A Four-Valued Semantics for Frame-Based Description Languages

Peter F. Patel-Schneider

Schlumberger Palo Alto Research
3340 Hillview Avenue
Palo Alto, CA 94304

ABSTRACT

One severe problem in frame-based description languages is that computing subsumption is computationally intractable for languages of reasonable expressive power. Several partial solutions to this problem are used in knowledge representation systems that incorporate such languages, but none of these solutions are satisfactory if the system is to be of general use in representing knowledge. A new solution to this problem is to use a weaker, four-valued semantics for frame-based description languages, thus legitimizing a smaller set of subsumption relationships. In this way a computationally tractable but expressively powerful knowledge representation system incorporating a frame-based description language can be built.

I Introduction

There is a trade-off between expressive power and computational tractability in knowledge representation formalisms [Levesque and Brachman, 1985]. If the formalism is expressively powerful, such as standard first-order logic, then reasoning in the formalism is time-consuming, perhaps even undecidable. This may make the formalism unsuitable as the basis of a knowledge representation system. Formalisms that are computationally tractable, such as standard databases, are much less expressive. Even many expressively limited formalisms are computationally intractable, as is standard propositional logic, which has NP-complete reasoning.

This trade-off is present in frame-based description languages [Brachman and Levesque, 1984]. These languages formalize the notion of frames, a notion present in many current knowledge representation systems, as structured types, often called *concepts.* The languages include a set of syntactic operations that are used to form concepts, and other, related, notions such as *slots.* They also include a formal model-theoretic semantics for these syntactic expressions. Thus frame-based description languages are a sort of logic, one which can be used to represent a useful kind of knowledge.

The concept-forming operators vary between different frame-based description languages but generally allow the creation of a concept as the conjunction of a set of more general concepts and a set of restrictions on the attributes of instances of the concept. Such concepts can be loosely rendered as noun phrases such as

> a **student** and a **female** whose **department** is **computer science**, and who has at least **3 enrolled-courses**, each of which is a **graduate-course** whose **department** is an **engineering-department**.

Frame-based description languages are part of KL-ONE [Brachman and Schmolze, 1985], NIKL [Schmolze, 1985], KRYPTON [Brachman et al., 1983, Brachman et al., 1985], and KANDOR [Patel-Schneider, 1984].

The most important operation in frame-based description languages is determining if one concept *subsumes* another. Informally, one concept subsumes another if all instances of the second must be instances of the first, that is, if the first is more general than the second. For example, the concept

> **person** each of whose **male friends** is a **doctor**

subsumes the concept

> **person** each of whose **friends** is a **doctor** who has some **speciality**,

in standard frame-based description languages. This is so because, in the standard semantics for frame-based description languages all instances of the second concept must also be instances of the first.

Unfortunately, subsumption is a complicated relationship and can be difficult to compute. This problem first came to light during the formalization of part of KL-ONE where it was discovered that the subsumption algorithm in KL-ONE was incomplete [Schmolze and Israel, 1983]. The complexity of computing subsumption in KL-ONE and NIKL, which has a similar frame-based description language, is still unknown and the problem may even be undecidable. More recently, Brachman and Levesque [Brachman and Levesque, 1984] showed that computing subsumption in a very simple frame-based description language was NP-hard, indicating that computing subsumption in more expressive frame-based description languages is very difficult, at least in the worst case. Since computing subsumption is the most important operation in frame-based description languages and will be performed often[1], this is a serious problem in these languages.

There are several ways to partially solve this problem. The first partial solution is to simply ignore the problem. The examples used by Brachman and Levesque to show that computing subsumption in their frame-based description language is NP-hard are not likely to occur in actual knowledge bases. Perhaps computing subsumption will be reasonably fast in actual knowledge bases. This sort of solution occasionally works well but will fail for more expressive frame-based description languages, such as NIKL's, which have no known total algorithm for computing subsumption, and whose best known solution is to translate the problem into a theorem-proving problem.

