

N93-29522

56-63

161806

P-10

## APPROXIMATE REASONING USING TERMINOLOGICAL MODELS

John Yen  
Center for Fuzzy Logic and Intelligent Systems  
Dept. of Computer Science  
Texas A&M University  
College Station TX 77843  
yen@cs.tamu.edu

Nitin Vaidya  
Texas Transportation Institute  
Texas A&M University  
701 North Post Oak #430  
Houston TX 77024  
nvaidya@cs.tamu.edu

### ABSTRACT

Term Subsumption Systems (TSS) form a knowledge representation scheme in AI that can express the defining characteristics of concepts through a formal language that has a well-defined semantics and incorporates a reasoning mechanism that can deduce whether one concept subsumes another. However, TSS have very limited ability to deal with the issue of uncertainty in knowledge bases. The objective of this research is to address issues in combining approximate reasoning with term subsumption systems. To do this we have extended an existing AI architecture (CLASP), that is built on the top of a term subsumption system (LOOM), in the following ways. First, the assertional component of LOOM has been extended for asserting and representing uncertain propositions. Second, we have extended the pattern matcher of CLASP for plausible rule-based inferences. Third, an approximate reasoning model has been added to facilitate various kinds of approximate reasoning. And finally, the issue of inconsistency in truth values due to inheritance is addressed using justification of those values. This architecture enhances the reasoning capabilities of expert systems by providing support for reasoning under uncertainty using knowledge captured in TSS. Also, as definitional knowledge is explicit and separate from heuristic knowledge for plausible inferences, the maintainability of expert systems could be improved.

### 1. INTRODUCTION

Knowledge exists in a variety of forms [1]. While most existing expert systems employ one or two knowledge representation schemes, expressing diverse knowledge in such a limited number of representation formalisms is difficult and time-consuming. Furthermore, it may not be possible to express completely all the knowledge required in an expert system. So, there is a need to integrate different knowledge representation schemes and to deal with the issue of incompleteness in a knowledge base. The objective of this research is to address these issues by combining two knowledge representation schemes, approximate reasoning and terminological reasoning.

Approximate reasoning concerns uncertain knowledge and data in expert systems. Uncertainty in expert systems may arise because of incompleteness in data, unreliability of data, impreciseness of data, or even uncertain knowledge. For example, judgmental knowledge used in medical expert systems is uncertain in nature. Hence, expert systems need to handle uncertainty in such a way that the conclusions are understandable and interpretable by the user [10]. In approximate reasoning, fuzzy logic makes it possible to deal with different types of uncertainty within a single framework as it subsumes predicate logic. It is suitable for inferring from imprecise knowledge as all uncertainty is allowed to be expressed as a matter of degree [22]. In addition fuzzy logic provides suitable operators for the combination of uncertainty, including a generalized modus ponens following from Zadeh [22] for making inferences based on rules.

Term Subsumption Systems (TSS), on the other hand, deal with terminological (i.e. definitional) knowledge. The representation scheme of term subsumption systems can express the defining characteristics of concepts through a formal language that has a well-defined semantics. The semantics of constructs that are often used to define concepts or roles are shown in Figure 1. Term subsumption systems provide a natural organization for terminological knowledge [3] through a structured taxonomy of conceptual entities with associated descriptions, which satisfy certain restrictions as well as have specific relationships to each other and where specific concepts can indirectly inherit characteristics from more general concepts. A guiding principle is that concepts are formal representational objects and that the epistemological relationships between formal objects must be kept distinct from the things represented by these formal objects [2]. For example the concept **Rich-Person** must be kept separate from an instance of **Rich-Person**. An example of terminological knowledge is shown in figure 2. In addition, the reasoning mechanism in these languages can deduce whether one concept subsumes another [12]. An automatic classifier places a concept in its proper location in a taxonomy so as to enforce network semantics

and consistency checking of logical subsumption relations between concepts [13]. Term subsumption systems originate from the ideas presented in the KL-ONE knowledge representation system, which was itself derived from semantic network formalisms [7]. Because of the formal semantics employed, term subsumption systems can be viewed as a generalization of frames and semantic networks [6], [17].

In this work we have extended two terminological architectures for approximate reasoning; LOOM and CLASP, which is built on top of LOOM. This paper has partly originated from Yen and Bonissone, who have both addressed the issue of extending TSS for uncertainty management and outlined a generic architecture in [19], and has been derived from Vaidya in [16].

