Evaluation-Based Reasoning with Disjunctive Information in First-Order Knowledge Bases

Gerhard Lakemeyer
Dept. of Computer Science
RWTH Aachen
52056 Aachen
Germany
gerhard@cs.rwth-aachen.de

Hector J. Levesque
Dept. of Computer Science
University of Toronto
Toronto, Ontario
Canada M5S 3A6
hector@cs.toronto.edu

Abstract

In previous work, Levesque proposed an evaluation-based reasoning procedure for so-called *proper* KBs, equivalent to a possibly incomplete possibly infinite set of function-free ground literals. The procedure, called V, preserved the efficiency and logical soundness of database query evaluation. Moreover, if the query was constrained to be in a special normal form, V was also logically complete. In this paper, we propose an extension to this work to handle disjunctive information in a KB. We define a query evaluation procedure X that generalizes V to deal with KBs that are equivalent to a possibly infinite set of function-free ground *clauses*. Deductive reasoning that is logically sound and complete in this case is undecidable in general, and so, we describe what we are willing to give up in X to preserve tractability.

1 Introduction

It was argued in [8] that the only procedure yet proposed to perform deductive reasoning effectively over extremely large first-order knowledge bases (KBs) (involving say 10^5 or more facts) was *query evaluation* over databases. A database, however, is a very special sort of KB, namely one that is equivalent (under certain assumptions) to a maximally consistent set of function-free ground literals. No incomplete knowledge is allowed. In [8], an evaluation-based reasoning procedure was proposed for proper KBs, equivalent to a possibly *incomplete* set of function-free ground literals. The procedure, called V, preserved the efficiency and logical soundness of database query evaluation. Moreover, if the query was constrained to be in a special normal form, V was also logically complete.

However, these results do not allow any form of disjunctive knowledge in the KB. There can be a very large table for the predicate *GradStudent*, for instance, including both positive and negative instances, and even leaving the predicate open for certain individuals. But the KB cannot include anything like

$$(\,\mathit{GradStudent}(\mathit{john}) \lor \mathit{GradStudent}(\mathit{mary})\,)$$

or even

$$\forall x(\,\mathit{GradStudent}(x) \supset \mathit{Student}(x)\,)$$

as a simple way of duplicating the table for the *Student* predicate. In this paper, we propose an approach to handling disjunctive information in a KB. We define a query evaluation procedure X that generalizes V to deal with KBs that are equivalent to a possibly infinite set of function-free ground *clauses*. Deductive reasoning that is logically sound and complete in this case is undecidable in general, and so, we describe what we are willing to give up in X to preserve tractability.

The rest of the paper is organized as follows. In the next section, we review the evaluation-based reasoning procedure V, and prove a new result, which is that in the case of proper KBs, V actually computes a well-known limited form of reasoning called *tautological entailment*. In Section 3, we discuss the difficulty of generalizing V to KBs with disjunction, and argue that tautological entailment does too little in some ways and too much in others. In Section 4, we define X, an evaluation-based reasoning procedure that deals with disjunction in a novel, more appropriate way. In Section 5, we show that X agrees with V and tautological entailment on proper KBs, that it performs logically sound reasoning, and that it is computable. Finally in Section 6, we discuss future work.

2 Proper Knowledge Bases

We start with a standard first-order language $\mathcal{L}$ with no function symbols other than constants and a distinguished equality predicate. We assume a countably infinite set of constants $\mathcal{C} = \{c_1, c_2, \ldots\}$ for which we will be making a unique-name assumption.

Notation: As usual, elements of $\mathcal{L}$ are called formulas and formulas without free variables are called sentences. We only consider the logical connectives $\neg$, $\vee$, and $\exists$ as part of the language, but we will freely use $\wedge$, $\forall$, and $\supset$ as the usual abbreviations. We will let l range over literals and we will write $\overline{l}$ to denote its complement. Clauses are, as usual, disjunctions of literals and we will sometimes write them using set notation. We will use θ to range over substitutions of all variables by constants, and write $\alpha\theta$ as the result of applying the substitutions to α. We will write α^x_d to denote α with all free occurrences of x replaced by d. We will let ρ range over atoms whose arguments are distinct variables, so that $\rho\theta$ ranges over ground atoms.[1] We will use $\forall\alpha$ to mean the universal closure of α. Finally, we will use e to range over *ewffs*, by which we mean quantifier-free formulas whose only predicate is equality.

V and later also X make the assumption that quantification can be understood substitutionally with respect to $\mathcal{C}$. This assumption corresponds to restricting one's attention to the following kind of interpretations:

Definition 1: A <u>*standard*</u> interpretation of $\mathcal{L}$ is one where $=$ is interpreted as identity, and the denotation relation between $\mathcal{C}$ and the domain of discourse is bijective.

As the following definition and theorem from [8] show, this restriction can be captured precisely using a set of axioms about equality, provided we consider logical theories which do not mention infinitely many constants, which is something finite knowledge bases always satisfy.[2]

Definition 2: The set $\mathcal{E}$ is the axioms of equality (equivalence relation, substitution of equals for equals) and the (infinite) set of formulas $\{(c_i \neq c_j) \mid i \neq j\}$.

Theorem 1: *(from [8])*
Suppose S is any set of closed formulas, and that that there is an infinite set of constants that do not appear in S. Then $\mathcal{E} \cup S$ is satisfiable iff it has a standard model.

Knowledge bases considered by V have the following form:

Definition 3: A set *KB* of formulas is called <u>*proper*</u> if $\mathcal{E} \cup$ *KB* is consistent and *KB* is a finite set of formulas of the form $\forall(e \supset \rho)$ or $\forall(e \supset \neg\rho)$.

[1] Because equality is treated separately, "atoms" and "literals" here do not include equalities.

[2] See also [9] where logics are considered based only on standard interpretations.

Note that restricting ρ to be an atom with only variables as arguments is no restriction. For example, if d is a constant then $\neg P(d)$ can be rewritten as $\forall x.(x = d) \supset \neg P(x)$.

Given a proper knowledge base *KB* and a query, V returns one of three values 0, 1, or $\frac{1}{2}$, where 0 means "known to be false," 1 means "known to be true," and $\frac{1}{2}$ means "unknown." Then we have

1. $V[\rho\theta] = \begin{cases} 1 & \text{if there is a } \forall(e \supset \rho) \in KB \\ & \text{such that } V[e\theta] = 1 \\ 0 & \text{if there is a } \forall(e \supset \neg\rho) \in KB \\ & \text{such that } V[e\theta] = 1 \\ \frac{1}{2} & \text{otherwise} \end{cases}$
2. $V[t = t'] = 1$ if t is identical to t', and 0 otherwise.
3. $V[\neg\alpha] = 1 - V[\alpha]$.
4. $V[\alpha \vee \beta] = \max\{V[\alpha], V[\beta]\}$.
5. $V[\exists x.\alpha] = \max_{d \in H_1^+}\{V[\alpha^x_d]\}$

Here, H_k^+ is the union of the constants in *KB*, those mentioned in the query α, and k new constants appearing nowhere in *KB* and α.

