Journal of Artificial Intelligence Research 31 (2008) 259-272 Submitted 08/07; published 02/08

# On the Expressiveness of Levesque's Normal Form

**Yongmei Liu** YMLIU@MAIL.SYSU.EDU.CN
*Department of Computer Science*
*Sun Yat-sen University*
*Guangzhou 510275, China*

**Gerhard Lakemeyer** GERHARD@CS.RWTH-AACHEN.DE
*Department of Computer Science*
*RWTH Aachen*
*52056 Aachen, Germany*

## Abstract

Levesque proposed a generalization of a database called a proper knowledge base (KB), which is equivalent to a possibly infinite consistent set of ground literals. In contrast to databases, proper KBs do not make the closed-world assumption and hence the entailment problem becomes undecidable. Levesque then proposed a limited but efficient inference method $V$ for proper KBs, which is sound and, when the query is in a certain normal form, also logically complete. He conjectured that for every first-order query there is an equivalent one in normal form. In this note, we show that this conjecture is false. In fact, we show that any class of formulas for which $V$ is complete must be strictly less expressive than full first-order logic. Moreover, in the propositional case it is very unlikely that a formula always has a polynomial-size normal form.

## 1. Introduction

As argued by Levesque (1998), there is only one deductive technique efficient enough to be feasible on knowledge bases (KBs) of the size seemingly required for common-sense reasoning: the deduction underlying classical database query evaluation. And yet, databases by themselves are too restricted to serve as the representational scheme for common-sense reasoning, since they require, among other things, complete knowledge of the domain. Levesque proposed a generalization of a database called a *proper* knowledge base, which is equivalent to a possibly infinite consistent set of ground literals. To illustrate what is meant by a proper KB consider the following example:

- Ann likes Bob, and Dan likes Fred.
  $Likes(ann, bob)$
  $Likes(dan, fred)$
- Ann does not like Dan.
  $\neg Likes(ann, dan)$
- Carol likes everyone.
  $\forall x. Likes(carol, x)$
- Eve does not like anyone other than Ann and herself.
  $\forall x. x \neq ann \land x \neq eve \supset \neg Likes(eve, x)$

©2008 AI Access Foundation. All rights reserved.

In contrast to databases, proper KBs do not make the closed-world assumption. For example, while $\neg Likes(eve, fred)$ follows from the above KB (if we assume unique names), neither $Likes(ann, eve)$ nor $\neg Likes(ann, eve)$ does. Sadly, even this very restricted form of incompleteness renders the entailment problem undecidable, as entailment from an empty KB reduces to validity in classical logic.

Nevertheless, given KBs like the above many queries seem easy to answer. For example, consider the formula

$$Likes(eve, carol) \supset Likes(carol, eve),$$

which follows from the KB simply because $Likes(carol, eve)$ does. In his work Levesque devised a limited but efficient inference mechanism $V$ which gets examples like these right at the expense of being incomplete on others, that is, $V$ sometimes answers "don't know" even though the query is logically entailed.

To give a flavor of how $V$ works, consider a sentence $\alpha$ in conjunctive normal form (CNF), that is $\alpha = c_1 \wedge c_2 \wedge \ldots \wedge c_n$, where each $c_i$ is a disjunction of ground literals. In order to see whether $\alpha$ follows from the KB, $V$ simply checks whether each $c_i$ contains a literal which is an instance of one of the sentences in the KB. Such an evaluation-based scheme is clearly sound but also easily seen to be incomplete. For example, $V$ would return "don't know" when given the query $Likes(ann, eve) \vee \neg Likes(ann, eve)$, as neither literal is contained in the KB.

In his paper, Levesque introduced a certain *normal form* ($\mathcal{NF}$) for sentences and proved that $V$ is logically complete for queries in $\mathcal{NF}$. In the propositional case, examples of sentences in $\mathcal{NF}$ are those in CNF which do not contain tautological clauses and which are otherwise closed under resolution. Since every propositional sentence is equivalent to a sentence in this form, it follows immediately that every propositional sentence can be converted into an equivalent one in $\mathcal{NF}$.

Levesque then conjectured that for every sentence in first-order logic there is also an equivalent one in $\mathcal{NF}$. In this note, we show that the above conjecture of Levesque is false. In fact, we can show that any class of formulas for which $V$ is complete must be strictly less expressive than full first-order logic. Moreover, in the propositional case it is very unlikely that a formula always has a polynomial-size normal form.

Note that Levesque's conjecture is weaker than the statement that there exists an algorithm which converts every first-order sentence into an equivalent one in $\mathcal{NF}$. The latter statement can be easily refuted since whether a first-order sentence is entailed by a proper KB is undecidable, but whether a $\mathcal{NF}$ sentence is entailed by a proper KB is decidable ($V$ is such a decision procedure).

