Generalizing Term Subsumption Languages to Fuzzy Logic

John Yen"
Department of Computer Science
Texas A&M University
College Station, TX 77843
Yen@CSSUNTAMU.EDU

Abstract

During the past decade, knowledge representation research in AI has generated a class of languages called term subsumption languages (TSL), which is a knowledge representation formalism with a well-defined logic-based semantics- Due to its formal semantics, a term subsumption system can automatically infer the subsumption relationships between concepts defined in the system. However, these systems are very limited in handling vague concepts in the knowledge base. In contrast, fuzzy logic directly deals with the notion of vagueness and imprecision using fuzzy predicates, fuzzy quantifiers, linguistic variables, and other constructs. Hence, fuzzy logic offers an appealing foundation for generalizing the semantics of term subsumption languages. Based on a test score semantics in fuzzy logic, this paper first generalizes the semantics of term subsumption languages. Then, we discuss impacts of such a generalization to the reasoning capabilities of term subsumption systems. The generalized knowledge representation framework not only alleviates the difficulty of conventional AI knowledge representation schemes in handling imprecise and vague information, but also extends the application of fuzzy logic to complex intelligent systems that need to perform high-level analyses using conceptual abstractions.

1 Introduction

During the past decade, knowledge representation works in AI have generated a class of languages called term subsumption languages (TSL), which is a knowledge representation formalism with a well-defined logic-based semantics. Using a TSL, a knowledge engineer can explicitly describe defining characteristics of concepts (unary terms) and roles (binary terms) [Patel-Schneider *et al.* 1990][1]. The major strength of term subsumption systems is their reasoning capabilities offered by a *classifier.* The classifier is a special purpose reasoner that automatically infers and maintains a consistent and accurate taxonomic lattice of logical subsumption relations between terms [Schmolze and Lipkis 1983], These formalisms generally descend from the ideas presented in KL-ONE [Brachman and Schmolze 1985]- Term subsumption languages are a generalization of both semantic networks and frames because the languages have well-defined semantics, which is often missing from frames and semantic networks [Woods 1975, Brachman 1983].

Term subsumption languages are limited to expressing crisp concept definitions. However, many useful concepts that are needed by an intelligent system do not have well-defined boundaries, i.e., they are vague concepts. For instance, we may say that a baseball player is a good hitter if the person's hitting ratio is *fairly high.* In an intelligent monitoring and control system, we may wish to define a critical valve as a valve that has a *low tolerance* of pressure. In all these examples, a concept is defined by referring to other vague terms such as "fairly high hitting ratio" and "low pressure tolerance". It is the difficulty to express these vague concepts in term subsumption languages that motivates us to generalize the languages.

Fuzzy logic, which is a generalization of conventional logic, directly deals with the notion of vagueness and imprecision using fuzzy predicates, fuzzy quantifiers, linguistic variables, and other constructs. Thus, it offers an appealing foundation for generalizing TSL to capture imprecise and vague linguistic terms. In particular, the test score semantics in fuzzy logic allows us to easily generalize the term forming expression in TSL into elastic constraints, which can be satisfied to a degree.

In the following sections, we first introduce the basics of term subsumption languages and test score semantics as a background. Section 3 generalizes the semantics of term subsumption languages and describes a complete and sound subsumption test algorithm for a simple fuzzy term subsumption language. A discussion of related works then follows. Finally, we summarize the potential benefits of our approach.

*This research was supported by Engineering Excellence Fund at Texas A&M University.

'A role in a TSL corresponds to a slot in other frame-based systems.

2 Background

2.1 Term Subsumption Languages

A term subsumption languages (TSL) distinguishes two kinds of concepts: primitive concepts and defined concepts. A primitive concept is a concept whose definition can not be stated in the language; a defined concept is a concept whose definition is described using other concepts and a set of concept forming expressions provided by the language. *Value restriction* and *number restriction* are two concept forming constructs that are offered by almost all TSL's. A value restriction restricts the type of a role value (i.e., slot value), while a number restriction constrains the cardinality of a role value. In this paper, a value restriction will be expressed in the form of (:all R C) which means that all values of the role R are of type C. A number restriction can be either in the form of (:at-least n R), which means that the role R has at least n values, or in the form of (:at-most n R), which states that the role R has at most n values.