The notions of soundness and completeness for evaluation methods like V was introduced in [8] as follows:

Definition 4: Let $S, T \subseteq \mathcal{L}$, and let $f \in [\mathcal{L} \to \{0, 1, \frac{1}{2}\}]$. Then

- f is logically <u>*sound*</u> wrt S for T iff for every $\alpha \in T$, if $f[\alpha] = 1$ then $S \models \alpha$, and if $f[\alpha] = 0$ then $S \models \neg\alpha$;
- f is logically <u>*complete*</u> wrt S for T iff for every $\alpha \in T$, if $S \models \alpha$ then $f[\alpha] = 1$, and if $S \models \neg\alpha$ then $f[\alpha] = 0$.

The proof of the soundness of V relies on the following theorem, which was proved in [8]:

Theorem 2: *(from [8])*
Let S be a set of closed formulas, let α be a formula with a single free variable x. Then for every constant $d \in \mathcal{C}$, there is a constant $c \in H_1^+(S \cup \{\alpha\})$ such that $\mathcal{E} \cup S \models \alpha^x_d$ iff $\mathcal{E} \cup S \models \alpha^x_c$.

Since V is decidable, it clearly is not complete for arbitrary queries. However, Levesque showed completeness for queries in a certain normal form called $\mathcal{NF}$. In the propositional case, for example, conjunctions of the non-tautologous prime implicates of a sentence are in $\mathcal{NF}$. In the first-order case, the situation is more complex. For the purposes of this paper, it suffices to note that any negation-free sentence is in $\mathcal{NF}$.

Interestingly, when we replace logical entailment by a variant of *tautological entailment*, a fragment of relevance logic [2], we obtain that V is complete for *arbitrary* queries. To show this, we first define tautological entailment semantically.

For simplicity, we define the semantics of tautological entailment only for standard models and, in addition, keep the semantics of equality classical. (It can easily be shown that our variant of tautological entailment coincides with the original one for sentences without equality.)

Let us call a set s of ground literals (not necessarily consistent) a <u>setup</u>. Then the support relation $\models^t$ between setups and sentences is defined as follows:

1. $s \models^t l$ iff $l \in s$ for any literal l;
2. $s \models^t (t = t')$ iff t is identical to t';
3. $s \models^t \neg(t_1 = t_2)$ iff t is not identical to t';
4. $s \models^t \neg\neg\alpha$ iff $s \models^t \alpha$;
5. $s \models^t (\alpha \vee \beta)$ iff $s \models^t \alpha$ or $s \models^t \beta$;
6. $s \models^t \neg(\alpha \vee \beta)$ iff $s \models^t \neg\alpha$ and $s \models^t \neg\beta$;
7. $s \models^t \exists x.\alpha$ iff $s \models^t \alpha^x_d$ for some $d \in \mathcal{C}$;
8. $s \models^t \neg\exists x.\alpha$ iff $s \models^t \neg\alpha^x_d$ for all $d \in \mathcal{C}$;

Note that because setups are arbitrary sets of literals and because there are separate rules of interpretation for sentences and their negations, the truth and falsity of a sentence may receive independent support.[3]

Definition 5: A set of sentences S <u>tautologically entails</u> a sentence α ($S \longrightarrow \alpha$) iff for all setups s, if $s \models^t \gamma$ for all $\gamma \in S$, then $s \models^t \alpha$.

By breaking the connection between the truth and the falsity of a sentence, tautological entailment becomes strictly weaker than logical entailment. For example, $p \wedge (p \supset q)$ does not tautologically entail q because there are setups which support both p and $\neg p$ and, hence, $p \wedge (p \supset q)$, yet do not support q. It turns out that this weakening is just enough to make V complete for tautological entailment and arbitrary queries.

To prove the theorem we need the following definition and lemmas.

Definition 6: For a proper *KB*, let
<u>*Lits(KB)*</u> $= \{l\theta \mid \forall(e \supset l) \in$ *KB* and $\mathcal{E} \models e\theta\}$.

It is easy to see that any setup which satisfies *KB* is a superset of *Lits(KB)*. In other words, *Lits(KB)* is the minimal setup which satisfies *KB*.

[3] Originally, setups were defined using the four truth values *true*, *false*, *neither*, and *both* assigned to atoms [4]. Defining them, as we do, in terms of sets of literals is equivalent.

Lemma 3: *For every setup s such that $s \models^t$ KB, if Lits(KB) $\models^t \alpha$ then $s \models^t \alpha$*

Proof: The proof is by a simple induction on the structure of α using the fact that any setup which satisfies *KB* is a superset of *Lits(KB)*. ∎

Lemma 4: *KB* $\longrightarrow \alpha$ *iff Lits(KB)* $\models^t \alpha$.

Proof: Let *KB* $\longrightarrow \alpha$. Since *Lits(KB)* $\models^t$ *KB*, *Lits(KB)* $\models^t \alpha$ follows. Conversely, let *Lits(KB)* $\models^t \alpha$. Then by Lemma 3 $s \models^t \alpha$ for any s such that $s \models^t$ *KB*. ∎

Lemma 5: *KB* $\longrightarrow \exists x.\alpha$ *iff KB* $\longrightarrow \alpha^x_c$ *for some* $c \in H^+_1$.

The proof is almost identical to that of of Theorem 2, which can be found in [8]. See also the proof of Theorem 14 in this paper, where a similar argument is used.

Theorem 6: *Let KB be proper and α an arbitrary sentence. Then*
$V[\alpha] = 1$ *iff KB* $\longrightarrow \alpha$ *and* $V[\alpha] = 0$ *iff KB* $\longrightarrow \neg\alpha$.

Proof: The proof is by induction on the structure of α.

For closed atomic formulas $\rho\theta$ we have $V[\rho\theta] = 1$ iff there is a $\forall(e \supset \rho) \in$ *KB* such that $V[e\theta] = 1$ iff $\rho\theta \in$ *Lits(KB)* (since$V[e\theta] = 1$ iff $\mathcal{E} \models e\theta$ by Lemma 7 of [8]) iff *Lits(KB)* $\models^t \rho\theta$ iff *KB* $\longrightarrow \rho\theta$ by Lemma 4. The case for $V[\rho\theta] = 0$ is similar.