In the next section, we briefly review Levesque's evaluation-based inference method for proper KBs. Section 3 contains our main result, that is, we show that not every sentence has an equivalent normal form. Section 4 considers the size of $\mathcal{NF}$ formulas in the propositional case.

## 2. Levesque's Evaluation-Based Reasoning Procedure $V$

The underlying language $\mathcal{L}$ is a standard first-order dialect with equality. There are countably infinitely many first-order variables and predicate symbols of every arity (including the binary equality predicate). In addition there is a countably infinite set $\mathcal{C} = \{d_1, d_2, \ldots\}$

of constants (but no other function symbols). The logical connectives are $\neg, \wedge$, and $\forall$. The atomic formulas of $\mathcal{L}$ are predicate symbols with variables or constants as arguments. The set of formulas of $\mathcal{L}$ is the least set which contains the atomic formulas, and if $\alpha$ and $\beta$ are in the set and $x$ is a variable, then $\neg\alpha$, $\alpha \wedge \beta$ and $\forall x.\alpha$ are in the set.

We sometimes refer to the propositional subset of the language, which consists of the ground atoms of $\mathcal{L}$ other than equality and is closed under negation and conjunction.

**Some Notation:** As usual, equality is written as $=$ using infix notation. We will freely use the connectives $\vee$, $\supset$, $\equiv$, and $\exists$, which are understood as the usual abbreviations. Formulas without free variables are called sentences. Variables are written as $x, y, z$ with sub- and superscripts. By *ewffs* we mean quantifier-free formulas whose only predicate is equality. For example, $(x = y \wedge z \neq d_1)$ is an ewff. We use $\forall\phi$ to denote the universal closure of $\phi$. For example, $\forall(x = y \wedge x \neq z \supset P(x, y, z))$ stands for $\forall x \forall y \forall z. x = y \wedge x \neq z \supset P(x, y, z)$. We write $\phi^x_d$ to denote $\phi$ with all free occurrences of $x$ replaced by constant $d$. A clause is a disjunction of literals, which we identify with the set of literals contained in it. We let the meta-variable $c$ range over clauses and we write $\bar{c}$ to denote the set $\{\bar{l} \mid l \in c\}$, where $\bar{l}$ is the complement of literal $l$. When $\Gamma$ is a finite set of formulas, we write $\bigwedge\Gamma$ to denote the conjunction of its elements (and *true*, when $\Gamma$ is empty). We use $H(\Gamma)$ to denote the set of constants appearing in the set of formulas $\Gamma$ and $H^+(\Gamma)$ to denote the set of constants appearing in $\Gamma$ plus an extra one not occurring in $\Gamma$. The set returned by $H^+(\Gamma)$ can be made unique by assuming that the constants are ordered and letting the new constant be the least constant not appearing in $\Gamma$. When $\Gamma$ is a singleton $\{\phi\}$ we simply write $H(\phi)$ or $H^+(\phi)$.

Levesque considered a special class of Tarskian interpretations called *standard interpretations,* where equality is interpreted as identity and the set of constants is isomorphic with the domain of discourse. As shown by the following definition and theorem, the restriction to standard interpretations can be captured precisely by a set of axioms, provided we limit ourselves to what is logically implied by *finite* sets of sentences.

**Definition 1** *Let the set $\mathcal{E}$ consist of the following axioms:*

1. $\forall x. x = x$;
2. $\forall(x_i = y \supset (P(x_1, \ldots, x_i, \ldots, x_n) \supset P(x_1, \ldots, y, \ldots, x_n)))$;
3. $\{(d_i \neq d_j) \mid i \neq j\}$.

(1) – (2) are a version of the axioms of equality. Here $P$ ranges over all predicate symbols including equality. (3) asserts the unique names assumption for constants.

**Theorem 2 (Levesque)** *Let $\Gamma$ be a finite set of sentences. Then $\mathcal{E} \cup \Gamma \models \phi$ iff every standard interpretation of $\Gamma$ is also a model of $\phi$.*

From now on we write $\Gamma \models_{\mathcal{E}} \phi$ for $\mathcal{E} \cup \Gamma \models \phi$.

A database can be viewed as a maximally consistent set of ground literals. Levesque proposed a generalization of a database called a proper KB, which is equivalent to a (possibly infinite) consistent set of ground literals.

In the following we use $e$ to range over ewffs and $\rho$ to range over atoms (excluding equality) whose arguments are distinct variables.

**Definition 3 (Levesque)** *A set of sentences $\Sigma$ is proper if $\mathcal{E} \cup \Sigma$ is consistent, $\Sigma$ is finite and every sentence in $\Sigma$ has the form $\forall(e \supset \rho)$ or $\forall(e \supset \neg\rho)$.*

In the propositional case, the definition simplifies to $\Sigma$ being a finite consistent set of ground literals excluding equality. (As equality plays no role in this case, $\mathcal{E}$ can be ignored.)