Expression e	Interpretation [e]
:primitive $C_1$	a unique primitive concept
:primitive $R_1$	a unique primitive relation
(:and $C_1 C_2$ )	$\lambda x. [C_1] \wedge [C_2]$
(:and $R_1 R_2$ )	$\lambda xy. [R_1] \wedge [R_2]$
(:at-least 1 R)	$\lambda x. \exists y. [R](x,y)$
(:exactly 1 R)	$\lambda x. \exists y. [R](x,y) \wedge \forall yz. ([R](x,y) \wedge [R](x,z)) \rightarrow y = z$
(:all R C)	$\lambda x. \forall y. [R](x,y) \rightarrow [C](y)$
(:domain C)	$\lambda xy. [C](x)$
(:range C)	$\lambda xy. [C](y)$

Figure 1. Semantics of Some Terminological Expressions

```
(defconcept RICH-PERSON :is :p)
(defconcept MILLIONAIRE :is (:and :p RICH-PERSON))
(defconcept BILLIONAIRE :is (:and :p MILLIONAIRE))
(defconcept CAR :is :p)
(defconcept NEW-CAR :is (:and :p CAR ))
(defrelation HAS-CAR :is (:and :p (:domain PERSON) (:range CAR)))
```

Figure 2. An Example of Terminological Knowledge

## 2. ISSUES IN APPROXIMATE REASONING WITH TERMINOLOGICAL MODELS

In this section we outline four issues that need to be addressed in integrating approximate reasoning with terminological systems. This paper will focus on the first three issues. The fourth issue has been addressed in [18].

(1) *Extending the assertional component of a TSS for stating uncertain propositions:* One form of uncertainty in TSS concerns the uncertainty about the "instance of" relation. For example if there is a concept **Rich-Person**, a person may be a **Rich-Person** only to a certain extent. This issue concerns representing and asserting uncertain propositions and requires extension of the assertional component of TSS (often referred as the ABox).

(2) *Maintaining consistency of truth values associated with propositions:* Another issue needs to be dealt with. This is related to the inheritance of concepts. The truth value of an instance in a concept may be inconsistent with the truth value of the same instance in another concept which subsumes the first concept or is subsumed by the first concept. For example, the degree of membership in the concept **Millionaire** can not be lower than the degree of membership in the concept **Billionaire** as a **Millionaire** subsumes a **Billionaire**. As such, a truth

maintenance mechanism is required to maintain consistency of truth values of the propositions.

(3) *Extending the semantic pattern matcher for partial matching:* Another form of uncertainty could occur in the judgmental knowledge for reasoning with assertional components of term subsumption systems. For example, an owner of a new car may or may not be a rich person. From experts or from statistical data we may obtain a number to represent the likelihood that a person who owns a new car is also a rich person. This second issue concerns integration of such uncertainty with the uncertainty represented in the assertional component of the term subsumption language. For this purpose the Semantic Pattern Matcher of CLASP needs to be modified for performing partial matching of conditions.

(4) *Extending the semantics of terminological component of TSS for making plausible inferences:* This issue concerns the representation of and reasoning with uncertainty in the terminological knowledge of term subsumption systems.

### 3. INTEGRATING APPROXIMATE REASONING WITH TERMINOLOGICAL MODELS

The architecture for integrating approximate reasoning with term subsumption systems is an extension of the architecture of CLASP. For incorporating approximate reasoning the architecture has extended LOOM to include a representation scheme for uncertain propositions, a fuzzy assertional language for asserting and retracting such propositions, a fuzzy truth maintenance system and an assertion processor. Moreover, the architecture has extended CLASP and provides for representation of uncertain rules, a fuzzy rule language, a modification to the semantic pattern matcher of CLASP for partial matches and an approximate reasoner which reasons with the uncertainty expressed in instances and rules. The architecture is represented in Figure 3.