$V[t = t'] = 1$ iff t and t' are identical iff $s \models^t (t = t')$ for all setups s iff *KB* $\longrightarrow (t = t')$, and similarly for the case $V[t = t'] = 0$.

$V[\neg\alpha] = 1$ iff $V[\alpha] = 0$ iff (by induction) *KB* $\longrightarrow \neg\alpha$, and similarly for $V[\neg\alpha] = 0$.

$V[\alpha \vee \beta] = 1$ iff $V[\alpha] = 1$ or $V[\beta] = 1$ iff (by induction and Lemma 4) *Lits(KB)* $\models^t \alpha$ or *Lits(KB)* $\models^t \beta$ iff *Lits(KB)* $\models^t (\alpha \vee \beta)$ iff *KB* $\longrightarrow (\alpha \vee \beta)$, again by Lemma 4. The case $V[\alpha \vee \beta] = 0$ is proved analogously.

$V[\exists x.\alpha] = 1$ iff $V[\alpha^x_c] = 1$ for some $c \in H^+_1$ iff (by induction) *KB* $\longrightarrow \alpha^x_c$ iff (Lemma 4) *Lits(KB)* $\models^t \alpha^x_c$ iff *Lits(KB)* $\models^t \exists x.\alpha$ (by Lemma 5) iff *KB* $\longrightarrow \exists x.\alpha$. Similarly for $V[\exists x.\alpha] = 0$. ∎

This theorem shows that for proper KBs, V coincides with tautological entailment. It therefore follows that for proper KBs, tautological entailment is decidable and moreover agrees with logical entailment on queries in $\mathcal{NF}$.

3 Including clauses in a KB

In the previous section, we saw that it was possible to efficiently perform logically sound first-order reasoning using an evaluation-based procedure if the KB in question was proper. Furthermore, the reasoning was logically complete if the query was in a normal form, $\mathcal{NF}$. But a proper KB is one that is equivalent to a (possibly infinite) set of ground *literals*. What we now want to consider is this: is there an evaluation-based reasoning procedure that works for KBs that are equivalent to a (possibly infinite) set of ground *clauses*? We will argue that the answer is *yes*.

3.1 Negation is not the problem

Of course, the entailment problem we are contemplating will be undecidable in general, just as it was for proper KBs. However, unlike with proper KBs, the problem remains undecidable even for queries in $\mathcal{NF}$:

Theorem 7: *The question as to whether KB $\models \alpha$ is undecidable, even if α is negation-free and KB is of the form* $\{\forall(P_1(\vec{x}_1) \vee Q_1(\vec{x}_1)), \ldots, \forall(P_n(\vec{x}_n) \vee Q_n(\vec{x}_n))\}$.

Proof: The theorem is a special case of the negation-free logic considered in [3]. A formula α is said to be in *negation normal form (NNF)* if $\neg$ appears only in front of atomic formulas. The idea is to reduce validity or logical implication for formulas in NNF to logical implication involving only negation-free formulas. Hence let α be a formula in NNF without $=$ and let $P_1, \ldots P_n$ be the predicate symbols mentioned in α. Let $Q_1, \ldots, Q_n$ be predicate symbols not occurring in α where Q_i and P_i have the same arity. Let α' be α with all occurrences of $\neg P_i$ replaced by Q_i. Note that α' is negation-free. Then it was shown in [3] that α is valid iff $\{\forall(P_1(\vec{x}_1) \vee Q_1(\vec{x}_1)), \ldots, \forall(P_n(\vec{x}_n) \vee Q_n(\vec{x}_n))\} \models \alpha' \vee \bigvee_i(\exists\vec{x}_i.P_i(\vec{x}_i) \wedge Q_(\vec{x}_i)$). Since the validity problem for formulas in NNF is undecidable, the theorem follows. ∎

Since the query in this theorem is negation-free, it is guaranteed to be in $\mathcal{NF}$. There is also a propositional version of the theorem:

Theorem 8: *The question as to whether KB $\models \alpha$ is co-NP hard, even if KB is of the form $\{(p_1 \vee q_1), \ldots, (p_n \vee q_n)\}$ and α is propositional and negation-free.*

Proof: The proof uses a special case of the reduction of the previous proof. co-NP hardness follows from the fact that logical implication is co-NP hard in the propositional case. ∎

So the problem with including clauses in the KB is not due to *negation* and so will not be resolved by restricting the queries to be in $\mathcal{NF}$. Note, for example, that in both cases the KB is already closed under Resolution. This means that any attempt to limit reasoning by changing how negation will be understood will not work here. As noted above, this is what tautological entailment does: we fail to conclude q from p and $(\neg p \vee q)$ precisely because we admit setups that support both p and its negation. Without negation, tautological entailment and classical entailment coincide:

Theorem 9: *Suppose that KB is a set of formulas $\forall(e \supset c)$ where each c is negation-free, and α is negation-free. Then*

$$\mathcal{E} \cup KB \models \alpha \quad \textit{iff} \quad KB \longrightarrow \alpha$$

Proof: To prove the only-if direction, let $\mathcal{E} \cup \text{KB} \models \alpha$ and suppose $s \models^t KB$. Let s^- be s with all negative literals removed. It is easy to prove by induction that s and s^- agree on all negation-free sentences. Hence, in particular, $s^- \models^t KB$. Now let M be a standard model such that for all closed atoms $\rho\theta$, $M \models \rho\theta$ iff $\rho\theta \in s^-$. Again, it is easy to prove by induction that $M \models \gamma$ iff $s^- \models^t \gamma$ for all negation-free sentences γ. Then $M \models KB$ and hence, by assumption, $M \models \alpha$ (standard models clearly satisfy $\mathcal{E}$). Since α is negation-free, we also have $s^- \models^t \alpha$ from which $s \models^t \alpha$ follows.

Conversely, let $KB \longrightarrow \alpha$ and let $M \models \mathcal{E} \cup KB$. It was shown in [8] (proof of Theorem 2) that there is a standard model M' such that M and M' agree on all sentences mentioning only constants in *KB* and α. Thus, in particular, $M' \models KB$. Now let s be the setup such that $l \in s$ iff $M' \models l$ for all closed literals l. A simple induction proves that for all sentences γ, $M' \models \gamma$ iff $s \models^t \gamma$. Hence $s \models^t KB$ and thus, by assumption, $s \models^t \alpha$. Then $M' \models \alpha$ and, since M' and M agree on sentences mentioning only constants in *KB* and α, $M \models \alpha$. ∎

So in a sense, tautological entailment does not do enough (it fails to do simple *Modus Ponens* even over Horn clauses), and in another sense, it tries to do far too much (it is subject to the intractability theorems above). It is significant that previous work proposing a limited form of reasoning based on tautological entailment for a KB with disjunction [7, 5] only worked in the propositional case and when the query was in CNF. The first-order case later studied in [10] and [6] required considerable machinery beyond tautological entailment. Moreover, this additional effort only resulted in fewer inferences compared to tautological entailment and hence has only limited appeal.