To describe the formal semantics of these concept forming constructs in TSL's, we need to introduce the following terminology. A terminological knowledge base, denoted by X, consists of concepts and roles (which are also called relations in some TSL's) defined using a term subsumption language. An interpretation I_T of T is a pair *(V, £)* where *V* is a set of individuals described by terms in *T* and € is an *extension function* that maps concepts in T to subsets of D*,* and roles in *T* to subsets of the Cartesian product, D x D , Based on these notations, we describe the semantics of some term forming expressions in TSL's below [Nebel 1988]:

$$\mathcal{E}[(\textbf{:and } C_1 ... C_n)] = \mathcal{E}[C_1] \cap \cdots \cap \mathcal{E}[C_n],$$
$$\mathcal{E}[(\textbf{:all } \mathrm{R\ C})] =$$
$$\{x \in D | \forall y :< x,\ y > \in \mathcal{E}[R] \Rightarrow y \in \mathcal{E}[C]\},$$
$$\mathcal{E}[(\textbf{:at-least } \mathrm{n\ R})] =$$
$$\{x \in \mathcal{D} |\ \| \{y \in \mathcal{D} | < x,\ y > \in \mathcal{E}[R]\} \| \geq n\},$$
$$\mathcal{E}[(\textbf{:at-most } \mathrm{n\ R})] =$$
$$\{x \in \mathcal{D} |\ \| \{y \in \mathcal{D} | < x,\ y > \in \mathcal{E}[R]\} \| \leq n\},$$
$$\mathcal{E}[(\textbf{:range } \mathrm{R\ C})] =$$
$$\{< x,\ y > \in \mathcal{D} \times \mathcal{D} | < x,\ y > \in \mathcal{E}[R] \wedge y \in \mathcal{E}[C]\}.$$

2.2 Test Score Semantics

In 1981, while the first KL-ONE workshop was being held, Lotfi A, Zadeh was advocating a meaning-representation language called PRUF based on possibility theory in fuzzy logic [Zadeh 1978b]. The semantics underlying PRUF is what Zadeh referred to as *test score semantics,* which interprets the meaning of a predicate as having an *elastic constraint* on objects in the database [Zadeh 1981, Zadeh 1982]. In PRUF, a query is processed by first applying a sequence of tests to database objects, yielding a collection of test scores. By aggregating these test scores, the system obtains an overall test score that measures the compatibility between the query and the database.

Test-score semantics is more general than the semantics of term subsumption languages, in which test scores are limited to true and fate- Unlike KL-ONE, however, PRUF was not concerned about developing efficient special purpose reasoners. By combining works in these two areas, we can develop a knowledge representation system that takes advantage of both the generality of test score semantics and the efficient reasoning capabilities of term subsumption systems.

3 Generalizing the Semantics of Term Subsumption Languages

In this section, we generalize the semantics of term subsumption language. First, we generalize the extension of a term and the subsumption relationship between terms. Then we describe how various concept forming expressions can be generalized into elastic constraints using test score semantics in fuzzy logic.

3.1 Generalizing the Extension Function and the Subsumption Relationship

We first generalize the extension function £ such that the extension of a concept is a fuzzy subset of *V,* and the extension of a role is a fuzzy subset of *V* x *V.* A fuzzy subset C of Pis characterized by a membership function u_c that maps elements of Dto the interval [0, 1]. The degree to which an element x of *V* belongs to a concept C is denoted as u_c *(z)-* Similarly, the degree to which an ordered pair < *x,y* > belongs to a role R is denoted as U_R (x,y)- Moreover, we can generalize the extension of a term forming expression to a fuzzy set and denote the degree to which an object x satisfies a term forming expression *e* by $u_e(x)$.