The second partial solution is to limit the expressive power of the frame-based description language. The problem with this solution is that the expressive power must be very severely limited

[1] In the current design of KRYPTON, new concepts are created as part of resolution steps, leading to a great number of new subsumption questions being asked during deductions.

to achieve computational tractability, as discovered by Brachman and Levesque. Nevertheless, this was the solution used in the version of KRYPTON actually implemented [Brachman et al., 1985], which has a very limited frame-based description language, in which subsumption in easy to compute.

A combination of these two solutions was used in KANDOR. In this system the frame-based description language is more powerful than that of KRYPTON, but still limited expressively. Computing subsumption in KANDOR is co-NP-complete in the size of numbers appearing in concepts but is otherwise tractable. Moreover, this computation is quite fast in normal circumstances. Also, KANDOR, like many other similar systems, keeps track of subsumption relationships in a concept taxonomy so that each subsumption question need only be asked once. In this way, the worst case behavior of the subsumption problem in KANDOR is rendered less harmful.[2]

A third solution is to provide only a partial subsumption algorithm, one which does not discover all subsumption relationships, only an easy-to-calculate subset of them. This is the solution used in KL-ONE and NIKL, where only simple subsumption relationships are discovered. In this solution the algorithm for computing subsumption is no longer fully defined by the semantics of the frame-based description language. There is little basis for deciding exactly which subsumption relationships to discover, except the reasons of expediency and tractability. The danger is that the discovered subsumption relationships will be simply an *ad hoc* set, with little relationship to the semantics of the frame-based description language, and can be neither characterized nor used effectively.

Given that none of these solutions is satisfactory for frame-based description languages, where reasonable expressive power implies very difficult subsumption and where the semantics must be followed because the system is supposed to be representational, is there a solution to the problem? Unfortunately there is no solution if the standard semantics for frame-based description languages is to be strictly followed. However, there is a way to legitimize the third solution, by using a weaker semantics for frame-based description languages, one that supports fewer subsumption relationships and which has tractable subsumption.[3] This fourth solution is the one that will be explored in this paper.

Weaker semantics have also been proposed for assertional knowledge representation systems. (An assertional knowledge representation system is concerned with assertions or facts instead of frames.) Levesque [Levesque, 1984] suggested that propositional tautological entailment, a weak version of propositional relevance logic, could be used as the basis of a simple knowledge representation system. The advantage of using propositional tautological entailment is that computing inference in it is computationally tractable if formulae are kept in conjunctive normal form. This work was later extended [Patel-Schneider, 1985] to produce a decidable variant of first-order tautological entailment that could be used as the basis of a knowledge representation system.[4] Both of these efforts were based on a four-valued model-theoretic semantics where propositions can be assigned not only true or false, but also neither true nor false, and also both true and false. The semantics used in this paper are very similar to these four-valued semantics.

Of course, there are problems with using different semantics. The standard two-valued Tarskian semantics for logic, which serves as the basis for the standard semantics of frame-based description languages, has been in existence for quite some time now. It is generally agreed that this semantics captures our intuitions of how the world actually is and that the inferences sanctioned by it are a reasonable set of inferences. Any other semantics is liable to be less intuitive than this standard semantics and perhaps may be so counter-intuitive that it is useless for knowledge representation purposes. The goal of this endeavor is to produce a semantics for frame-based description languages that is still intuitive but which also has tractable subsumption.

II A Frame-Based Description Language

The benefits and problems of using four-valued semantics for this purpose will be illustrated using a particular frame-based description language. The language given here is considerably more general than the language $\mathcal{FL}$ presented in [Brachman and Levesque, 1984], for which subsumption in the standard semantics is intractable. This language, called $\mathcal{FL}^+$, is meant to be similar to the frame-based description languages in KL-ONE and NIKL and the initial specifications of KRYPTON [Brachman et al., 1983], except for the lack of number restrictions.[5]

$\mathcal{FL}^+$ has two major syntactic types—*concepts* and *roles*, corresponding to the frames and slots of most frame-based knowledge representation systems. As in these other systems, concepts represent descriptions of related individuals and roles describe relations between these individuals. The intuitive meanings of the various constructs in the language are simple and are based on the intuitive meanings of the basic constructs in typical frame-based knowledge representation systems. Constructs in this language which have counterparts in $\mathcal{FL}$, NIKL, or KRYPTON have analogous intuitive meanings.