In general, a proper KB $\Sigma$ represents the set of ground literals $lits(\Sigma) = \{l\theta \mid \forall(e \supset l) \in \Sigma \text{ and } \mathcal{E} \models e\theta\}$, which is consistent, since $\mathcal{E} \cup \Sigma$ is consistent. As a special case, a database can be represented as a proper KB: each relation $R = \{\vec{d_1}, \ldots, \vec{d_m}\}$ is represented by $\forall(e \supset R(\vec{x}))$ and $\forall(\neg e \supset \neg R(\vec{x}))$, where $e$ is $\vec{x} = \vec{d_1} \vee \ldots \vee \vec{x} = \vec{d_m}$. But more importantly, a proper KB can represent an incomplete set of literals, by specifying some positive instances and some negative instances and leaving the status of the rest open.

Here is the example from the introduction formulated as a proper KB. The rephrasing is needed as the predicates of the right hand sides of the implications may only mention distinct variables.

- Ann likes Bob, and Dan likes Fred.
  $\forall x \forall y.\, x = ann \wedge y = bob \vee x = dan \wedge y = fred \supset Likes(x, y)$
- Ann does not like Dan.
  $\forall x \forall y.\, x = ann \wedge y = dan \supset \neg Likes(x, y)$
- Carol likes everyone.
  $\forall x \forall y.\, x = carol \supset Likes(x, y)$
- Eve does not like anyone other than Ann and herself.
  $\forall x \forall y.\, x = eve \wedge y \neq ann \wedge y \neq eve \supset \neg Likes(x, y)$

Again, note that this information cannot be expressed in a traditional database where, for example, we cannot leave open whether Ann likes Eve.

Levesque's evaluation-based inference procedure $V$ for proper KBs is defined as follows.

We use $\theta$ to range over substitutions of all variables by constants, and write $\phi\theta$ to mean the result of applying $\theta$ to formula $\phi$. We restrict our attention to Boolean queries, that is, sentences from $\mathcal{L}$. Given a proper KB $\Sigma$ and a sentence from $\mathcal{L}$, $V$ returns one of three values 0 (known to be false), 1 (known to be true), or $\frac{1}{2}$ (unknown). More precisely,

1. $V[\Sigma, \rho\theta] = \begin{cases} 1 & \text{if there is a } \forall(e \supset \rho) \text{ in } \Sigma \text{ such that } V[\Sigma, e\theta] = 1 \\ 0 & \text{if there is a } \forall(e \supset \neg\rho) \text{ in } \Sigma \text{ such that } V[\Sigma, e\theta] = 1 \\ \frac{1}{2} & \text{otherwise} \end{cases}$
2. $V[\Sigma, d = d'] = 1$ if $d$ is identical to $d'$, and 0 otherwise;
3. $V[\Sigma, \neg\phi] = 1 - V[\Sigma, \phi]$;
4. $V[\Sigma, \phi \wedge \psi] = \min\{V[\Sigma, \phi], V[\Sigma, \psi]\}$;
5. $V[\Sigma, \forall x\phi] = \min_{d \in H^+(\Sigma \cup \{\phi\})} V[\Sigma, \phi^x_d]$.

It is not hard to show that this procedure is logically sound, that is, whenever it returns 1 or 0, either the query or its negation follows from the knowledge base. $V$ is also obviously decidable as quantification is handled by a finite number of variable substitutions. As shown by Liu and Levesque (2003), it is also efficient in the sense of database retrieval.

To see why $V$ is incomplete, let $\Sigma$ be the set of sentences in our example KB and $\phi = (\mathit{Likes}(\mathit{ann}, \mathit{eve}) \vee \neg \mathit{Likes}(\mathit{ann}, \mathit{eve}))$. Then $\phi$ obviously follows from $\Sigma$, yet $V[\Sigma, \phi] = \frac{1}{2}$ because $V$ returns $\frac{1}{2}$ for both $\mathit{Likes}(\mathit{ann}, \mathit{eve})$ and $\neg \mathit{Likes}(\mathit{ann}, \mathit{eve})$. The problem is, roughly, that $V$ requires one of the disjuncts to be derivable in order for the whole disjunction to be derivable.

For a slightly more complex example, let $\phi = (p \supset q) \wedge (q \supset r)$, where $p = \mathit{Likes}(\mathit{ann}, \mathit{bob})$, $q = \mathit{Likes}(\mathit{ann}, \mathit{eve})$, and $r = \mathit{Likes}(\mathit{ann}, \mathit{dan})$. Then again, $V[\Sigma, \phi] = \frac{1}{2}$, but the correct answer should be 0 since $\Sigma \models \neg\phi$ because $\Sigma \models p$ and $\Sigma \models \neg r$. However, notice that the clauses $(p \supset q)$ and $(q \supset r)$ entail the clause $(p \supset r)$. If we were to conjoin $(p \supset r)$ to $\phi$, logical equivalence would be preserved and $V$ would now return the correct answer 0 since $V[\Sigma, p] = 1$ and $V[\Sigma, r] = 0$.

