A Decidable First-Order Logic for Knowledge Representation

Peter F. Patel-Schneider

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

ABSTRACT

Even though logic has played an important role in knowledge representation (KR) research, there has been little effort expended on devising decidable logics for KR. Most modifications to logic suggested for KR are either extensions to first-order logic (e.g., to handle non-monotonicity) or *ad hoc* changes in its inference mechanism. This paper presents a variant of first-order relevance logic that has a decidable algorithm for determining tautological entailment. Although this logic is considerably weaker than standard first-order logic, it can be used effectively in a KR system when semantically correct answers to queries are required within a finite amount of time.

Knowledge representation has always played a major role in Artificial Intelligence research. However, the exact nature of that role has been left undefined. So-called knowledge representation (KR) systems can range from packages for manipulating data structures to complete AI systems that plan or do resource management. The view of KR used here (similar to that espoused in [Levesque, 1984a]) is that a KR system is to provide a fact management service. That is, the job of a KR system is to manage a knowledge base (KB) in some representation language (including both syntax and semantics) and to answer questions about what semantically follows from the current KB.[1] An important part of this view, when applied to KR, is that the KR system must not only be sound, it must also be complete. This means that every answer given by the system, including "answering'' 'unknown' or even refusing to answer, must be sanctioned by the semantics. One important effect of this formulation is that the behavior of the KR system cannot depend on extra-linguistic considerations, such as length of derivations or elapsed time.

The main problem in this formulation of KR is to select or create a suitable representation language, with an appropriate semantics. There are several properties desirable in such a language. First, the language should have expressive power sufficient to represent many situations. Second, the semantics should correspond to our intuitive notions about the meanings of the constructs of the language. Third, there should be a decidable and, moreover, reasonably fast procedure for computing whether some statement follows from a KB.

Obviously a representation language in which it is impossible to formulate many problem definitions and questions is inadequate. However, frame-based KR systems, as popular as they are, suffer from just this inability. The kind of language typically used in such systems cannot adequately express most disjunctions or similar weak statements, which are required in many contexts. This shortcoming has often been sidestepped by adding extra-linguistic hooks to frame-based systems such as user-definable procedures that perform some part of the inference process. Such actions destroy the semantics of the representation language and therefore systems that resort to such subterfuges do not satisfy the view of KR espoused here.

[1]This is in accordance with the view expounded by McDermott (McDermott, 1978] and, more recently, by Frisch (Frisch, 1985). However, most current KR systems do not satisfy this view.

It is generally accepted in KR that the expressive power of at least standard first-order logic (FOL) is needed in a representation language. FOL itself is a good candidate because it has a semantics that corresponds well with our intuitive ideas about the world. Unfortunately, this logic has a severe problem—determining whether one sentence follows from another is, in general, undecidable. This makes a KR system built on FOL unsuitable for AI systems that depend on receiving answers.

This problem with FOL has led to many attempts to create KR systems based on FOL that always produce answers. Most of these systems retain the syntax of FOL while modifying its inferences in some way. The crudest of them simply take a theorem prover for FOL and place some *ad hoc* restrictions on it, such as terminating the search for a proof after a pre-set amount of time or a certain number of proof steps. Such modifications produce systems that cannot be given an adequate semantics and have no means of completely characterizing answers except by referring to the actions of the modified theorem prover. This destroys most of the advantages of using logic in the first place.

A more principled approach is to devise a semantics for first-order sentences that can model something like these limitations on inferences. For example, Konolige's system [Konolige, 1985] includes a Bet of inference rules in the model structure and Frisch and Allen [Frisch and Allen, 1982] encode a variant of first-order logic back into FOL, thus capturing a particular set of inferences. Such systems have been characterised in [Levesque, 1984b] as *syntactic* variants because their semantic structures have to include syntactic entities. Because their semantics is used to model the inference process, every choice of how this process proceeds shows up in the semantics. This makes for a very complicated semantics which only reflects the inference process and does not give any guidance as to what the correct inferences should be.