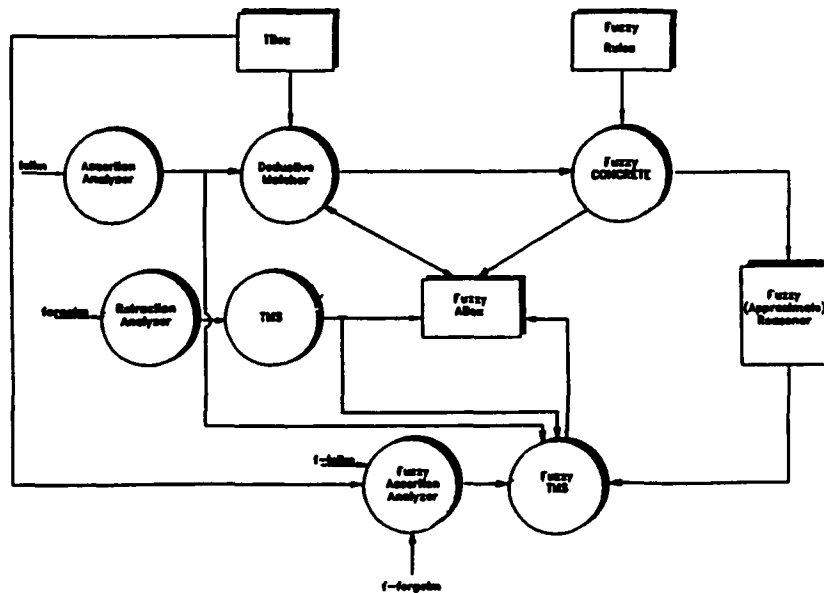


Figure 3. Architecture for Approximate Reasoning Using Terminological Models

#### 3.1 Extended Assertional Component

##### 3.1.1 A Fuzzy Assertional Language

The extended assertional language includes a truth value which expresses the degree of certainty of the membership of an instance in the corresponding concept or role. Please refer to Figure 5 for examples of the assertional language. It may be noted that the *f-tellm* statement causes assertion of propositions, whereas the *f-forgetm* statement causes retraction of propositions.

### 3.1.2 Internal Representation

The internal representation has been extended to include a representation for uncertainty in instances and also includes a justification structure for uncertainty. This representation scheme is the basis for truth maintenance and reasoning in the system. An example of internal representation of an instance is given in Figure 4.

---

```
(Instance(John)
.
(fuzzy-db-type: ((Rich-Person 0.5))
(fuzzy-role: ((Has-Car Mercedes) 0.7))
.
(justification-for-uncertainty:
  ( (RoleOrConcept: Rich-Person
    Certainty-Measure: 0.5
    Origin: Rule<New-Car-Owners-Are-Rich>)
    (RoleOrConcept: (Has-Car Mercedes)
    Certainty-Measure: 0.7
    Origin: "USER")
  )
)
)
```

Figure 4. Example of Internal Representation of an Instance.

---

## 3.2 Fuzzy Truth Maintenance System (FTMS)

The Fuzzy Truth Maintenance System (FTMS) performs consistency checking for truth values on all assertions, retractions and inferences.

### 3.2.1 Consistency Checking for Truth values of Propositions

An fuzzy proposition in a fuzzy TSS needs to be checked for consistency because the truth value of a fuzzy proposition may be constrained by the truth values of other fuzzy propositions. The truth value representing the degree of membership of an instance in a concept needs to be compared with the truth values for the same instance in other concepts below or above C in the concept subsumption hierarchy. Such a comparison is based on the following two general principles:

- (1) The truth value of an instance in a concept C cannot be greater than the truth value of the same instance in any of C's parent concepts.
- (2) The truth value of an instance in a concept C cannot be less than the truth value of the same instance in any of C's children concepts.

In summary, if  $C_1 > C_2$  then  $\mu_{C_1}(x) \geq \mu_{C_2}(x)$  where ">" denotes the subsumption relation between concepts.

To illustrate the above, assume concept  $C_1$  subsumes concept  $C_2$ , which subsumes concept  $C_3$ . Now if an instance has a degree of membership  $a_1$  in  $C_1$ ,  $a_2$  in  $C_2$  and  $a_3$  in  $C_3$ , then the condition  $a_1 \geq a_2 \geq a_3$  must be satisfied. Any assertion or retraction that result in truth values that violate this condition is inconsistent. An example of inconsistency is shown in Figure 5.

There are a number of sources that may cause inconsistency in truth values of data. Because inconsistencies due to different sources need to be handled differently we list possible sources of inconsistency below:

- (1) Inconsistency due to deduction based on the terminological model.
- (2) Inconsistency due to an assertion or retraction by the user.
- (3) Inconsistency due to an inference made by fuzzy rules.