A concept *C* subsumes a concept C_2 if and only if the extension of the former is a fuzzy superset of the extension of the latter. More formally, we say that C_1 subsumes C_2 if and only if for any set *V* and any extension function £ over *V* the following holds:

$$\forall d \in \mathcal{D} : \mu_{C_1}(d) \geq \mu_{C_2}(d) \qquad 0)$$

3.2 Soft Value Restriction

A value restriction in a terminological language constrains all the role values of an object to be instances of a given class. For instance, a type of valve can be defined by restricting its pressure tolerance to an interval. We can generalize this kind of constraint to an "elastic constraint" or "soft constraint" (e.g., (:all Pressure-Tolerance Low-pressure)) in two ways. The logic implication in the original semantics

$$\forall y\ \mathtt{Pressure\text{-}Tolerance}(x, y) \Rightarrow \mathtt{Low\text{-}pressure}(y) \quad (2)$$

can be generalized to a fuzzy implication operator. Thus, the degree to which a value restriction is satisfied by an instance x is determined by the degree to which the implication is true for x. This can be formulated as follows:

$$\mu_{(:\mathrm{all\ R\ C})}(x) \;=\; \inf_{y_i}\left[\mu_{\mathrm{R}(x,y) \Rightarrow \mathrm{C}(y)}(x, y_i)\right] \quad (3)$$

where $U_R(x_p y)$ =>c(y) (*»#) can be defined using € various fuzzy implication operators [Magrez and Smets 1989].

An alternative approach to generalizing the semantics of a value restriction is to use the notion of conditional necessity in possibility theory [Zadeh 1978a, Dubois and Prade 1988]:

$$\mu_{(\text{:all R C})}(x) = Nec(\text{C}(y)|\text{R}(x,y)) \quad (4)$$
$$= 1 - Poss(\neg\text{C}(y)|\text{R}(x,y)) \quad (5)$$
$$= 1 - \frac{\max_y \{\min[1-\mu_{\text{C}}(y), \mu_{\text{R}}(x,y)]\}}{\max_y \mu_{\text{R}}(x,y)} \quad (6)$$

In essence, this formula computes a measure that a Pressure-Tolerance of x is *necessarily* a college graduate. It is easy to verify that both generalizations of the value restriction above are consistent with the original semantics. For the rest of our discussion, we will be using Equation 6 as the generalized semantics of value restrictions.

3.3 Soft Number Restriction

The cardinality of a fuzzy set is defined using sigma-counts in test-score semantics[Zadeh 1981]:

$$\Sigma COUNT(A) = \sum_{i=1}^{n} \mu_A(x_i) \quad (7)$$

where A is a fuzzy set characterized by a membership function μ_A. We can thus generalize the number restriction in terminological languages to a "soft" number restriction using sigma-counts and fuzzy numbers:

$$\mu_{(\text{:at-least n R2})}(x) = \mu_{at-least-n}\left(\sum_y \mu_{R2}(x,y)\right) \quad (8)$$

$$\mu_{(\text{:at-most n R2})}(x) = \mu_{at-most-n}\left(\sum_y \mu_{R2}(x,y)\right) \quad (9)$$

where at-least-n and at-most-n are fuzzy subsets of real numbers characterized by the following membership functions:

$$\mu_{at-least-n}(z) = \begin{cases} 0 & z \leq n-1 \\ z-n+1 & n-1 \leq z \leq n \\ 1 & z \geq n \end{cases}$$

$$\mu_{at-most-n}(z) = \begin{cases} 1 & z \leq n \\ n+1-z & n \leq z \leq n+1 \\ 0 & z \geq n+1 \end{cases}$$

where z is a real number.