What is needed is a semantics that does not involve the inference process but is instead based on the usual model theoretic ideas of interpretations and truth and falsity, thus allowing the semantics to dictate the correct inferences as opposed to vice *versa.* This semantics must correspond with some part of our intuitions about the world while also having decidable inference. It should also be weaker than FOL so that all reasoning in it is sound with respect to FOL. The semantics might not be as intuitive as that of FOL but the idea is to develop a semantics that captures part of our intuitions and trade off whatever is lost for decidability and, hopefully, tractability of inference.

I Relevance Logic

One possible logic for this purpose is the logic of tautological entailments—a simple type of propositional relevance logic [Anderson and Belnap, 1975]. Propositional tautological entailment has been used by Levesque [Levesque, 1984b] to model explicit propositional belief. Relevance logic has also been used by Shapiro [Shapiro and Wand, 1976] as part of a semantic network

KR system. In fact, one of the main proponents of relevance logic suggested that it would be a useful logic for KR [Belnap, 1977].

The syntax of the logic of propositional tautological entailment is the same as that of standard propositional logic (PL), but without an implication operator. Its semantics is based on the four-valued *setups* of propositional relevance logic (as opposed to the two-valued *assignments* of PL). In these setups propositional letters and, from them, sentences can be assigned *true* or *false* (as in PL) but can also be assigned *neither* true nor false or *both* true and false. In some ways these setups model the state of affairs that may be encountered by KR systems better than two-valued assignments do because such systems may have no information on whether some proposition is true or false and also may have been, inadvertently, told that some other proposition is both true and false [Belnap, 1977]. A key difference between this way of defining truth and falsity and that used in PL is that there are no sentences which are true in all setups nor are there any which are false in all setups. For example, $p \vee \neg p$ is neither true nor false in a setup that supports neither p's truth nor p's falsity.

From this basis, the idea of propositional tautological entailment (a relevance logic analogue of implication) can be defined. Here α entails β (written $\alpha \rightarrow \beta$) iff β is true whenever α is and α is false whenever β is. If $\alpha \rightarrow \beta$ then β follows from α in PL, because the set of two-valued assignments is included in the set of setups. However, entailment is a much weaker notion than implication. For example, $a \wedge \neg a \not\rightarrow b$ and $a \not\rightarrow b \vee \neg b$, showing that a contradiction does not entail everything nor does every sentence entail a tautology. Further, $a \wedge (\neg a \vee b) \not\rightarrow b$, showing that *modus ponens* is not a valid rule for entailment.

In fact, there is a simple algorithm for determining if one sentence entails another. To determine if $\alpha \rightarrow \beta$ first put α and β into conjunctive normal form (CNF). Then $\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta_1 \wedge \ldots \wedge \beta_m$ (where each α_j and β_i are disjunctions of primitive propositions) iff for each β_i there is an α_j such that $\alpha_j \subseteq \beta_i$ (treating α_j and β_i as sets). This algorithm is a bit disappointing at first glance because putting α and β into CNF can cause an exponential increase in their size and the main rationale for using tautological entailment was to have a faster algorithm. However, KBs are almost always a conjunction of many facts, each of which is much smaller than the KB as a whole. This means that KBs are naturally in almost CNF so normalising them will not increase their size drastically. Then, if α and β are in CNF, computing whether $\alpha \rightarrow \beta$ takes time proportional to the product of their sizes [Levesque, 1984b]. On the other hand the problem of determining if α implies β is co-NP complete even if α and β are in CNF.

Therefore propositional tautological entailment would be a good choice for a KR system provided that propositional sentences were expressively adequate for KR. This logic sanctions a subset of the inferences of PL that is easy to compute and corresponds to a semantics that is at least plausible for KR.

However, propositional logic is generally considered not expressive enough for a KR system. This leads immediately to the idea of using first-order tautological entailment instead of propositional tautological entailment as a logic for KR. First-order tautological entailment is to FOL as propositional tautological entailment is to PL. The syntax of the logic is the same as that of FOL, again without an implication sign. The semantics of the logic is very much like the semantics of FOL except for the provisions for the four relevance logic truth values.

The development of the semantics starts with first-order situations, the analogue of first-order models.