3.2 Doing more and less than tautological entailment

In the next section, we will propose a new evaluation-based reasoning procedure X that works with disjunctions in the KB, agrees with V when the KB is proper, is reasonably efficient, logically sound, and sometimes even logically complete. It is based on the observation that although disjunction can be used in many ways, it has two major applications: (1) to represent *rules* such as Horn clauses, where we may need to perform chaining in the reasoning; and (2), to represent *incomplete knowledge* about some individual(s), where we may need to split cases in the reasoning. As argued above, tautological entailment does not do (1) at all, but like classical logic, attempts to handle (2).

As it turns out, we believe that (2) is the computational problem. To see why, consider the following example KBs:

KB1	**KB2**
$(P(a) \vee P(e) \vee P(f))$	$(P(a) \vee Q(e) \vee Q(c))$
$(P(a) \vee P(e) \vee Q(f))$	$(Q(d) \vee P(b) \vee Q(a))$
$(P(a) \vee Q(e) \vee P(c))$	$(P(a) \vee P(e) \vee P(f))$
$(P(a) \vee Q(e) \vee Q(c))$	$(P(c) \vee Q(e) \vee P(a))$
$(Q(a) \vee P(b) \vee P(d))$	$(Q(a) \vee Q(b) \vee Q(g))$
$(Q(a) \vee P(b) \vee Q(c))$	$(P(a) \vee P(e) \vee Q(f))$
$(Q(a) \vee Q(b) \vee P(g))$	$(Q(b) \vee Q(a) \vee P(g))$
$(Q(a) \vee Q(b) \vee Q(g))$	$(Q(a) \vee P(d) \vee P(b))$

The reader is invited to confirm that one and only one of these logically entails (and hence tautologically entails) $\exists x.(P(x) \wedge Q(x))$. So being required to handle (2) is also being required to solve combinatorial puzzles like this automatically.

Our approach will be to preserve (1), but to give up on (2). To deal with (2) in a limited way, X will split disjunctions and reason by cases, but the number of such splits will be limited by the structure of the query. For example, if we have the clause $(P(a) \vee P(b))$ in the KB, X will infer $(P(a) \vee P(b) \vee P(c))$ and $\exists x.P(x)$ since both of these queries will allow us to split the disjunction. However, if we also have $(\neg P(a) \vee P(b))$ in the KB, then X may be unable to infer $P(b)$, since this query will not allow us to split a disjunction. This restriction will force us into logically incomplete reasoning since, among other things, X will be unable to solve the puzzle above. This is a feature, not a bug.

4 Retrieval with clauses

We begin by considering X_0, a simplified version of X, where the knowledge base is restricted to a (possibly infinite) set of ground clauses. This has the advantage that the basic principles underlying X can be illustrated in a conceptually simpler setting.

Different from V, X_0 returns only two values 0 or 1 where 1 means "known to be true," as before, but 0 now means "not known to be true" instead of "known to be false" in the case of V. As we will see, this change is necessary because, similar to tautological entailment, we want to allow inconsistent knowledge bases. In particular, this means that X_0 should sometimes return 1 for both α and $\neg\alpha$, which is incompatible with the 3-valued answers of V. Note, however, that we are not really losing anything in terms of expressiveness compared to V. While for V a single query α suffices to establish that α is known to be true or known to be false, we now need to pose two queries, α and $\neg\alpha$, to have the same effect.

Definition 7: Let S be a set of ground clauses. Then UP(S) is the least set which contains S and if $\{l\} \cup c$ and $\bar{l}$ are in UP(S), then so is c. In other words, UP(S) is the closure of S under *unit propagation*.

Let S be a non-empty, not necessarily consistent and not necessarily finite, set of ground clauses. Then X_0 is defined as follows:

1. $X_0[S,l] = \begin{cases} 1 & \text{if } l \in \text{UP}(S),\ l \text{ a literal} \\ 0 & \text{otherwise} \end{cases}$
2. $X_0[S, t = t'] = 1$ if t is identical to t', and 0 otherwise
3. $X_0[S, \neg(t = t')] = 1 - X_0[S, t = t']$
4. $X_0[S, \neg\neg\alpha] = X_0[S, \alpha]$
5. $X_0[S, \alpha \vee \beta] =$ $\begin{cases} 1 & \text{if for some } c \in S \text{ and for all } l \in c, \\ & X_0[S \cup \{l\}, \alpha] = 1 \text{ or } X_0[S \cup \{l\}, \beta] = 1, \\ 0 & \text{otherwise} \end{cases}$
6. $X_0[S, \neg(\alpha \vee \beta)] = \min\{X_0[S, \neg\alpha], X_0[S, \neg\beta]\}$
7. $X_0[S, \exists x.\alpha] =$ $\begin{cases} 1 & \text{if for some } c \in S \text{ and for all } l \in c, \\ & \text{there is a } d \in \mathcal{C} \text{ such that } X_0[S \cup \{l\}, \alpha^x_d] = 1 \\ 0 & \text{otherwise} \end{cases}$
8. $X_0[S, \neg\exists x.\alpha] = \min_{d \in \mathcal{C}}\{X_0[S, \neg\alpha^x_d]\}$

Besides the fact that the definition of X_0 is somewhat more long-winded than V, the main differences lie in the treatment of literals, disjunctions, and existentials.

When it comes to establishing whether a literal can be inferred, the use of UP(S) ensures simple applications of *Modus Ponens*. For example, if S contains p, $\neg s$, $(\neg p \vee q)$, and $(\neg q \vee r \vee s)$, then $X_0[S, r] = 1$.

Whenever a disjunction is encountered in the query, Rule 5 allows reasoning by cases, but only with respect to a single clause in S. The same is true for existential quantifiers using Rule 7.