3.4 Fuzzy Conjunction

Finally, the degree an instance satisfies a conjunction of sub-expressions can be computed using the "min" operator in fuzzy set theory. For instance, suppose **Critical-Valve** is defined as a valve whose pressure tolerance is low. This can be expressed as

```
(defconcept Critical-Valve (:and Valve
  (:all Pressure-Tolerance Low-pressure)))
```

The degree to which an instance is a `Critical-Valve` can thus be defined as follows:

$$\mu_{\text{Critical-Valve}}(x) = \min\{\mu_{\text{Valve}}(x), \mu_{(\text{:all PT L})}(x)\} \quad (10)$$

where PT and L stand for "Pressure-Tolerance" and "Low-pressure" respectively.

It should be noted, however, that other Triangular Norms operators could be used to represent the conjunction of the sub-expressions. Moreover, by considering the bounds imposed by all Triangular Norms, we could represent the lower and upper bounds of such intersection as:

$$\mu_{\text{Critical-Valve}}(x) = [\max\{0, \mu_{\text{Valve}}(x) + \mu_{(\text{:all PT L})}(x) - 1\}, \min\{\mu_{\text{Valve}}(x), \mu_{(\text{:all PT L})}(x)\}] \quad (11)$$

3.5 Defining Fuzzy Concepts Using Membership Functions

A fuzzy concept can also be defined by describing its membership function explicitly, or by modifying the membership function of an existing fuzzy concept. To do the former, we also need to specify the domain of the membership function, which is called *the universe of discourse* in fuzzy set theory. For instance, we may define the fuzzy concept `Low-pressure` by specifying its membership function and its universe of discourse as follows:

```
(defconcept Low-pressure
  :universe-of-discourse Air-pressure
  :membership-fx (lambda (p) (low p)))
```

where low is a function that returns the membership degree for a given pressure. Once such a fuzzy set is completely specified, we can define many other fuzzy sets using modifiers (also called hedges in fuzzy logic) such as NOT, VERY, SLIGHTLY, etc. This can be expressed in a generalized term subsumption language as follows:

```
(defconcept Very-Low-pressure
    (:VERY Low-pressure) ).
```

4 Subsumption Test

The major component of a term subsumption system's reasoner is a subsumption test, which determines whether a term description subsumes (i.e., is more general than) another term description. In our generalized term subsumption systems, a term a subsumes another term b if and only if the extension of a is a fuzzy superset of the extension of b (i.e., $\forall x \in \mathcal{D}\ \mu_a(x) \geq \mu_b(x)$).

In a landmark paper that discussed the tradeoff between the expressiveness and the tractability of subsumption test, Ronald J. Brachman and Hector J. Levesque described a simple term subsumption language $\mathcal{FL}^-$ that has a sound and complete algorithm for the subsumption test [Brachman and Levesque 1984][2]. In this section, we show that a similar fuzzy term subsumption language, called $\mathcal{FTSL}^-$, also has a sound and complete algorithm for the subsumption test. The grammar of a $\mathcal{FTSL}^-$ is shown below.

[2] Even though Brathman and Levesque's algorithm for computing the subeumption of concept descriptions in Fl⁻ has polynomial time complexity, Bernhard Ncbel has recently shown that the problem of determining the subsumption of terms, in general, is intractable[Nebel 1989].

<term-definition> ::=
<primitive-concept-definition>
| <defined-concept-definition>
| <primitive-role-definition>
<primitive-concept-definition> ::=
(defconcept <c-name>
(:and <c-name>* :primitive))
<defined-concept-definition> ::=
(defconcept <c-name>
(:and <concept-forming-expr>+))
| (defconcept <c-name>
:universe-of-discourse <c-name>
:membership-fx <lambda expression>)
| (defconcept <c-name>
(<modifier> <c-name>))
<modifier> ::= :NOT | :VERY | :SLIGHTLY
<concept-forming-expr> ::= <c-name>
| (:all <r-name> <c-name>)
| (:some <r-name>)
<primitive-role-definition> ::=
(defrole <r-name> :primitive)

The subsumption algorithm for $\mathcal{FTSL}^-$ is a slightly modified version of that of $\mathcal{FL}^-$ presented in [Brachman and Levesque 1984]:

Subsumption Algorithm for $\mathcal{FTSL}^-$: SUBS?[a,b]

1. If a and b are both defined by membership functions, then return true if they have the same universe of discourse and their membership functions satisfy the condition[3]:

$$\forall x \in U \mu_a(x) \geq \mu_b(x)$$

where U is the universe of discourse of these concepts. Otherwise, return false.