Definition 1 *A situation consists of a non-empty set D, the domain of the situation, and three mappings, h, t, and f. h maps each function letter, f_j^n, into a function from D^n to D, and t and f map each predicate letter, A_j^n, into an n-ary relation on D.*

The mapping h is the usual mapping of function letters (and constant letters) into functions over the domain (and constants in the domain). The mappings t and f determine which atomic formulae are true and which are false and correspond to the interpretation function of FOL. Two mappings are needed because an atomic formula can be assigned true, false, neither, or both.

Definition 2 A variable map *is a mapping from variables into some set. If ν is a variable map into D, x is a variable, and d is an element of D, then ν_d^x is a variable map into D with*

$\nu_d^x(y) = d$, *if* $y = x$, *and* $\nu_d^x(y) = \nu(y)$, *otherwise.*

Given a situation, s, and a variable map, ν, a mapping, ν_s^*, from terms into the domain of s can be defined as follows:

$\nu_s^*(x) = \nu(x)$, *if x is a variable,*

$\nu_s^*(f_j^n(t_1, \ldots)) = (h(f_j^n))(\nu_s^*(t_1), \ldots)$, *otherwise.*

Definition 3 *The support relationships of first-order relevance logic are defined as follows:*

$s, \nu \models_t A_j^n(t_1, \ldots, t_n)$ *iff* $\langle \nu_s^*(t_1), \ldots, \nu_s^*(t_n) \rangle \in t_s(A_j^n)$

$s, \nu \models_f A_j^n(t_1, \ldots, t_n)$ *iff* $\langle \nu_s^*(t_1), \ldots, \nu_s^*(t_n) \rangle \in f_s(A_j^n)$

where s is a situation, ν is a variable map into the domain of s, A_j^n is a predicate letter, and t_i is a term.

This is just the obvious way of assigning meaning to atomic formulae. (The way to read $s, \nu \models_t \alpha$ ($s, \nu \models_f \alpha$) is "s supports the truth (falsity) of α under ν".) These relationships are extended to arbitrary first-order formulae by the following rules:

1. $s, \nu \models_t \neg\alpha$ iff $s, \nu \models_f \alpha$
   $s, \nu \models_f \neg\alpha$ iff $s, \nu \models_t \alpha$
2. $s, \nu \models_t \alpha \vee \beta$ iff $s, \nu \models_t \alpha$ or $s, \nu \models_t \beta$
   $s, \nu \models_f \alpha \vee \beta$ iff $s, \nu \models_f \alpha$ and $s, \nu \models_f \beta$
3. $s, \nu \models_t \alpha \wedge \beta$ iff $s, \nu \models_t \alpha$ and $s, \nu \models_t \beta$
   $s, \nu \models_f \alpha \wedge \beta$ iff $s, \nu \models_f \alpha$ or $s, \nu \models_f \beta$
4. $s, \nu \models_t \forall x \alpha$ iff for all $d \in D$ $s, \nu_d^x \models_t \alpha$
   $s, \nu \models_f \forall x \alpha$ iff for some $d \in D$ $s, \nu_d^x \models_f \alpha$
5. $s, \nu \models_t \exists x \alpha$ iff for some $d \in D$ $s, \nu_d^x \models_t \alpha$
   $s, \nu \models_f \exists x \alpha$ iff for all $d \in D$ $s, \nu_d^x \models_f \alpha$

It is easy to see that this semantics is very similar to standard Tarskian semantics except for the change from truth values of true and false to truth values of subsets of true and false. As argued above, this change has some benefits with respect to KR. As in the propositional case, there are no tautologies (sentences which are true in all situations) in this semantics.

First-order tautological entailment is then defined by:

Definition 4 *If α and β are first-order sentences, α entails β iff for all situations, s, and all variable maps, ν, if $s, \nu \models_t \alpha$ then $s, \nu \models_t \beta$ and if $s, \nu \models_f \beta$ then $s, \nu \models_f \alpha$.*

First-order tautological entailment captures a similar set of inferences to those of the propositional version, where reasoning by contradiction and *modus ponens* are not valid. Based on the success with propositional tautological entailment, one would hope that the first-order version is decidable. However, it turns out that first-order tautological entailment can be used to simulate first-order implication and thus is undecidable. This unfortunate finding destroys most of the utility of this logic for KR.