---

Refer to terminological knowledge in Figure 2 and fuzzy-rule in Figure 7.  
Consider the following sequence of assertions:

(f-tellm ((Has-Car John Mercedes) 0.7))  
(f-tellm ((New-Car Mercedes) 0.5))  
(f-tellm ((Millionaire John) 0.6))

The the first two assertions would cause the fuzzy-rule to fire and result in the inference

((Rich-Person John) 0.5)

However, the last assertion would cause an inconsistency as the truth value of John being a Billionaire (0.6) exceeds the previously inferred truth value of his being a Rich-Person (0.5) though Rich-Person subsumes Billionaire.

#### Figure 5. Example of Inconsistency

---

To deal with inconsistency, we have developed a fuzzy truth maintenance system (FTMS) that processes these different kinds of inconsistencies. This FTMS records the justification of propositions in a list of justification structures associated with each instance. A justification structure specifies (i) a fuzzy proposition and (ii) whether the proposition was asserted by the user, deduced by the terminological model, or inferred by a rule. For example, the justification structure in Figure 5 indicates that the justification that John may be a house-owner with a truth value of 0.5 is that a rule "Rich-People-Are-House-Owners" made such an inference. Whenever a new fuzzy proposition is added to by the system, the FTMS incorporates the truth value of the current proposition with the truth values for the same proposition in the justification list. If there is an inconsistency, then the user is notified, else the modification is completed. If the proposition is a binary predicate, the consistency checking uses the role subsumption lattice. An algorithm for truth maintenance of propositions is outlined in Figure 6.

### 3.2.2 Assertion Processor

The assertion processor translates user asserted statements and fuzzy rule inferences into internal assertional changes and propagates these changes to the deductive reasoner and the approximate reasoner. Asserted propositions have the highest precedence followed by propositions deduced by the deductive reasoner. Propositions inferred by the approximate reasoner have the lowest precedence. The deductive reasoner overrides the plausible inference of the approximate reasoner when a deduction is made, and when the deduced proposition is retracted the plausible conclusion is reactivated.

### 3.3 Extending the Semantic Pattern Matcher for Partial Matching

We have modified CONCRETE, the pattern matcher of CLASP, for plausible rule based inferences. CONCRETE is a semantic pattern matcher which uses a combination of Forgy's Rete pattern matcher and LOOM's deductive pattern matcher [20],[21]. We first outline the fuzzy rule language. Then we describe the deductive pattern matcher of LOOM and semantic pattern matcher (CONCRETE) of CLASP and our extension to semantic pattern matcher for partial match. Finally, we describe the approximate reasoner for plausible inferences.

---

*Module Update-Fuzzy-DB(P,T)*

1. Let the fuzzy proposition, P be  $[\alpha, \mu_i]$ , where  $\alpha$  is the argument of proposition and  $\mu_i$  is the truth value of the proposition and T is the "type" of the fuzzy proposition, i.e., one of asserted by user, retracted by user, inferred by fuzzy rule or deduced by terminological model.
2. If a fuzzy proposition P is asserted by the user then perform *Consistency-Checker* for the asserted truth value  $\mu_i$ .
2. If T is either inferred by a fuzzy rule, or is deduced by the terminological model (e.g., inheritance links), or is retraced by the user, then
  - (a) (1) if a justification structure of the proposition exists then compute the new truth value  $\mu_j$  of the proposition else  
(2) Create a justification structure if it does not exist and assign the value of  $\mu_i$  to  $\mu_j$ .
  - (b) (1) Create a fuzzy proposition  $P_j$  as  $[\alpha, \mu_j]$ .  
(2) Perform *Consistency-Checker(P<sub>j</sub>)* for the resultant truth value.
3. If *Consistency-Checker(P<sub>j</sub>)* returns true then
  - (a) Update the justification structure as follows: If T is retraction by user remove the fuzzy proposition P from it else add the fuzzy proposition {P,T} to the it.
  - (b) Update the proposition in the fuzzy database to  $P_j$ .
  - (c) Return True.

*Module Consistency-Checker(P)*

1. Let the fuzzy proposition, P be  $[\alpha, \mu_j]$ , where  $\alpha$  is the argument of proposition and  $\mu_j$  is the truth value of the proposition.
2. Find all parent fuzzy propositions with the same argument  $\alpha$  in the fuzzy database.
3. Let ConsistencyCheck be the logical conjunction of the values returned by *Parent-Check(P, P<sub>j</sub>)* for each parent fuzzy proposition  $P_j = > [\alpha, \mu_s]$ .
4. Return ConsistencyCheck.