2. If only one of the two concepts are defined using membership functions, return false.
3. If both a and b are defined using concept forming expressions, normalize their descriptions by recursively replacing all non-primitive concepts in the descriptions by their definitions.
4. Flatten the normalized concept description by removing all nested :and operators.
5. Collect ail arguments to an :all for a given role.
6. Assuming the description of a is now (:and a_1 ... a_n) and the description of b is now (:and b_1 ... b_m), then return true iff for each a_i
   (a) if a_i is an atom (i.e., the name of a primitive concept) or a :some, then one of the bj is a,*.
   (b) if a_i is (:all r c_1), then one of the b_j is (;all r c_2) where SUBS?[c_1,c_2].

By slightly modifying Brachman and Levesque's proof about the soundness and completeness of FL^- 'S subsumption algorithm in [Brachman and Levesque 1984], we can show that the algorithm above is both complete and sound. To prove the soundness of the algorithm, we must show that if SUBS?[a,b] is true, then a indeed subsumes b. Suppose SUBS?[a,b] is true, a and b must be in one of cases below:

1. Both a and b have the same universe of discourse and their membership functions satisfy the test that a is a fuzzy superset of b.
2. Both a and b are defined using concept forming expressions. For any conjunct in a, say a_i, either a_i is among the b_j or it is of the form (:all r c_1). In the latter case, there is an (:all r c_2) among the b_j where SUBS?[c_1,c_2]. To prove by induction, we need to show that if $\mathcal{E}[c_1] \supseteq \mathcal{E}[c_2]$, then

$$\mathcal{E}[(:\texttt{all}\ r\ c_1)] \supseteq \mathcal{E}[(:\texttt{all}\ r\ c_2)]. \qquad (12)$$

Since $\mathcal{E}[c_1] \supseteq \mathcal{E}[c_2]$, we have

$$\forall y\ \mu_{c_1}(y) \geq \mu_{c_2}(y).$$

Using the generalized semantics for :all (i.e., Equation 6), we get

$$\mu_{(:\texttt{all}\ r\ c_1)}(x) \geq \mu_{(:\texttt{all}\ r\ c_2)}(x).$$

Equation 12 thus follows. So no matter what a_i is, the extension of b (which is the conjunction of all the b_j's) must be a subset of a_i. Since this is true for every a_i, the extension of b must also be a subset of the extension of a.

So, whenever SUBS?[a,b] is true, a subsumes b.

To prove the completeness of the algorithm, we need to show that anytime SUBS?[a,b] is false, there is an extension function $\mathcal{E}$ such that $\mathcal{E}[a] \not\supseteq \mathcal{E}[b]$. There are five cases that may cause SUBS?[a,b] to return false.

1. Assume that a, b are both defined by membership functions, but they have different universe of discourse or there exists an x_i such that $\mu_a(x_i) < \mu_b(x_i)$. In either case, the extension of a is not a superset of the extension of b.
2. Assume that a is defined by concept expressions, and b is defined by a membership function. Let * be an object not in b's universe of discourse, we can construct an extension function $\mathcal{E}$ that assigns * to all concepts defined by concept expressions, and assigns <*,*> to all roles. Hence, * is in the extension of a, but not that of b.
3. Assume that some atom a_i does not appear among the b_j. Let $\mathcal{E}$ assign the ordered pairs <0,1>,<1,1> to every role and 0,1 to every primitive concept except a_i, to which it assigns 1. Hence, 0 is in the extension of b, but not that of a.
4. Assume that a_i is (:**some** r), which does not appear among the b_j. Let $\mathcal{E}$ assign 0,1 to every primitive concepts and <0,1>,<1,1> to every role except r, to which it assigns only <1,1>. Hence, 0 is in the extension of b, but not that of a.
5. Assume that a_i is (:**all** r c_1), where if (:**all** r c_2) appears among the b_j, then, by induction, c_1 does not subsume c_2. Let $\mathcal{E}^*$ be an extension function not using 0 or 1 but such that some object