To get a better feel for what can and cannot be inferred using X_0, let us consider the following examples. Suppose

$$S = \{(P(a) \vee Q(b)), (\neg P(a) \vee Q(b))\}$$

Then $X_0[S, \exists x.Q(x)] = 1$ because the existential quantifier in the query allows us to split a clause. If we choose the first clause, we obtain $X_0[S \cup \{P(a)\}, Q(b)] = 1$ because $Q(b) \in$ UP$(S \cup \{P(a)\})$ and $X_0[S \cup \{Q(b)\}, Q(b)] = 1$, from which the existential follows. On the other hand,

$X_0[S, Q(b)] = 0$ because no splitting of clauses is allowed and $Q(b)$ is not in UP(S), which is the same as S. Now let

$$S = \{(P(a) \vee P(b)), Q(a), Q(b)\}$$

and let $\alpha = \exists x.P(x) \wedge Q(x)$. Then X_0 correctly determines that α follows from S, that is, $X_0[S, \alpha] = 1$. This is because the first clause can be used to split cases when evaluating the existential.

Finally, if we let S be *KB*2 from Section 3, then $X_0[S, \alpha] = 0$ even though *KB*2 logically entails α. The reason is that it is simply not enough to reason by cases with respect to only one clause (when evaluating $\exists$) to solve the puzzle.

As noted already at the beginning of this section, X_0, in contrast to V, applies to knowledge bases which may be logically inconsistent. To see how inconsistencies are dealt with, let us consider two examples. Let

$$S_1 = \{p, q, (q \supset \neg p)\}.$$

S_1 is clearly inconsistent and $X_0[S_1, p \wedge \neg p] = 1$ since both p and $\neg p$ are in UP(S_1). Now consider

$$S_2 = \{(p \vee q), (\neg p \vee q), (p \vee \neg q), (\neg p \vee \neg q)\}.$$

While S_2 is again inconsistent, this time $X_0[S_2, p \wedge \neg p] = 0$ since UP$(S_2) = S_2$. Intuitively, X_0 will only discover shallow inconsistencies, in particular those which it can discover using unit propagation and limited forms of reasoning by cases, depending on the occurrence of disjunctions or existential quantifiers in the query. In a sense, the treatment of inconsistencies by X_0 is more powerful than tautological entailment (for example, S_1 does not tautologically entail $(p \wedge \neg p)$), yet it is still far less powerful than classical logical reasoning, which it needs to be to make it computationally viable.

We conclude our discussion of X_0 by showing that, in the case of proper knowledge bases, X_0 is upward compatible with V in that X_0 returns 1 as an answer just in case V does. The main reason for the agreement is that proper *KB*'s consist only of literals and hence there are no clauses to split.

Theorem 10: *Let KB be non-empty and proper and let Lits(KB)* $= \{l\theta \mid \forall(e \supset l) \in KB$ *and* $\mathcal{E} \models e\theta\}$.
Then for all α, $X_0[$*Lits(KB)*$, \alpha] = 1$ *iff* $V[\alpha] = 1$.

Proof: The proof is by induction on the structure of α. Let $\rho\theta$ be an atomic formula in *Lits(KB)*. Then $X_0[$*Lits(KB)*$, \rho\theta] = 1$ iff $\rho\theta \in$ *Lits(KB)* iff $V[\rho\theta] = 1$ by definition and the fact that for any ewff e and substitution θ, $V[e\theta] = 1$ iff $\mathcal{E} \models e\theta$ (Lemma 7 of [8]). Similarly, $X_0[$*Lits(KB)*$, \neg\rho\theta] = 1$ iff $\neg\rho\theta \in$ *Lits(KB)* iff $V[\rho\theta] = 0$ iff $V[\neg\rho\theta] = 1$.

The proof for $(t = t')$ and $\neg(t = t')$ follows immediately because the definitions of X_0 and V agree in this case.

$X_0[$*Lits(KB)*$, \neg\neg\alpha] = 1$ iff $X_0[$*Lits(KB)*$, \alpha] = 1$ iff (by induction) $V[\alpha] = 1$ iff $V[\neg\neg\alpha] = 1$.

Let $X_0[$*Lits(KB)*$, (\alpha \vee \beta)] = 1$. Since *Lits(KB)* consists only of literals, there is an $l \in$ *Lits(KB)* such that $X_0[$*Lits(KB)*$\cup\{l\}, \alpha] = 1$ or $X_0[$*Lits(KB)*$\cup\{l\}, \beta] = 1$. By induction and since *Lits(KB)* $=$ *Lits(KB)* $\cup \{l\}$, $V[\alpha] = 1$ or $V[\beta] = 1$, from which $V[(\alpha \vee \beta)] = 1$ follows.

Conversely, Let $V[(\alpha \vee \beta)] = 1$. Then $V[\alpha] = 1$ or $V[\beta] = 1$ and hence, by induction, $X_0[$*Lits(KB)*$, \alpha] = 1$ or $X_0[$*Lits(KB)*$, \beta] = 1$. Let l be any literal in *Lits(KB)*. Then $X_0[$*Lits(KB)* $\cup \{l\}, \alpha] = 1$ or $X_0[$*Lits(KB)* $\cup \{l\}, \beta] = 1$, from which $X_0[$*Lits(KB)*$, (\alpha \vee \beta)] = 1$ follows by definition.

Let $X_0[$*Lits(KB)*$, \neg(\alpha \vee \beta)] = 1$ iff $X_0[$*Lits(KB)*$, \neg\alpha] = 1$ and $X_0[$*Lits(KB)*$, \neg\beta] = 1$ iff (by induction) $V[\alpha] = 0$ and $V[\beta] = 0$ iff $V[\neg\alpha] = 1$ and $V[\neg\beta] = 1$ iff $V[\neg(\alpha \vee \beta)] = 1$.

Let $X_0[$*Lits(KB)*$, \exists x.\alpha] = 1$. Then for some literal $l \in$ *Lits(KB)* and $c \in \mathcal{C}$, $X_0[$*Lits(KB)*$\cup\{l\}, \alpha^x_d] = 1$. By induction and since *Lits(KB)* $=$ *Lits(KB)* $\cup \{l\}$, $V[\alpha^x_d] = 1$. By Theorem 9 of [8], there is a $d' \in H_1^+$ such that $V[\alpha^x_{d'}] = 1$. Therefore, $V[\exists x.\alpha] = 1$.

Conversely, let $V[\exists x.\alpha] = 1$. Then for some $d \in H_1^+$, $V[\alpha^x_d] = 1$ and, by induction, $X_0[$*Lits(KB)*$, \alpha^x_d] = 1$. Then for any literal l, $X_0[$*Lits(KB)* $\cup \{l\}, \alpha^x_d] = 1$, from which $X_0[$*Lits(KB)*$, \exists x.\alpha] = 1$ follows.

The case $\neg\exists x.\alpha$ follows easily by induction. ∎

While X_0 is clearly limited in what it does compared to logical entailment, more needs to be done to ensure its practicality. For one, we want to allow more than just ground clauses in the *KB*. For another, Rules 5, 7, and 8 appeal to substitutions over an infinite domain, which needs to be constrained.