The grammar of $\mathcal{FL}^+$ is as follows: (A linear syntax is used here for purposes of clarity and brevity. The keywords in this grammar are derived from the other frame-based description languages.)

```
<concept> ::= <atom> |
              (and <concept>+) |
              (some <role>) |
              (all <role> <concept>) |
              (rvm <role> <role>) |
              (sd <concept> <binding>+)
<binding> ::= (⊆ <role> <role>) |
              (⊇ <role> <role>)
<role> ::=    <atom> |
              (and-role <role>+) |
              (restr <role> <concept>)
```

Here atoms are the names of primitive concepts or roles. The **and** construct for concepts and the **and-role** construct for roles allow the creation of conjunctions. For example, **(and adult male person)** would represent the concept of something that was an adult, a male, and a person, *i.e.*, a man.

The **some** construct guarantees that there will be at least one filler of the role which is its argument. The **all** construct restricts the fillers of a role to belong to a certain concept and the **rvm** (for "role-value-map") construct similarly restricts the fillers of a role to stand in some other relationship to the individual. The

[2] However, ARGON [Patel-Schneider et al., 1984], a query language using KANDOR to represent its knowledge, builds concepts for each query, thus leading to a large number of subsumption questions being asked. Therefore, the performance of subsumption in KANDOR is of vital importance to ARGON.

[3] It is important that the new semantics be weaker than the standard semantics so that all reasoning in it is sound with respect to the standard semantics.

[4] These developments are along the general line of Frisch who has argued that any Artificial Intelligence program, knowledge representation systems in particular, should be the complete implementation of some formalism with model-theoretic semantics [Frisch, 1985].

[5] The reason for not including number restrictions will be seen later.

restr (for "value restriction") construct allows for the creation of roles constrained by the types of their fillers. In this way the concept of

> a **person** with at least one **child**, each of whose **sons** is a **lawyer**, and each of whose **siblings** is also a **friend**

is rendered as

> **(and person (some child)**
> **(all (restr child male) lawyer)**
> **(rvm sibling friend)).**

The **sd** (for "structural description") construct permits the tying together of various fillers by means of some other object, a feature borrowed from KL-ONE. This construct allows the concept of a **project-broadcast message** or

> a **message** for which some **project** exists such that each **sender** of the message is a **project-member** of the project, and each **project-member** of the project is a **recipient** of the message

to be rendered as

> **(and message**
> **(sd project (⊆ sender project-member)**
> **(⊇ recipient project-member))).**

Here the **senders** and the **recipients** are tied together by means of some **project**, which has a certain relationship to both the **senders** and the **recipients**.

III Formal Semantics

The above discussion defines the syntax of $\mathcal{FL}^+$ and indicates intuitively what the constructs are supposed to model. However there is no formal definition of the exact meaning of each construct. This meaning is defined in terms of the following extensional semantics.

The basic ideas behind the semantics are similar to the ideas behind other denotational semantics. There is a set of possible worlds or models and a mapping which maps syntactic objects into semantic entities in each of these possible worlds. However, since the semantics is four-valued, the mapping is more complicated. This is because it is not sufficient to simply state conditions specifying where something is true and rely on the fact that where something is not true, it must be false. Instead, separate conditions for truth and falsity must be given, as in three-valued logics.

In a particular world, each concept is mapped into two sets of individuals. The first set is the set of individuals that belong to the concept, called its *positive extension*. The second set is the set of individuals that definitely do not belong to the concept, called its *negative extension*. Unlike in two-valued semantics, these two sets need not be complements of each other. There may be individuals that are members of neither of these sets, and also individuals that are members of both of these sets.