[3] We assume that before testing the subsumption of a concept that is defined using modifiers, the system has computed its membership function using the standard interpretation of those modifiers in fuzzy logic-

$*$ has a higher membership degree in c_2 than in c_1 (i.e., $\mu_{c_1}(*) < \mu_{c_2}(*)$). Then, let $\mathcal{E}$ contain $\mathcal{E}^*$ and assign 0,1 to every primitive concepts and <0,1>,<1,1> to every role except r, to which it assigns <1,1>,<0,*>. Based on the generalized semantics of `:all` construct, we have

$$\mu_a[0] = \mu_{a_1}[0] = 1 - (1 - \mu_{c_1}(*)) = \mu_{c_1}(*) \quad (13)$$

$$\mu_b[0] = \mu_{(:\mathtt{all}\ r\ c_2)}[0] = \mu_{c_2}(*) \quad (14)$$

$$(15)$$

It then follows that the membership degree of 0 in a is less than that in b.

In all cases, we have shown that £[a] is not a fuzzy superset of £[b]. So, a does not subsume b when SUBS?[a,b] is false. Therefore, we have proved that the subsumption algorithm is sound and complete.

5 Related Work

Most existing works in extending frame-based knowledge representation languages for uncertainty management lie in the category of probabilistic extensions. Lokendra Shastri has developed a framework, based on the principle of maximum entropy, for dealing with uncertainty in semantic networks [Shastri and Feldman 1985, Shastri 1989]. His approach is based on the assumption that the system has certain statistical data (e.g., the number of red apples, the number of sweet apples, ...). Based on these statistical data, Shastri's evidential theory answers questions of the following kind: *Given that an instance, x, is red and sweet, is x more likely to be an apple or a grape 9* The major shortcoming of Shastri's theory is the difficulty in obtaining marginal probability judgements that are required by his model.

Heinsohn and Owsnicki-Klewe recently proposed a model of probabilistic reasoning in hybrid term subsumption systems [Heinsohn and Owsnicki-Klewe 1988]. Probabilistic knowledge is represented as *probabilistic implications* in the form of $C_1 \Rightarrow_s C_2$ where s denotes the conditional probability $P(C_2(x) \backslash C_1(x))$, C_1 and C_2 are concepts defined in the terminological knowledge base. The reasoning mechanism of their model is *proba bilistic inheritance* (i.e., the inheritance of probabilistic implications in concept taxonomy). The issue of nonmonotonicity of probabilistic inheritance has also been discussed in [Grosof 1986].

Even though these probabilistic extensions to frame-based reasoning could potentially enlarge the applicability of term subsumption systems, they do not directly address the issue of representing and reasoning about the subsumption relationships between vague concepts,

6 Summary

We have described an approach for generalizing term subsumption languages to fuzzy logic. Using test score semantics, we have generalized the concept forming constructs in term subsumption languages into elastic constraints. By slightly modifying previous works in term subsumption languages, we are able to show a complete and sound subsumption algorithm for a simple fuzzy term subsumption language. The generalized knowledge representation framework not only alleviates the difficulty of conventional AI knowledge representation schemes in handling imprecise and vague information in an intelligent system, but also enables an intelligent system to construct an abstraction hierarchy automatically based on the semantics of elastic concept descriptions, some of which may be vague and imprecise. Hence, our approach facilitates the development of complex intelligent systems where the system's capability in performing high-level analysis using conceptual abstraction and analyzing vague and imprecise information are both essential.