To go beyond ground clauses, we first generalize the notion of a proper *KB* introduced in Section 2. The idea is that instead of universally quantified literals, possibly restricted by ewffs, we now allow arbitrary clauses to take the place of the literals.

Definition 8: Let e be an ewff and c a disjunction of literals whose arguments are distinct variables. Then $\forall(e \supset c)$ is called a <u>$\forall$-clause</u>.

Definition 9: A *KB* is called <u>*proper*$^+$</u> if *KB* is a finite non-empty collection of $\forall$-clauses. Given a proper$^+$ *KB*, <u>*gnd(KB)*</u> is defined as $\{c\theta \mid \forall(e \supset c) \in KB$ and $\mathcal{E} \models e\theta\}$.

Note that, under the standard interpretations introduced in Section 2, a proper$^+$ *KB* is a finite representation of the (usually infinite) set of ground clauses gnd(*KB*). In particular, a standard interpretation satisfies *KB* iff it satisfies gnd(*KB*). We are now ready to define X for proper$^+$ *KB*'s and arbitrary α.

1. $X[KB, l] = \begin{cases} 1 & \text{if } l \in \text{UP}(\text{gnd}(KB)),\ l \text{ a literal} \\ 0 & \text{otherwise} \end{cases}$
2. $X[KB, t = t'] = 1$ if t is identical to t', and 0 o.w.
3. $X[KB, \neg(t = t')] = 1 - X[KB, t = t']$
4. $X[KB, \neg\neg\alpha] = X[KB, \alpha]$
5. $X[KB, \alpha \vee \beta] =$
$\begin{cases} 1 & \text{there is a } \forall(e \supset c) \in KB \text{ and a } \theta \in H_k^+ \text{ such that}^4 \\ & X[KB, e\theta] = 1 \text{ and for all } l \in c, \\ & X[KB \cup \{l\theta\}, \alpha] = 1 \text{ or } X[KB \cup \{l\theta\}, \beta] = 1, \\ & \text{where } k \text{ is the number of free variables in } c \\ 0 & \text{otherwise} \end{cases}$
6. $X[KB, \neg(\alpha \vee \beta)] = \min\{X[KB, \neg\alpha], X[KB, \neg\beta]\}$
7. $X[KB, \exists x.\alpha] =$
$\begin{cases} 1 & \text{if there is a } \forall(e \supset c) \in KB \text{ and a } \theta \in H_k^+ \text{ such that} \\ & X[KB, e\theta] = 1 \text{ and for all } l \in c \\ & \text{there is a } d \in H_{k+1}^+ \text{ such that } X[KB \cup \{l\theta\}, \alpha_d^x] = 1, \\ & \text{where } k \text{ is as above.} \\ 0 & \text{otherwise} \end{cases}$
8. $X[KB, \neg\exists x.\alpha] = \min_{d \in H_1^+}\{X[KB, \neg\alpha_d^x]\}$

Using X with a *KB* is very similar to using X_0 with gnd(*KB*). The main difference is the way disjunctions and existentials are handled. While in X_0 we allow any clause from S to be chosen for case splitting, in X we can only split substitution instances over H_k^+ of a $\forall$-clause of *KB*. Also, X_0 evaluates negated existentials (universals) with respect to a finite set of constants, just like V does. Nevertheless, this is done without loss of generality in that X and X_0 agree on their answers. To prove this we need the following lemmas.

Let $*$ be any bijection from $\mathcal{C}$ to $\mathcal{C}$. Let α^* mean α with c replaced by c^*. Let S^* mean $\{\alpha^* \mid \alpha \in S\}$. Let θ^* be the substitution which assigns variable x the value c^* if θ assigns x the value c.

Lemma 11: $X_0[S, \alpha] = X_0[S^*, \alpha^*]$.

Lemma 12: $c \in UP(S)$ *iff* $c^* \in UP(S^*)$.

The lemmas are proved by simple induction arguments over the structure of α and the length of a derivation by unit propagation, respectively.

[4] Here, $\theta \in H_k^+$ means that the substitutions range only over the elements in H_k^+.

Lemma 13: $gnd(KB)^* = gnd(KB^*)$.

Proof: Let $c' \in \text{gnd}(KB)^*$. Then $c' = c^*\theta^*$ for some c and θ such that $c\theta \in \text{gnd}(KB)$ with $\forall(e \supset c) \in KB$ and $\mathcal{E} \models e\theta$. A simple induction shows that for any ewff e, $\mathcal{E} \models e$ iff $\mathcal{E} \models e^*$. Hence $\mathcal{E} \models e^*\theta^*$. Since $\forall(e^* \supset c^*) \in KB^*$, we have that $c^*\theta^* \in \text{gnd}(KB^*)$, that is, $c' \in \text{gnd}(KB^*)$. The converse is proved analogously. ∎

Theorem 14: *For any proper$^+$ KB and any sentence α,* $X[KB, \alpha] = 1$ *iff* $X_0[gnd(KB), \alpha] = 1$.

Proof: Note that X and X_0 already agree on all rules except 5, 7, and 8. We show that these rules can also be made identical in the case of proper$^+$ *KB*s. The theorem then follows by a simple induction argument.

It suffices to show the following:

1. $X_0[\text{gnd}(KB), \neg\exists x.\alpha] = \min_{d \in H_1^+}\{X_0[\text{gnd}(KB), \neg\alpha_d^x]\}$

2. If $X_0[\text{gnd}(KB), (\alpha \vee \beta)] = 1$ then for some $\forall(e \supset c) \in KB$ and $\theta \in H_k^+$ s.t. $X_0[\text{gnd}(KB), e\theta] = 1$ and for all $l \in c$, $X_0[\text{gnd}(KB) \cup \{l\theta\}, \alpha] = 1$ or $X_0[\text{gnd}(KB) \cup \{l\theta\}, \beta] = 1$.

3. If $X_0[\text{gnd}(KB), \exists x.\alpha] = 1$ then for some $\forall(e \supset c) \in KB$ and $\theta \in H_k^+$ s.t. $X_0[\text{gnd}(KB), e\theta] = 1$ and for all $l \in c$ there is a $d \in H_{k+1}^+$ such that $X_0[\text{gnd}(KB) \cup \{l\theta\}, \alpha_d^x] = 1$.

We now prove these cases in turn.