II A Variant of Relevance Logic

First-order tautological entailment, even though it is undecidable, is a very interesting starting place for further investigation because it forms a subset of FOL, based on a semantics that is reasonable for KR, and that is, in some sense, much weaker than FOL. What is needed then is a variant of first-order tautological entailment which is decidable. One problem with first-order tautological entailment is that quantification is equivalent to infinite conjunction or disjunction. It is this equivalence that makes quantification too powerful. For example, $(Pa \wedge Qa) \vee (Pb \wedge Qb)$ entails $\exists x\, Px \wedge Qx$ thus preventing the creation of an entailment algorithm similar to the propositional one.

To prevent this sort of entailment the quantifiers can be given an intuitionistic reading, under which interpretation the formula $\exists x\ Px$ would be read "there exists a known individual for which the P is true". To formalise this change requires a significant modification of first-order relevance logic semantics. Instead of talking about situations, sets of situations are needed. For $\exists x\ Px$ to be true in a set of situations there must be some domain object, a single object common across all the situations, such that P is true for that object in each of the situations.

The semantics of this variant starts out with first-order situations, variable maps, and the support relationships for atomic formulae just as in regular first-order relevance logic. However, the next step is the definition of *compatible* sets of situations. A compatible set of situations is a set of situations with the same domain and the same mapping of function letters to functions. In other words, the situations differ only on their assignments of truth and falsity. Given S, a compatible set of situations each with domain D, and ν, a variable map into D, the two support relations for this logic, $S,\nu \models_t \alpha$ and $S,\nu \models_f \alpha$, are defined as follows:

1. $S,\nu \models_t \forall x\alpha$ iff for all $d \in D$ $S,\nu^x_d \models_t \alpha$
   $S,\nu \models_f \forall x\alpha$ iff for some $d \in D$ $S,\nu^x_d \models_f \alpha$
2. $S,\nu \models_t \exists x\alpha$ iff for some $d \in D$ $S,\nu^x_d \models_t \alpha$
   $S,\nu \models_f \exists x\alpha$ iff for all $d \in D$ $S,\nu^x_d \models_f \alpha$
3. $S,\nu \models_t \alpha$ iff for all $s \in S$ $s,\nu \models_t \alpha$
   $S,\nu \models_f \alpha$ iff for all $s \in S$ $s,\nu \models_f \alpha$
   for α quantifier free

where $s,\nu \models_t \alpha$ and $s,\nu \models_f \alpha$ are as above. So $\exists x\alpha$ is true in S under variable map ν if there is some domain element, common across all the situations in S, which, when taken as the mapping of x, makes α true in each situation. The same formula is false if all domain elements, when taken as the mapping of x, make α false in each situation. This means that it is fairly easy for an existential to be true (although more difficult than in regular first-order relevance logic), but quite hard for it to be false, which is as things should be. (Note that the semantics given here is defined only on formulae in prenex form (PF). There is a more complicated semantics for arbitrary formulae[2] which is equivalent to this one for formulae in PF and which has a very strong normal form theorem allowing all formulae to be converted to prenex CNF (PCNF) without changing their meaning.[3])

The next step is to define entailments in this logic. As it turns out, there are three different versions of entailment here. Recall that if β follows from α, then β must be true whenever α is and, conversely, α must be false whenever β is. These two conditions are equivalent by definition in standard two-valued logic where being false is equivalent to not being true and also happen to be equivalent in regular first-order tautological entailment. However, the conditions are not equivalent in this variant, giving rise to three versions of entailment, *t-entailment* (written $\rightarrow_t$), which carries the first condition, *f-entailment* (written $\rightarrow_f$), which carries the second condition, and *tf-entailment* (written $\rightarrow$), which carries both conditions.