Acknowledgements

I am grateful to many discussions with Loth* A. Zadeh, Robert MacGregor, Piero Bonissone, and Jonathan Lee. I would also like to thank the referees for their comments on an earlier draft of the paper.

References

[Brachman and Levesque, 1984] Ronald J. Brachman and Hector J. Levesque. The tract ability of subsumption in frame-based description languages. In Proceedings *of AAAI-84,* pages 34-37, Austin, Texas, August 1984.

[Brachman and Schmolze, 1985] R.J, Brachman and J.G. Schmolze, An overview of the KL-ONE knowledge representation system. *Cognitive Science,* 9(2):171-216, August 1985.

[Brachman, 1983] 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,

[Dubois and Prade, 1988] D. Dubois and H. Prade. *Possibility Theory.* Plenum Press, New York, N.Y., 1988.

[Grosof, 1986] B. N. Grosof. Non-monotonicity in probabilistic reasoning. In *Proceedings of the AAAI Workshop on Uncertainty and Probability in Artificial Intelligence,* pages 91-98, Philadelphia, PA, August 1986. AAAI and RCA.

[Heinsohn and Owsnicki-Klewe, 1988] Jochen Heinsohn and Bernd Owsnicki-Klewe. Probabilistic inheritance and reasoning in hybrid knowledge representation systems. In *Proceedings of the 12th German Workshop on Artificial Intelligence (GWAI-88),* pages 51-60, Springer, Berlin, 1988.

[Magrez and Smets, 1989] P. Magrez and P. Smets. Fuzzy modus ponens: A new model suitable for applications in knowledge-based systems. *International Journal of Intelligent Systems,* 4:181-200, 1989.

[Nebel, 1988] Bernhard Nebel. Computational complexity of terminological reasoning in back. *Artificial Intelligence,* 34(3), April 1988,

[Nebel, 1989] Bernhard Nebel. Terminological reasoning is inherently intractable. Technical Report IWBS Report 82, IWBS, IBM Deutschland, W. Germany, October 1989.

[Patel-Schneider *et al.,* 1990] Peter F. Patel-Schneider, Bernd Owsnicki-Klewe, Alfred Kobsa, Nicola Guarino, Robert MacGregor, William S. Mark, Deborah McGuinness, Bernhard Nebel, Albrecht Schmiedel, and John Yen. Term subsumption languages in knowledge representation. *AI Magazine,* ll(2):16-23, 1990.

[Schmolze and Lipkis, 1983] James Schmolze and Thomas Lipkis. Classification in the kl-one knowledge representation system. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence.* IJCAI, 1983.

[Shastri and Feidman, 1985] Lokendra Shastri and Jerome A. Feldman. Evidential reasoning in semantic networks. In *Proceedings of IJCAI-85,* pages 18-23, Los Angeles, California, August 1985.

[Shastri, 1989] Lokendra Shastri. Default reasoning in semantic networks: A formalization of recognition and inheritance. *Artificial Intelligence,* 39(3):283-355, July 1989.

[Woods, 1975] William A. Woods. Whats's in a link: Foundations for semantic networks. In Daniel Bobrow and Allan Collins, editors, *Representation and Understanding: Studies in Cognitive Science.* Academic Press, 1975.

[Zadeh, 1978a] L. A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems,* 1:3-28, 1978.

[Zadeh, 1978b] L. A. Zadeh. PRUF - a meaning representation language for natural languages. *Int. J. Man-Machine Studies,* 10:395-460, 1978.

[Zadeh, 1981] L. A. Zadeh. Test-score semantics for natural languages and meaning representation via pruf. In Burghard B. Rieger, editor, *Empirical Semantics,* pages 281-347. Studienverlag Dr. N. Brockmeyer, Bochum, 1981.

[Zadeh, 1982] L. A. Zadeh. Test-score semantics for natural languages. In *Proceedings of the Ninth International Conference on Computational Linguistics,* pages 425-430, Prague, 1982.