Int. Computer Sci. Conf.* 575–579, 1988.

13. Zadeh, L. A., A fuzzy-set-theoretic interpretation of linguistic hedges, *J. Cybernetics*, 2(3), 4–34, 1972.

14. Zadeh, L. A. PRUF—A meaning representation language for natural languages, *Int. J. Man-Machine Studies*, 10, 395–460, 1978.

15. Cayrol, M., Farreny, H., and Prade, H., Fuzzy Pattern Matching, *Kybernetes*, 11, 103–116, 1982.