*Module Parent-Check(P, P<sub>j</sub>)*

1. If  $P_j$  subsumes P and  $\mu_s < \mu_j$ , notify the user of inconsistency. Let ReturnValue be False.
2. If P subsumes  $P_j$  and  $\mu_j < \mu_s$ , notify the user of inconsistency. Let ReturnValue be False.
3. If neither of the above, then let ReturnValue be assigned the value returned by *Update-Fuzzy-DB(P<sub>j</sub>, deduced\_by\_terminological\_model)*
4. Return the ReturnValue.

**Figure 6. Algorithm for Truth Maintenance**

---

### 3.3.1 Fuzzy Rule Language

Uncertainty in a rule may be expressed in the consequent side of the rule which is assertional in nature. Example of a fuzzy rule is given in Figure 7.

---

```
(def-fuzzy-rule New-Car-Owners-Are-Rich
  :if (:AND (NEW-CAR ?y)
            (HAS-CAR ?x ?y) )
  :then ((RICH-PERSON ?x) 0.6) )
```

**Figure 7. Example of a Fuzzy Rule**

---

Note that the actual truth value to be recorded for an inferred proposition as a result of the firing of the rule may, however, be different from the truth value of the rule as a consequence of approximate reasoning and truth maintenance.

### 3.3.2 Semantic Pattern Matching in LOOM and CLASP

Terminological knowledge can be viewed as a perspicuous encoding of bidirectional definitional rules. In classification based systems, an instance is matched to a pattern, by the realizer, by first abstracting it and then by classifying the abstraction [11]. A concept P is associated with a pattern P(x); thus matching an individual to a pattern corresponds to recognizing an instantiation relationship between the individual and the corresponding concept.

The deductive pattern matcher in LOOM is an extension to the realizer [11]. The classifier in LOOM's pattern matcher can ask questions about the individual being classified during classification, using backward chaining, and a sufficiently detailed abstraction is built up incrementally. In addition the pattern matcher can also perform a forward inference. Thus it has mixed both forward deduction and backward deduction.

The semantic pattern matcher in CLASP combines Forgy's Rete Pattern Matcher with the deductive matcher of LOOM. The rule compiler builds a concept classification Rete (CONCRETE) net as rules are loaded into the rule base. The LOOM matcher computes assertional changes that can be deduced from the terminological knowledge and it informs the CONCRETE net about relevant changes. To avoid long chains of CONCRETE nodes and early unnecessary joins a data dependency analysis is performed on the patterns [20],[21].

### 3.3.3 Semantics-based Fuzzy Pattern Matching

To deal with uncertainty, a fuzzy pattern matcher needs to handle tokens that express uncertainty. For this it needs to record the degree of match, which is the extent to which an uncertain token matches a condition of a rule, in appropriate nodes in the CONCRETE net. The pattern matcher also needs to combine the partial matches as tokens are propagated down the CONCRETE net. The fuzzy pattern matcher also needs to generate instantiations of fuzzy rules. In addition, as concept nodes of type TRUE do not have their own memory, the pattern matcher needs to query LOOM about partial class memberships.

The pattern matcher of CLASP, CONCRETE, has been modified in three ways.

- (1) The pattern matcher has been extended to query LOOM for partial class memberships.
- (2) The instantiation structure of the CONCRETE has been extended to represent the degree of partial matching.
- (3) The updating mechanism for a node has been modified to calculate or update the matching degree

of instantiations stored in the node's memory.

### 3.3.4 Approximate Reasoner

The approximate reasoner makes plausible inferences based on terminological knowledge, fuzzy propositions and uncertain rules. It also interacts with the FTMS to maintain consistency of the propositions database and to infer truth values to be used in the recording of inferred propositions. The use of justification structures in FTMS also helps in the combination of truth values associated with the same inference in different rules. In addition the approximate reasoner informs the deductive reasoner about only those additions or deletions to the propositions database whose certainty degree is one. Moreover, the deductive reasoner informs the approximate reasoner about all additions or deletions to the propositions database.

#### 3.3.4.1 Uncertainty Calculi

The approximate reasoning model can support different kinds of approximate reasoning. The user may specify the model he wishes to chose. At present two models are supported. Both are based on triangular norms in fuzzy logic [4].