Individuals which are in both the positive and negative extension of a concept are hard to characterize using the above description. A description of the semantics in terms of knowing better characterizes such individuals. Under this reading, the first set is the set of individuals known to belong to the concept and the second set is the set of individuals known not to belong to the concept. Individuals that are members of neither set then are not known to belong to the concept and are not known not to belong to the concept. This is a perfectly reasonable state for a system that is not a perfect reasoner. Individuals that are members of both sets are, inconsistently, both known to belong to the concept and known not to belong to the concept. This is a slightly harder state to rationalize but can be considered a possibility in the light of inconsistent information.

Similarly, roles are mapped into two sets of ordered pairs of individuals.

There are restrictions on this mapping, corresponding to the intuitive meaning of each of the syntactic constructs of the language. For example, the positive extension of **(and $\mathbf{c}_1$ $\mathbf{c}_2$)** must be the intersection of the positive extension of $\mathbf{c}_1$ and $\mathbf{c}_2$ and its negative extension must be the union of their negative extensions. In this way the intuitive notion of conjunction is made formal.

Now subsumption is defined as one concept subsumes another if the positive extension of the first is always a superset of the positive extension of the second and the negative extension of the first is always a subset of the negative extension of the second. This is the obvious way of defining subsumption in a four-valued semantics.

The semantics is strictly defined in terms of situations. A *situation* is a triple, $\langle \mathcal{D}, \mathcal{E}^t, \mathcal{E}^f \rangle$. $\mathcal{D}$ is a set of individuals. $\mathcal{E}^t$ is a function from concepts to subsets of $\mathcal{D}$ and from roles to subsets of $\mathcal{D} \times \mathcal{D}$, mapping concepts and roles into their positive extensions. $\mathcal{E}^f$ is a function from concepts to subsets of $\mathcal{D}$ and from roles to subsets of $\mathcal{D} \times \mathcal{D}$, mapping concepts and roles into their negative extensions. $\mathcal{E}^t$ and $\mathcal{E}^f$ also map bindings into subsets of $\mathcal{D} \times \mathcal{D}$.

$\mathcal{E}^t$ and $\mathcal{E}^f$ must satisfy the following constraints:

$d \in \mathcal{E}^t[(\mathbf{and}\ \mathbf{c}_1 \ldots \mathbf{c}_n)]$ iff for each i, $d \in \mathcal{E}^t[\mathbf{c}_i]$
$d \in \mathcal{E}^f[(\mathbf{and}\ \mathbf{c}_1 \ldots \mathbf{c}_n)]$ iff for some i, $d \in \mathcal{E}^f[\mathbf{c}_i]$
$d \in \mathcal{E}^t[(\mathbf{some}\ \mathbf{r})]$ iff $\exists e\ \langle d,e\rangle \in \mathcal{E}^t[\mathbf{r}]$
$d \in \mathcal{E}^f[(\mathbf{some}\ \mathbf{r})]$ iff $\forall e\ \langle d,e\rangle \in \mathcal{E}^f[\mathbf{r}]$
$d \in \mathcal{E}^t[(\mathbf{all}\ \mathbf{r}\ \mathbf{c})]$ iff $\forall e\ \langle d,e\rangle \in \mathcal{E}^f[\mathbf{r}]$ or $e \in \mathcal{E}^t[\mathbf{c}]$
$d \in \mathcal{E}^f[(\mathbf{all}\ \mathbf{r}\ \mathbf{c})]$ iff $\exists e\ \langle d,e\rangle \in \mathcal{E}^t[\mathbf{r}]$ and $d \in \mathcal{E}^f[\mathbf{c}]$
$d \in \mathcal{E}^t[(\mathbf{rvm}\ \mathbf{r}\ \mathbf{s})]$ iff $\forall e\ \langle d,e\rangle \in \mathcal{E}^f[\mathbf{r}]$ or $\langle d,e\rangle \in \mathcal{E}^t[\mathbf{s}]$
$d \in \mathcal{E}^f[(\mathbf{rvm}\ \mathbf{r}\ \mathbf{s})]$ iff $\exists e\ \langle d,e\rangle \in \mathcal{E}^t[\mathbf{r}]$ and $\langle d,e\rangle \in \mathcal{E}^f[\mathbf{s}]$
$d \in \mathcal{E}^t[(\mathbf{sd}\ \mathbf{c}\ \mathbf{b}_1 \ldots \mathbf{b}_n)]$ iff $\exists e\ e \in \mathcal{E}^t[\mathbf{c}]$ and, for each i, $\langle d,e\rangle \in \mathcal{E}^t[\mathbf{b}_i]$
$d \in \mathcal{E}^f[(\mathbf{sd}\ \mathbf{c}\ \mathbf{b}_1 \ldots \mathbf{b}_n)]$ iff $\forall e\ e \in \mathcal{E}^f[\mathbf{c}]$ or, for some i, $\langle d,e\rangle \in \mathcal{E}^f[\mathbf{b}_i]$