So what are these three versions of entailment like? First of all, they are all very weak—weaker than regular first-order tautological entailment and much weaker than implication. Second, any sentence will entail its PCNF version and thus the order of conjuncts and disjuncts does not matter. Third, as in other relevance logics, *modus ponens* is not a valid rule. This, in large part, is what makes the logic very weak. Fourth, sentences can be weakened by removing elements of conjunctions and adding elements to disjunctions. Finally, the entailments for quantifiers can be summed up as follows:

$\forall x\ Px$	$\rightarrow$	Pa	Pa	$\rightarrow$	$\exists x\ Px$
$\forall x\ Px$	$\rightarrow_t$	$Pa \wedge Pb$	$Pa \vee Pb$	$\not\rightarrow_t$	$\exists x\ Px$
$\forall x\ Px$	$\not\rightarrow_f$	$Pa \wedge Pb$	$Pa \vee Pb$	$\rightarrow_f$	$\exists x\ Px$
$\forall x\ Px$	$\not\rightarrow$	$Pa \wedge Pb$	$Pa \vee Pb$	$\not\rightarrow$	$\exists x\ Px$

This shows that all versions of entailment have a universal entailing a single instantiation of itself and a single instantiation entailing an existential. Also, a universal t-entails the conjunction of any number of instantiations whereas a disjunction of instantiations does not t-entail an existential. For f-entailment the opposite is true, and so, of course, neither way works for tf-entailment.

Of these three versions of entailment, t-entailment seems to be the best for KR. It is preferable to tf-entailment, because it is stronger. It and f-entailment are duals but the entailments retained by t-entailment (such as $\forall x\ Px \rightarrow_t Pa \wedge Pb$) are more natural for KR than those retained by f-entailment (such as $Pa \vee Pb \rightarrow_f \exists x\ Px$). This is especially true when taking the intuitionistic view that to demonstrate the existence of an object a particular object must be produced. Further, this last type of entailment is a sort of reasoning by cases, a type of reasoning generally not valid in relevance logic. These considerations indicate that t-entailment is the best of the three variants for KR so it will be emphasised from now on.

This logic would not be very interesting it if did not have a decidable algorithm for determining entailment. The first step in developing such an algorithm for this variant of relevance logic is a Skolemisation theorem for entailment which states that for all three versions of entailment $\alpha \rightarrow \beta$ iff $\alpha_{S\exists} \rightarrow \beta_{S\forall}$, where $\alpha_{S\exists}$ is α with all existentially quantified variables Skolemised and $\beta_{S\forall}$ is β with all universals Skolemised. Then the following theorem characterises t-entailment between sentences in normal form:

Theorem 1 *If α and β are sentences in Skolemised PCNF ($\alpha = \forall \vec{x} \bigwedge \alpha_j$ and $\beta = \exists \vec{z} \bigwedge \beta_i$) then $\alpha \rightarrow_t \beta$ iff there exists θ, a substitution for $\vec{z}$, such that for each β_i there exists some α_j and ψ, a substitution for $\vec{x}$, such that $\alpha_j\psi \subseteq \beta_i\theta$ (treating $\alpha_j\psi$ and $\beta_i\theta$ as sets of literals).*

An algorithm for computing entailment between two sentences in Skolemised PCNF can now be derived. Consider α and β as above. For each α_j and β_i, compute a set of most general substitutions Θ_{ij} such that for $\theta \in \Theta_{ij}$, $\alpha_j\theta \subseteq \beta_i\theta$. For each element of Θ_{ij}, define a new substitution the same as that element except that occurrences of elements of $\vec{x}$ are systematically replaced by variables occurring nowhere else. Let Ψ_{ij} be the set of these substitutions and let $\Psi_i = \bigcup_j \Psi_{ij}$. Then $\alpha \rightarrow_t \beta$ iff there is some substitution ϕ which is the most general unifier of some $\{\psi_i : \psi_i \in \Psi_i\}$. The substitution ϕ plays the role of θ and all the ψ in the theorem above.

This algorithm does demonstrate that t-entailment in this logic is decidable. Although its worst case behavior is exponential in the size of α and β it will always terminate and under normal conditions, where clauses are not long and there are many different predicates, will be quite fast. This characterisation also gives a good indication of how t-entailment works showing that the five informal comments above are the basis for a complete description of t-entailment. Note, however, that it is the semantics that *defines* t-entailment, not this algorithm.

Although axiomatisations are not important in this view of KR, the logic presented here is sufficiently removed from familiar logics that an axiomatisation of the three versions of entailment is useful for illustrative purposes. The axiomatisation for entailment given below was designed to be easily understandable. It includes an axiom simply stating the equivalence of a formula and is PCNF version. Although this is a decidedly unusual axiom, it is an effective statement and thus qualifies as a real axiom. Because of the lack of rules such as *modus ponens* in this logic, the only other way to get this equivalence would be to include a set of about fifteen axioms, an approach much harder to understand.