Uncertainty is propagated using T-norm operators in fuzzy logic. T-norms are binary functions that satisfy conjunction while T-conorms are binary functions that satisfy disjunction. Both are 2-place  $[0,1] \times [0,1]$  to  $[0,1]$  functions that are monotonic, commutative and associative and their corresponding boundary conditions satisfy the truth tables of the logical AND and OR operators. A function  $T(a,b)$  aggregates the degree of certainty of two clauses in the same premise. A function  $S(a,b)$  aggregates the degree of certainty of the same conclusions derived from two rules. The associativity property may be used for representation of conjunction of a large number of clauses.

The user may select one of the two following types of T-norm operators:

- (a)  $T1(a,b) = ab$  and  $S1(a,b) = a + b - ab$
- (b)  $T2(a,b) = \min(a,b)$  and  $S2(a,b) = \max(a,b)$

#### 3.3.4.2 Inference Mechanism

The reasoner performs plausible inference in a data driven, forward-chaining manner. Fuzzy rules only specify plausible inferences which in turn update instances. As a result of firing of these fuzzy rules, the truth-value of an instance in a concept or in a role may be added or updated.

A fuzzy rule, after firing once, can be instantiated again if

- (1) When one of the conditions in its pattern is no longer satisfied, or
- (2) An assertion or inference by another rule updates the truth-value of an existing proposition.

## 4. RELATED WORK

Lokendra Shastri has developed a framework, based on the principle of maximum entropy, for dealing with representation of and reasoning with semantic networks [14],[15]. His framework treats statements as evidential assertions, assigning a number to each to represent the evidential import. Given statistical data it can answer questions like "given the state of knowledge of an agent, which choice is most probably correct". While his framework can handle exceptions, multiple inheritance and ambiguities, it has two limitations. First, his approach is based on the availability of statistical data which may not be available. Second, there is no classifier to maintain the consistency of the terminology because the concepts and roles are not of the definitional type.

Heinsohn and Owsnicki have proposed a model of probabilistic reasoning in hybrid term subsumption systems



[9]. Uncertain knowledge is represented as probabilistic implications and probabilistic inheritance is used as a reasoning mechanism. They consider universal knowledge to be related to the extensions of concepts, i.e., the set of real world objects. This empirical or belief knowledge is stored in a Probabilistic Box (PBox). They have extended a term subsumption language by defining the syntax and semantics of probabilistic implication, which quantifies the relative degree of intersection of two extensions. While the range of applicability of hybrid term subsumption systems may be enlarged with this model, it is limited in the kind of uncertain knowledge it can represent. Most rules in expert systems involve complex conditions which may not be completely expressible as concept definitions. Therefore, probabilistic implications need to be extended before these could be used for building expert systems.

Bonissone *et al.* have developed a T-norm based reasoning architecture, RUM, for frame based systems [5]. The premise is that treatment of uncertainty must address representation, inference and control layers in expert systems. The representation uses a certainty frame with set of associated slots. However, the limitation of RUM is that it cannot use terminological knowledge, unlike term subsumption systems.

## 5. SUMMARY

An architecture has been implemented and described for approximate reasoning with terminological systems. The assertional component has been extended for representing and reasoning with uncertain propositions. Using terminological knowledge, fuzzy-rules, T-norm based uncertainty calculi and a fuzzy truth maintenance system, plausible inference can be made. The fuzzy truth maintenance system ensures the consistency of truth values of propositions and the assertion processor translates and propagates internal changes.

*This architecture presents some benefits for developing expert systems. First, expert systems can be built which can refer to terminological knowledge and also reason under uncertainty. Second, it allows for representation and reasoning using uncertainty in the assertional component as well as uncertainty in judgmental knowledge. These two features improve the reasoning capability of expert system. Third, terminological knowledge is applied to both deductive and approximate reasoning, i.e., it is reusable. And fourth, the maintainability and explanation capabilities of expert systems could be improved because meanings of terms are explicitly represented and are separated from heuristic knowledge that is used for plausible inferences.*