$\langle d,e\rangle \in \mathcal{E}^t[(\subseteq\ \mathbf{r}\ \mathbf{s})]$ iff $\forall x\ \langle d,x\rangle \in \mathcal{E}^f[\mathbf{r}]$ or $\langle e,x\rangle \in \mathcal{E}^t[\mathbf{s}]$
$\langle d,e\rangle \in \mathcal{E}^f[(\subseteq\ \mathbf{r}\ \mathbf{s})]$ iff $\exists x\ \langle d,x\rangle \in \mathcal{E}^t[\mathbf{r}]$ and $\langle e,x\rangle \in \mathcal{E}^f[\mathbf{s}]$
$\langle d,e\rangle \in \mathcal{E}^t[(\supseteq\ \mathbf{r}\ \mathbf{s})]$ iff $\forall x\ \langle d,x\rangle \in \mathcal{E}^t[\mathbf{r}]$ or $\langle e,x\rangle \in \mathcal{E}^f[\mathbf{s}]$
$\langle d,e\rangle \in \mathcal{E}^f[(\supseteq\ \mathbf{r}\ \mathbf{s})]$ iff $\exists x\ \langle d,x\rangle \in \mathcal{E}^f[\mathbf{r}]$ and $\langle e,x\rangle \in \mathcal{E}^t[\mathbf{s}]$

$\langle d,e\rangle \in \mathcal{E}^t[(\mathbf{and\text{-}role}\ \mathbf{r}_1 \ldots \mathbf{r}_n)]$ iff for each i, $\langle d,e\rangle \in \mathcal{E}^t[\mathbf{r}_i]$
$\langle d,e\rangle \in \mathcal{E}^f[(\mathbf{and\text{-}role}\ \mathbf{r}_1 \ldots \mathbf{r}_n)]$ iff for some i, $\langle d,e\rangle \in \mathcal{E}^f[\mathbf{r}_i]$
$\langle d,e\rangle \in \mathcal{E}^t[(\mathbf{restr}\ \mathbf{r}\ \mathbf{c})]$ iff $\langle d,e\rangle \in \mathcal{E}^t[\mathbf{r}]$ and $\langle e\rangle \in \mathcal{E}^t[\mathbf{c}]$
$\langle d,e\rangle \in \mathcal{E}^f[(\mathbf{restr}\ \mathbf{r}\ \mathbf{c})]$ iff $\langle d,e\rangle \in \mathcal{E}^f[\mathbf{r}]$ or $\langle e\rangle \in \mathcal{E}^f[\mathbf{c}]$

For any two concepts, $\mathbf{c}'$ and $\mathbf{c}$, $\mathbf{c}'$ *subsumes* $\mathbf{c}$ if, for every situation, $\langle \mathcal{D}, \mathcal{E}^t, \mathcal{E}^f \rangle$, $\mathcal{E}^t[\mathbf{c}'] \supseteq \mathcal{E}^t[\mathbf{c}]$ and $\mathcal{E}^f[\mathbf{c}'] \subseteq \mathcal{E}^f[\mathbf{c}]$.