[2]The full semantics, along with proofs of all the theorems, can be found in [Patel-Schneider, 1985], an expanded version of this paper.

[3]The splitting and combining of quantifiers allowed in FOL are not valid in this logic. For example $\exists x\ Px \vee \exists y\ Qy$ is equivalent to $\exists x \exists y\ Px \vee Qy$ but not to $\exists x\ Px \vee Qx$.

The axiomatisation of entailment is then:

1. if $\alpha \rightarrow \beta$ then $\alpha \rightarrow_t \beta$ and $\alpha \rightarrow_f \beta$
2. if $\alpha \rightarrow \beta$ and $\beta \rightarrow \gamma$ then $\alpha \rightarrow \gamma$,
   this rule also holds for t-entailment and f-entailment
3. $\alpha \rightleftharpoons \alpha_{PCNF}$ where α_{PCNF} is α in PCNF,
   this includes axioms such as $\exists x\alpha \rightleftharpoons \neg\forall x\neg\alpha$
   and $\exists x(\alpha \vee \beta) \rightleftharpoons (\exists x\alpha) \vee \beta$ for x not free in β
4. $\alpha \rightarrow \alpha \vee \beta$ $\qquad$ $\alpha \wedge \beta \rightarrow \alpha$
5. $\alpha^x_a \rightarrow \exists x\alpha$ $\qquad$ $\forall x\alpha \rightarrow \alpha^x_a$
6. if $\alpha^x_a \rightarrow \beta$ then $\exists x\alpha \rightarrow \beta$ for a not in α or β
   if $\alpha \rightarrow \beta^x_a$ then $\alpha \rightarrow \forall x\beta$ for a not in α or β,
   the same rules hold for t-entailment and f-entailment
7. if $\alpha \rightarrow_t \beta$ and $\alpha \rightarrow_t \gamma$ then $\alpha \rightarrow_t \beta \wedge \gamma$
8. if $\alpha \rightarrow_f \gamma$ and $\beta \rightarrow_f \gamma$ then $\alpha \vee \beta \rightarrow_f \gamma$

The last two rules serve to distinguish the three variants of entailment from each other.

This axiomatisation is sound for each variant of entailment and is complete for t- and f-entailment. One other interesting item about the axiomatisation of t-entailment is that it is almost exactly equivalent to the syntactically motivated limitation of inference by Frisch and Allen called *knowledge retrieval* [Frisch and Allen, 1982].

III Conclusion

So what has been gained from this new logic? In essence, the logic provides a decidable subset of implication that does not contain *any* of the harder types of inference. It does not have the rule of *modus ponens* so no chaining can be done. It also does not do reasoning by contradiction or reasoning by cases. What *is* provided is a simple set of inferences, derived directly from the semantics, that performs *retrieval* of first-order sentences from a knowledge base specified as first-order sentences, much as database systems perform retrieval of facts from a database. Many inferences are not provided but this is the price that must be paid to guarantee decidability.

This decidability of entailment is not the only feature of the logic that makes it suitable for use in a KR system, although it is the most important one. The syntax of the logic is the same as that of FOL so, in some sense, it is possible to state in the new logic anything that can be stated in FOL. Also, because a subset of implication is provided, knowledge-based systems can treat sentences as sentences of FOL provided that they realise that the inferences provided are not complete. Further, since the semantics of the logic is closely related to regular first-order relevance logic semantics, the logic does not suffer from some of the problems and paradoxes of implication. Finally, the semantics is firmly based on the usual model theoretic ideas of truth and falsity and serves to dictate the allowable inferences and not *vice versa.* This allows a simple characterisation of entailment in terms of the truth of sentences. Together these points make the logic more suitable for KR than systems built by *ad hoc* modifications to a theorem prover or systems based on a particular set of inferences (and otherwise semantically unmotivated).

However, the logic presented above is simply the start of work on a decidable first-order logic-based KR system. Although this logic, especially t-entailment, has many of the right properties required for such a system there may be stronger logics that retain decidability. For example, there are stronger versions of relevance logic that might be used as the basis of such a logic. The search for such a logic is, of course, aided by this demonstration of one logic meeting the requirements.