1. Suppose $X_0[\text{gnd}(KB), \neg\alpha_d^x] = 1$ for all $d \in H_1^+$. We need to show that $X_0[\text{gnd}(KB), \neg\alpha_d^x] = 1$ for all $d \in \mathcal{C}$. Let $d \notin H_1^+$ and let d' be the constant in H_1^+ which does not appear in $KB \cup \{\alpha\}$. Let $*$ be a bijection which swaps d and d' and is the identity otherwise. Since $X_0[\text{gnd}(KB), \neg\alpha_{d'}^x] = 1$ by assumption, $X_0[\text{gnd}(KB)^*, \neg\alpha_{d'}^{x\ *}] = 1$ by Lemma 11. Since, by Lemma 13, $\text{gnd}(KB)^* = \text{gnd}(KB^*)$ and $KB^* = KB$, $\alpha^* = \alpha$, and $d = d'^*$ by the definition of $*$, $X_0[\text{gnd}(KB), \neg\alpha_d^x] = 1$ follows.

2. Let $X_0[\text{gnd}(KB), (\alpha \vee \beta)] = 1$. By definition, there is a $c\theta \in \text{gnd}(KB)$ s.t. $\forall(e \supset c) \in KB$ and $\mathcal{E} \models e\theta$ and for all $l \in c$, $X_0[\text{gnd}(KB) \cup \{l\theta\}, \alpha] = 1$ or $X_0[\text{gnd}(KB) \cup \{l\theta\}, \beta] = 1$. Suppose c contains k variables. Then $c\theta$ mentions at most k names not in H_k^+. Let these be $d_1, \ldots, d_m$ with $m \leq k$. Let $*$ be the bijection which is the identity everywhere except that it swaps d_i with d_i' where the d_i' are distinct elements of H_k^+ which do not appear in $KB \cup \{(\alpha \vee \beta)\}$. (These must exist given the definition of H_k^+.) Given the choice of $*$ and since the values which θ assigns to variables not occurring in c can be chosen arbitrarily, we can assume, without loss of generality, that $\theta^* \in H_k^+$. Also, by Lemma 11, $\mathcal{E} \models e^*\theta^*$ and for all $l^* \in c^*$, $X_0[(\text{gnd}(KB) \cup \{l\theta\})^*, \alpha^*] = 1$ or $X_0[(\text{gnd}(KB) \cup \{l\theta\})^*, \beta^*] = 1$. Since $\alpha = \alpha^*$, $\beta = \beta^*$,

$c = c^*$, $e = e^*$, $\text{gnd}(KB) = \text{gnd}(KB)^*$, we have that $X_0[\text{gnd}(KB), e\theta^*] = 1$ and for all $l \in c$, $X_0[\text{gnd}(KB) \cup \{l\theta^*\}, \alpha] = 1$ or $X_0[\text{gnd}(KB) \cup \{l\theta^*\}, \beta] = 1$.

3. Let $X_0[\text{gnd}(KB), \exists x.\alpha] = 1$. Using an argument completely analogous to the previous case one can show that there is an $\forall(e \supset c) \in KB$ and a $\theta \in H_k^+$ such that $X_0[\text{gnd}(KB), e\theta] = 1$ and for all $l \in c$ there is a $c \in \mathcal{C}$ such that $X_0[\text{gnd}(KB) \cup \{l\theta\}, \alpha_d^x] = 1$. All that is left to show is that d can be chosen from H_{k+1}^+. Suppose d is not in H_{k+1}^+, then consider a bijection $*$ which is the identity everywhere except that it swaps d with d' where $d' \in H_{k+1}^+$ and does not appear in either *KB*, $l\theta$, or α. Similar to the previous case it then follows that $X_0[\text{gnd}(KB) \cup \{l\theta\}, \alpha_{d'}^x] = 1$ and we are done. ∎

5 Advantages of X

But what does X have going for it, besides its close relationship with X_0? In particular, what can we say about X as a method of logical inference? Here we will show that X always performs logically sound reasoning, that it is fully compatible with V, and that reasoning as defined by X is decidable.

Before showing that X always performs sound reasoning, we need to adapt our original definition of soundness and completeness (Definition 4) to the case of two-valued evaluation methods:

Definition 10: Let $S, T \subseteq \mathcal{L}$, and let $f \in [\mathcal{L} \rightarrow \{0, 1\}]$. Then

- f is logically sound wrt S for T iff for every $\alpha \in T$, if $f[\alpha] = 1$ then $S \models \alpha$.
- f is logically complete wrt S for T iff for every $\alpha \in T$, if $S \models \alpha$ then $f[\alpha] = 1$.

We need the following simple properties to prove soundness. Since X has already been defined for closed atomic ewffs, we obtain by a simple induction:

Lemma 15: *For any proper$^+$ KB, ewff e and substitution θ, $X[KB, e\theta] = 1$ iff $\mathcal{E} \models e\theta$.*

We also use the following fact about logical entailment:

Lemma 16: *Let S be any set of sentences, and $c = (l_1, \ldots, l_n)$ a ground clause. Then for any sentence α, $S \cup \{c\} \models \alpha$ iff $S \cup \{l_i\} \models \alpha$ for all l_i.*

Theorem 17: *X is logically sound wrt to any proper$^+$ KB for $\mathcal{L}$.*

Proof: The proof is by induction on the structure of α. Let $X[KB, l] = 1$ for a literal l. Then $l \in \text{UP}(\text{gnd}(KB))$. Thus $\mathcal{E} \cup \text{UP}(\text{gnd}(KB)) \models l$ and, since $\mathcal{E} \cup \text{UP}(\text{gnd}(KB))$ and $\mathcal{E} \cup KB$ are logically equivalent, $\mathcal{E} \cup KB \models l$. If $X[KB, t = t'] = 1$ then t and t' are identical, which is clearly sound, and similarly for $X[KB, \neg(t = t')] = 1$.

The cases $\neg\neg\alpha$ and $\neg(\alpha \vee \beta)$ follow easily by induction.

Let $X[KB, (\alpha \vee \beta)] = 1$. Then for some $\forall(e \supset c) \in KB$ and $\theta \in H_k^+$ with $X[KB, e\theta] = 1$ we have that for all $l \in c$, $X[KB \cup \{l\theta\}, \alpha] = 1$ or , $X[KB \cup \{l\theta\}, \beta] = 1$. Hence, by induction, $\mathcal{E} \cup KB \cup \{l\theta\} \models \alpha$ or $\mathcal{E} \cup KB \cup \{l\theta\} \models \beta$ for all $l \in c$. Therefore, for all $l \in c$, $\mathcal{E} \cup KB \cup \{l\theta\} \models (\alpha \vee \beta)$. By Lemma 16 we have that $\mathcal{E} \cup KB \cup \{c\theta\} \models (\alpha \vee \beta)$. Since $X[KB, e\theta] = 1$, we also have $\mathcal{E} \models e\theta$ by Lemma 15, that is, $c\theta \in \text{UP}(\text{gnd}(KB))$. Therefore, since $\mathcal{E} \cup KB \cup \{c\theta\}$ and $\mathcal{E} \cup KB$ are logically equivalent, $\mathcal{E} \cup KB \models (\alpha \vee \beta)$ follows.