How well does this semantics reflect intuitions about the meaning of concepts and roles and how well does it model subsumption?

On the first point, the semantics does rather well. Define a *model* as a situation where, for every concept $\mathbf{c}$, the positive and negative extensions of $\mathbf{c}$ are disjoint and together exhaust the domain of the model, and, similarly, the positive and negative extensions of roles and bindings are disjoint and exhaustive. In models the above semantics reduces to the standard two-valued semantics for frame-based description languages. The semantics given here is a minimal mutilation required to go from a two-valued semantics to a four-valued semantics. There is nothing added besides what is needed to get from two truth values to four truth values, in this particular way.

Semantics based on four truth values, such as this one, the propositional semantics of [Levesque, 1984], and the first-order semantics of [Patel-Schneider, 1985], are reasonable for systems with limited reasoning power. Such systems do not have total information, thus the presence of truth-value gaps, and also cannot resolve inconsistencies, thus allowing for inconsistent situations.

On the second point, modeling subsumption, the semantics also does fairly well. First, since the set of models is a subset of the set of situations, and since the requirements for subsumption reduce to the standard ones on models, reasoning in this semantics is sound with respect to the standard semantics. This soundness is an important requirement if the semantics is to capture some of the intuitive ideas behind frame-based description languages.

Second, the actual subsumption relationships in this semantics form an interesting set (as will be shown more fully later). The sort of subsumption relationships that are valid are the simple ones, such as

(and person (all (restr friend male) doctor))

subsuming

(and person
(all friend
(and doctor (some speciality)))).

Subsumption relationships that are valid in the standard two-valued semantics but not here involve reasoning using the law of the excluded middle or *modus ponens*. For example,

(and person
(all friend doctor)
(all (restr friend doctor) (some speciality))),

i.e., a **person** whose **friends** are all **doctors** and whose **friends** who are **doctors** all have some **speciality**, is not subsumed by

(and person
(all friend (some speciality))),

i.e., a **person** whose **friends** all have some **speciality**. This is because, in four-valued situations, it is possible that some friend might both be a doctor and not be a doctor, as well as not specializing, thus falsifying **(all friend (some speciality))**, but falsifying neither **(all friend doctor)** nor **(all (restr friend doctor) (some speciality))**.[6]

IV Computing Subsumption

Showing that subsumption in this semantics is less powerful than subsumption in the standard semantics does not show that it is any easier to compute. To do this requires defining an algorithm, showing that is an algorithm for determining subsumption here, and calculating how fast it runs.

[6]Note that in a three-valued semantics, subsumptions like this one are still valid. The existence of a friend that is neither a doctor nor not a doctor prevents both the classes from being true and does not force either to be false, thus doing nothing to make the subsumption invalid.

The subsumption algorithm works as follows. First, use the following equivalences to transform roles and concepts into canonical form.

1. commutativity and associativity of **and** and **and-role**
2. **(all r (and c_1 c_2))** $\equiv$ **(and (all r c_1) (all r c_2))**
3. **(rvm r (and-role s_1 s_2))** $\equiv$ **(and (rvm r s_1) (rvm r s_2))**
4. **(rvm r (restr s c))** $\equiv$ **(and (rvm r s) (all r c))**
5. **(restr (restr r c_1) c_2)** $\equiv$ **(restr r (and c_1 c_2))**
6. **(and-role (restr r_1 c)r_2)** $\equiv$ **(restr (and-role r r_2) c)**

In canonical form, conjuncts of a concept are not themselves conjuncts, the second argument of **alls** is not a conjunct, the second argument of **rvms** is an atomic role, and all other roles are of the form **(and-role s_1 ... s_n)** or **(restr (and-role s_1 ... s_n) c)**, where each s_i is an atomic role.