## REFERENCES

- [1] G. Abrett and M.H.Burstein, "The KREME Knowledge Editing Environment," *International Journal of Man-Machine Studies*, 27, 103-126, 27, April, 1987.
- [2] R. L. Bates and R. MacGregor, "Knowledge Representation in NIKL," in *AI Tools and Techniques*, editor Mark Richer. Norwood,NJ: Ablex Publishing Corporation, 1989, Chapter 6, pp 183-196.
- [3] D. G. Bobrow and T. Winograd, "An Overview of KRL, a Knowledge Representation Language," in *Readings in Knowledge Representation*, editors R. G. Brachman and H. J. Levesque. Los Altos, CA: Kaufmann Publishers, pp 263-286, 1985.
- [4] P. P. Bonissone, "Summarizing and Propagating Uncertain Information with Triangular Norms," *International Journal of Approximate Reasoning* ,1. Amsterdam, Holland: Elsevier Publishing B.V., pp 71-101, 1987.
- [5] P. P. Bonissone, S. Gans and K. S. Decker, "RUM: A Layered Architecture for Reasoning with Uncertainty," In *Proceedings of the 10th International Joint Conference on Artificial Intelligence, AAAI*, pp 891-898, August 1987.
- [6] R.J.Brachman, "What is-a is and isn't: An Analysis of Taxonomic Links in Semantic Networks," *Computer*, 16(10):30-36, October 1983.

- [7] R.J.Brachman and J.G.Schmolze, "An Overview of the KL-ONE Knowledge Representation System," *Cognitive Science* 9, 171-216, August 1985.
- [8] C. L. Forgy, "Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem," *Artificial Intelligence*, 19(1982)17-37.
- [9] J. Heinsohn and B. Owsnicki-Klewe, "Probabilistic Inheritance and Reasoning in Hybrid Knowledge Representation System," in *Proceedings of the 12th German Workshop on Artificial Intelligence (GWAI-88)*, Berlin:Springer, pp 51-60, 1988.
- [10] N. S. Lee, Y. L. Grize and K. Dehnad, "Quantitative Models for Reasoning under Uncertainty in Knowledge-Based Systems," *International Journal of Intelligent Systems* Vol II, 15-38(1987).
- [11] R. MacGregor. "A Deductive Pattern Matcher," in *Proceedings of AAAI-88*, 1988.
- [12] P. F. Patel-Schneider, B. Owsnicki-Klewe, A.Kobsa, N. Guarino, R. MacGregor, W. S. Mark, D. McGuinness, B. Nebel, A. Schmiedel and J. Yen, "Report on the Workshop on Term Subsumption Languages in Knowledge Representation," *AI Magazine*, 11:16-23, 1990.
- [13] J. G. Schmolze and T. A. Lipkis, "Classification in the KL-ONE Knowledge Representation System," in *Proceedings of the Eighth International Joint Conference on Artificial Intelligence. IJCAI*, 1983.
- [14] L. Shastri, "Default Reasoning in Semantic Networks: A Formalization of Recognition and Inheritance," *Artificial Intelligence*, 39(3):283-355, July, 1989.
- [15] L. Shastri and J. A. Feldman, "Evidential Reasoning in Semantic Networks," in *Proceedings of IJCAI-85*, Los Angeles, California, pp 18-23, August 1985.
- [16] N. Vaidya, "Integrating Approximate Reasoning with Term Subsumption Systems", Masters Thesis, Department of Computer Science, Texas A&M University, College Station, Texas, August 1991.
- [17] W. A. Woods, "What's in a Link: Foundations for Semantic Network," *Representation and Understanding: Studies in Cognitive Science*, D. G.Bobrow and A. M. Collins, editors. Academic Press, pp 35-82, 1975.
- [18] J. Yen, "Generalizing Term Subsumption Languages to Fuzzy Logic", in *Proceedings of the 12th International Joint conference on Artificial Intelligence (IJCAI-91)*, pp 472-477, Sydney, Australia, August 1991.
- [19] J. Yen and Piero P. Bonissone, "Extending Term Subsumption Systems for Uncertainty Management", in *Proceedings of 6th Conference on Uncertainty in Artificial Intelligence*, pp 468-473, July 1990.
- [20] J. Yen, R Neches and R. MacGregor, "Classification-based Programming: A Deep Integration of Rules and Frames," Technical Report ISI/RR-88-213, USC / Information Sciences Institute, Marina del Rey, CA, 1989.
- [21] J. Yen, R. Neeches and R. MacGregor, "Integrating Term Subsumption Systems and Production Systems," *IEEE Transactions on Knowledge and Data Engineering*, Vol3, No.1, pp 25-32, March 1991.
- [22] L.A.Zadeh, "The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems," *Approximate Reasoning in Expert Systems*, M.M.Gupta, A.Kandell, W.Bandler, J.B.Kiszka (editors), (North-Holland): Elsevier Publishers B.V., pp 3-31, 1985.