The proof for $\exists x.\alpha$ is completely analogous to the previous case.

Finally, let $X[KB, \neg\exists x.\alpha] = 1$. Then $X[KB, \neg\alpha_d^x] = 1$ for all $d \in H_1^+$. Then, by induction, $\mathcal{E} \cup KB \models \neg\alpha_d^x$ for all $d \in H_1^+$. By Theorem 2, $\mathcal{E} \cup KB \models \neg\alpha_d^x$ for all $d \in \mathcal{C}$ and thus $\mathcal{E} \cup KB \models \neg\exists x.\alpha$. ∎

So if X ever returns 1, we know that the query is logically entailed by $(\mathcal{E} \cup KB)$.

Furthermore, if the *KB* is proper, that is, if all the clauses are unit clauses, then X agrees with V, which is an immediate consequence of Theorem 10 and 14.

Corollary 18: *If KB is proper then for any sentence α, $X[KB, \alpha] = 1$ iff $V[\alpha] = 1$.*

Intuitively, this is so because for a proper *KB* there are no clauses to split, and the rules of X for $\vee$ and $\exists$ essentially reduce to those of V. So in this case, X will perform logically sound and complete reasoning for queries in $\mathcal{NF}$, just like V, and just as efficiently.

Unlike V however, X does allow disjunctions in the *KB*. It will not necessarily perform logically complete reasoning over them, but many simple cases can be handled, as discussed in the examples above for X_0. As with X_0, the X procedure is able to perform chaining, but now using quantifiers and equality. For example, if a and b are distinct constants, and

$$KB = \{\, \forall(x \neq a \supset P(x)),\ \forall(x = y \supset \neg P(x) \vee Q(y)) \,\},$$

then $X[KB, Q(b)] = 1$. The X procedure also performs case splitting, including some that would have required an infinite number of splits in X_0. For example, suppose that

$$KB = \{\, \forall(P(x, a) \vee P(x, b)) \,\}.$$

Then we have that $X[KB, \forall x \exists y.P(x,y)] = 1$. This example also shows that it would not have worked to define X to first choose n clauses to split (where perhaps n is the number of connectives in the query), and then to use V on the resulting augmented *KB*.

The final desirable feature of X is its computability:

Theorem 19: *X is decidable.*

The argument is as follows: Since the *KB* and H_k^+ are both finite, there are only finitely many choices to consider in splitting cases. Moreover, and similar to V, only finitely many constants need to be considered as substitutions for an existentially quantified variable. So all that is left to show decidability is argue that for any ground literal l, it is always possible to decide if it is a member of UP(gnd(*KB*)). Because we are using unit propagation and not full Resolution, this is similar (ignoring equality) to deciding if a ground atom is entailed by a collection of quantified Horn clauses. This can be done since there are no function symbols.

6 Conclusions

While the evaluation method V considered in previous work has many advantages in that it provides an efficient, logically sound, and sometimes complete reasoning method, its perhaps greatest shortcoming is that it only applies to proper knowledge bases, that is, sets of literals. In this paper we proposed X, which is compatible with V and, in addition, allows *KB*'s to contain arbitrary clauses. The main challenge in developing X has been to constrain it in such a way that reasoning remains decidable and, at the same time, allows for interesting inferences such as limited applications of *Modus ponens.* We have seen that on both counts X is preferable over limited reasoning methods like tautological entailment.

Many future questions remain to be explored. For example, while we have results concerning the computability of X, a stronger argument needs to be made concerning its practicality. In normal cases, we would expect X to be *efficiently* computable. It is true that the procedure can split a clause for each disjunction or existential in the query, but this would be worst-case exponential only in the number of disjunctions and existentials in the query (and with some effort, we believe this can be made exponential in the depth of connectives only). As in [8] we would normally expect the query to be at most logarithmic in the size of the *KB*, and the depth of connectives to be be logarithmic in the size of the query. Finally, as suggested above, testing membership of a ground literal in UP(gnd(*KB*)) is comparable to query answering over Horn clauses with a finite Herbrand universe, that is, Datalog [1]. Nevertheless, it still needs to be shown that we can indeed deal effectively with a KB containing (say) a large number of atomic facts and a few disjunctions. Also, can we implement X in such a way that in cases where disjunction is not needed, standard database techniques (like projections, joins *etc.*) can be used?

On the logical side, we have not attempted to characterize useful cases of queries and KBs for which X is logically complete, beyond those of V: proper KBs with queries that are in normal form. We do believe that X computes precisely the entailment relation for a new *limited logic* that is interestingly different from tautological entailment. We will characterize this logic in a separate publication.

References

[1] Abiteboul, S., Hull, R., and Vianu, V., *Foundations of Databases*. Addison-Wesley, Reading, MA, 1995.

[2] Anderson, A. R. and Belnap, N. D., *Entailment, The Logic of Relevance and Necessity*. Princeton University Press, 1975.

[3] Delgrande, J. and Gupta, A., The complexity of minimum truth assignments and implication in negation-free formulae. *Annals of Mathematics and Artificial Intelligence* **18**, 1996, pp. 51–67.

[4] Dunn, J. M., Intuitive Semantics for First-Degree Entailments and Coupled Trees. *Philosophical Studies* **29**, 1976, pp. 149–168.

[5] Frisch, A., Inference without chaining. *Proceedings of IJCAI-87*, Morgan Kaufmann, San Francisco, pp. 515–519, 1987.

[6] Lakemeyer, G., *Models of belief for decidable reasoning in incomplete knowledge bases*. Ph. D. thesis, Dept. of Computer Science, University of Toronto, Toronto, Ont., 1990.

[7] Levesque, H., A logic of implicit and explicit belief. in *Proc. AAAI-84*, pp. 198–202, Austin, TX, 1984.

[8] Levesque, H. J., A completeness result for reasoning with incomplete first-order knowledge bases. *Proceeding of the 6th International Conference on Knowledge Representation and Reasoning (KR'98)*, Morgan Kaufmann, San Francisco, 1998, pp. 14–23.

[9] Levesque, H. J. and Lakemeyer, G., *The Logic of Knowledge Bases.* MIT Press, 2001.

[10] Patel-Schneider, P., A decidable first-order logic for knowledge representation. *Proceedings of IJCAI-85*, Los Angeles, CA, pp. 455–458, 1985.