Then **(and c_1 ... c_n)** is subsumed by **(and c'_1 ... c'_m)**, where both are in canonical form, iff for each i from 1 to m there is a j in the range from 1 to n such that one of the following cases holds:

1. c'_i is an atomic concept and $c_j = c'_i$,
2. c'_i = **(some r')** and c_j = **(some r)** with **r'** subsuming **r**,
3. c'_i = **(all r' d')** and c_j = **(all r d)** with **d'** subsuming **d** and **r'** subsumed by **r**,
(If **d'** is of the form **(all s'_1 d'_1)**, then use **(restr r' (some s'_1))** instead of **r'**, and if d'_1 is of the form **(all s'_2 d'_2)** use **(restr s'_1 (some s'_2))** instead of s'_1, *etc.*)
4. c'_i = **(rvm r' s')** and c_j = **(rvm r s')** with **r'** subsumed by **r**, (recall that **s'** must be an atomic role),
5. c'_i = **(sd d' b'_1 ... b'_n)** and there is some j such that c_j = **(sd d b_1 ... b_m)** and **d** is subsumed by **d'** and, for each i, if b'_i is of the form **($\subseteq$ r' s')**, then **(and-role s_1 ... s_l)** is subsumed by **s'**, where, for each k, there is some j such that b_j = **($\subseteq$ r s_k)** and **r** subsumes **r'**, and, if b'_i is of the form **($\supseteq$ r' s')**, then **(and-role r_1 ... r_l)** is subsumed by **r'**, where, for each k, there is some j such that b_j = **($\supseteq$ r_k s)** and **s** subsumes **s'**.

Also **(restr (and-role s_1 ... s_m) c)** is subsumed by **(restr (and-role s'_1 ... s'_n) c')** iff for all i there exists j such that $s_j = s'_i$ and **c** is subsumed by **c'**. The other cases for roles are the obvious modifications to this rule.

There are two very important properties of these algorithms.

Theorem 1 *The algorithms correctly determine subsumption in this semantics,* i.e., *they are both sound and complete.*[7]

Theorem 2 *The algorithms run in time proportional to the square of the sum of the sizes of the two expressions.*

Therefore, subsumption in this semantics is easy to compute, as opposed to subsumption for $\mathcal{FL}$ in the two-valued semantics.

This computational gain would not be very interesting if the subsumption relationships in the semantics are totally uninteresting. Of course, something is lost, but the remaining subsumption relationships must, at least, form an interesting subset of the subsumption relationships of the standard two-valued semantics.

[7]The proofs of these theorems are too long to fit in this paper but will be included in a longer paper on four-valued semantics for frame-based description languages.

Fortunately, this is the case. An examination of the subsumption algorithms given above shows that subsumption in this semantics is very closely related to the subsumptions computed by the subsumption algorithm of NIKL. (The only important difference between the two is the caveat attached to case 3 above.) Both are examples of "structural subsumption", where each piece of structure in the subsuming concept or role must be mirrored by an appropriate piece of structure in the subsumed concept or role. As such, they capture an interesting subset of the subsumption relationships in the standard semantics for frame-based description languages, one that contains the simple subsumption relationships and leaves the complex and hard-to-compute ones out.

V Summary

What has been gained from this new semantics for frame-based description languages? First of all, the semantics is a reasonable semantics, especially when considering systems with limited reasoning capabilities. Second, subsumption in this semantics is easy to compute, at least for the language given here. Also, the valid subsumption relationships form an interesting set—one that includes the easy subsumptions and leaves out the less obvious ones. This set corresponds closely to the set of subsumption relationships computed in NIKL, lending a degree of credence to that set. Third, certain extensions to the language, such as adding negation and disjunction or adding compositions of roles or structural descriptions as in NIKL, cause no problems.

However, there are some problems with the semantics. First, the semantics is not as intuitive as the two-valued semantics. This is a problem with all alternative semantics but the four-valued semantics given here is still a reasonable semantics, especially for limited reasoners. Second, subsumption in this semantics gets only the very easy cases, leaving many that might be important, such as those involving a single application of *modus ponens*. It seems that, in order to get a uniform, simple semantics with a fast subsumption algorithm, there is no way around this extreme weakness.