One way to strengthen the logic developed here without destroying decidability is by adding to it a terminological reasoner like the terminological component of KRYPTON (Brachman *et al.,* 1086]. To do this correctly would require building a full semantics for the combination of terminological reasoning and t-entailment, perhaps similar to the semantics in (Brachman *et al.*, 1085]. A wrinkle in developing terminological reasoners for this new logic is determining which language to use. As shown in [Brachman and LeveBque, 1084], terminological reasoning can be computationally intractable for even very simple terminological languages under standard semantics. If the KR system as a whole is to perform adequately, the terminological component must be tractable. One possibility for producing tractable terminological reasoners is to perform the same sort of reduction on the terminological component as the logic devised here does on the assertional component. This would produce a semantics of frame subsumption that only catches the easy subsumptions and leaves the hard and time-consuming ones out.

Any KR system based on this new logic is going to be quite weak. One way of making such a system stronger is to make entailment be only the first method for answering questions. If entailment fails to produce an answer then a stronger method could be used, perhaps some domain specific method. The advantage of this is that entailment is a semantically motivated and well-defined notion so that it is possible to understand what its failure to produce an answer means. An interesting idea for the fall-back method is to use a stronger logical system. A first idea for such a fall-back is, of course, FOL, but other logics, such as regular first-order tautological entailment, could be used.

In summary, the decidable first-order logic presented here forms an important first step toward building decidable semantically-motivated KR systems.

References

[Anderson and Belnap, 1975] Anderson, Alan R., and Nuel D. Belnap, Jr. *Entailment: The Logic of Relevance and Necessity,* Volume I. Princeton University Press, 1975.

[Belnap, 1977] Belnap, Nuel D., Jr. "A Useful Four-Valued Logic". In *Modern Uses of Multiple-Valued Logic,* ed. G. Epstein and J. M. Dunn. Boston: Reidel, 1977, pp. 8-37.

[Brachman and Levesque, 1984] Brachman, Ronald J., and Hector J. Levesque. "The Tractability of Subsumption in Frame-Based Description Languages". *Proceedings AAAI-84,* August 1984, pp. 34-37.

[Brachman *et al*., 1985] Brachman, Ronald J., Victoria Pigman Gilbert, and Hector J. Levesque. "An Essential Hybrid Reasoning System". *Proceedings IJCAI-85,* August 1985.

[Frisch and Allen, 1982] Frisch, Alan M., and James F. Allen. "Knowledge Retrieval as Limited Inference". In *Proceedings 6th International Conference on Automated Deduction,* ed. D. W. Loveland. New York: Springer-Verlag, 1982.

[Frisch, 1085] Frisch, Alan M. "Using Model Theory to Specify AI Programs". *Proceedings IJCAI-85,* August 1985.

[Konolige, 1085] Konolige, Kurt. "Belief and Incompleteness". In *Formal Theories of the Commonsense World,* ed. Jerry R. Hobbs and Robert C. Moore. Ablex, 1985.

[Levesque, 1084a] Levesque, Hector J. "A Fundamental Tradeoff in Knowledge Representation and Reasoning". *Proceedings CSCSI/SCEIO 1984,* May 1984, pp. 141-152.

[Levesque, 1084b] Levesque, Hector J. "A Logic of Implicit and Explicit Belief". *Proceedings AAAI-84,* August 1984, pp. 198-202.

[McDermott, 1078] McDermott, Drew V. "Tarskian Semantics, or No Notation Without Denotation!" *Cognitive Science* 2, 3 (July-September 1078), pp. 277-282.

[Patel-Schneider, 1985] Patel-Schneider, Peter F. "A Decidable Logic for Knowledge Representation". AI Technical Report Number 45, Schlumberger Palo Alto Research, 1085.

[Shapiro and Wand, 1076] Shapiro, Stuart C, and M. Wand. "The Relevance of Relevance". Technical Report No. 46, Computer Science Department, Indiana University, 1076.

[Stickel, 1085] Stickel, Mark E. "Automated Deduction by Theory Resolution". *Journal of Automated Reasoning,* (1085).