The worst problem with this semantics is its inability to solve computational problems involving number restrictions (generalizing **some** to **at-least** and adding **at-most**). Although it is easy to define these concepts in this semantics, by using the size of sets in the situations, reasoning with numbers is hard, just as it is in the standard semantics. This is because identity is a two-valued notion, which legitimizes more deductions than are usual in a four-valued semantics. The most promising way of getting around this problem is to go to a four-valued notion of equality, which, of course, further changes the semantics from the standard one, and introduces several complications to the analysis of subsumption.

The most important point about this new semantics is that it forms a principled way to defuse the tradeoff between expressive power and computational complexity. It justifies a limited set of subsumption relationships that are easy to compute and, moreover, captures an interesting subset of the standard subsumption relationships. This is not a total solution, because no total solutions are possible (unless $P = NP$) and is not even a finalized solution, because it does not yet handle number restrictions. However, this semantics does form an important step towards a principled, computationally tractable yet expressively powerful, knowledge representation system, and thus serves to alleviate the computational problems of frame-based description languages reported by Brachman and Levesque.

Acknowledgments

Ron Brachman and Pat Hayes, as the previous and current leaders of the knowledge (representation) group at SPAR, have been instrumental in maintaining a comfortable research environment for me. Hector Levesque and Ron Brachman, through their investigation of the complexity of computing subsumption in standard semantics for frame-based description languages, provided the impetus for this research. Dave McAllester and Dan Carnese provided useful comments on earlier drafts of the paper.

References

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

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

[Brachman et al., 1983] Ronald J. Brachman, Richard E. Fikes, and Hector J. Levesque. KRYPTON: a functional approach to knowledge representation. *IEEE Computer*, 16(10, Special Issue on Knowledge Representation):67–73, October 1983.

[Brachman et al., 1985] Ronald J. Brachman, Victoria Pigman Gilbert, and Hector J. Levesque. An essential hybrid reasoning system: knowledge and symbol level accounts of KRYPTON. In *Proceedings IJCAI-85*, pages 532–539, August 1985.

[Frisch, 1985] Alan M. Frisch. Using model theory to specify AI programs. In *Proceedings IJCAI-85*, pages 148–154, August 1985.

[Levesque, 1984] Hector J. Levesque. A logic of implicit and explicit belief. In *Proceedings AAAI-84*, pages 198–202, August 1984.

[Levesque and Brachman, 1985] Hector J. Levesque and Ronald J. Brachman. A fundamental tradeoff in knowledge representation and reasoning (revised version). In Ronald J. Brachman and Hector J. Levesque, editors, *Readings in Knowledge Representation*, pages 42–70, Morgan Kaufmann Publishers, Los Altos, California, 1985.

[Patel-Schneider, 1984] Peter F. Patel-Schneider. Small can be beautiful in knowledge representation. In *Proceedings IEEE Workshop on Principles of Knowledge-Based Systems*, pages 11–16, December 1984.

[Patel-Schneider, 1985] Peter F. Patel-Schneider. A decidable first-order logic for knowledge representation. In *Proceedings IJCAI-85*, pages 455–458, August 1985.

[Patel-Schneider et al., 1984] Peter F. Patel-Schneider, Ronald J. Brachman, and Hector J. Levesque. ARGON: knowledge representation meets information retrieval. In *Proceedings of The First Conference on Artificial Intelligence Applications*, pages 280–286, December 1984.

[Schmolze, 1985] James G. Schmolze. The language and semantics of NIKL. Draft, BBN Laboratories, April 1985.

[Schmolze and Israel, 1983] James G. Schmolze and David J. Israel. KL-ONE: semantics and classification. In *Research in Knowledge Representation for Natural Language Understanding—Annual Report, 1 September 1982–31 August 1983*, pages 27–39, Technical Report 5421, BBN Laboratories, 1983.