Despite this limitation, Levesque showed that for queries in a certain normal form called $\mathcal{NF}$, $V$ is actually complete. We first state the result, followed by the definition of $\mathcal{NF}$.

**Theorem 4 (Levesque)** *Let $\Sigma$ be proper. Then*
*for every $\phi \in \mathcal{NF}$, $V[\Sigma, \phi] = 1$ iff $\Sigma \models_{\mathcal{E}} \phi$; and $V[\Sigma, \phi] = 0$ iff $\Sigma \models_{\mathcal{E}} \neg\phi$.*

The definition of $\mathcal{NF}$ is based on that of *logical separability*:

**Definition 5 (Levesque)** *A set $\Gamma$ of sentences is logically separable iff for every consistent set of ground literals $L$, if $L \cup \Gamma$ has no standard interpretation, then $L \cup \{\phi\}$ is inconsistent for some $\phi \in \Gamma$.*

The intuition behind logical separability is that if a consistent set of literals entails a disjunction, then one of the disjuncts must be entailed. To see this, consider the propositional case with $\Gamma = \{\neg p, \neg q\}$ and $L$ a consistent set of propositional literals. Suppose $L \cup \Gamma$ has no standard interpretation, which in the propositional case is the same as $L \cup \Gamma$ is inconsistent or, equivalently, $L \models (p \vee q)$. Then $L$ must contain either $p$ or $q$. Hence either $L \cup \{\neg p\}$ or $L \cup \{\neg q\}$ is inconsistent, that is, either $L \models p$ or $L \models q$. In any cases, this proves that $\Gamma$ is logically separable.

The set $\{p, \neg p\}$, on the other hand, is not logically separable. This is because $\{p, \neg p\}$ by itself is already inconsistent and we can let $L$ be the empty set. In this case, both $L \cup \{p\}$ and $L \cup \{\neg p\}$ are consistent.

**Definition 6 (Levesque)** *$\mathcal{NF}$ is the least set such that*

1. *if $\phi$ is a ground atom or ewff, then $\phi \in \mathcal{NF}$;*
2. *if $\phi \in \mathcal{NF}$, then $\neg\phi \in \mathcal{NF}$;*
3. *if $\Gamma \subseteq \mathcal{NF}$, $\Gamma$ is logically separable, and $\Gamma$ is finite, then $\bigwedge \Gamma \in \mathcal{NF}$;*
4. *if $\Gamma \subseteq \mathcal{NF}$, $\Gamma$ is logically separable, and for some $\phi$, $\Gamma = \{\phi^x_d \mid d \in \mathcal{C}\}$, then $\forall x \phi \in \mathcal{NF}$.*

(1) and (2) say, roughly, that $\mathcal{NF}$ contains all ground atoms and is closed under negation. (3) and (4) say that closure under conjunction and universal generalization is restricted to formulas which are logically separable.

The idea behind the definition is that a formula in $\mathcal{NF}$, in a sense, does not contain any logical puzzles. This is the case, for example, for any non-tautologous clause. To see why, consider any consistent set of literals. Similar to the earlier example of $\{\neg p, \neg q\}$, it can be shown that the set is logically separable. Hence the conjunction of the literals and its negation, which is a clause, are in $\mathcal{NF}$. On the other hand, a tautology like $(p \vee \neg p)$ is not in $\mathcal{NF}$. As shown above, the set $\{p, \neg p\}$ is not logically separable. Hence neither $p \wedge \neg p$ nor its negation, that is, $(p \vee \neg p)$ is in $\mathcal{NF}$.

Levesque (1998) showed that in the propositional case, a CNF formula is in $\mathcal{NF}$ if its clauses are non-tautologous and closed under resolution, that is, the resolvent of any two of the clauses also belongs to the clauses. Consider $\alpha = (p \supset q) \wedge (q \supset r)$ from our earlier example. It is not in $\mathcal{NF}$ since its clauses are not closed under resolution. However, $(p \supset q) \wedge (q \supset r) \wedge (p \supset r)$ is in $\mathcal{NF}$.

In the first-order case, two examples of formulas in $\mathcal{NF}$ are $\forall x(P(x) \wedge Q(x))$ and $\exists x(P(x) \vee Q(x))$. To see why let $d$ be any constant. Just as $\{p, q\}$ is logically separable, so is $\{P(d), Q(d)\}$. Thus $(P(d) \wedge Q(d))$ is in $\mathcal{NF}$. Now consider $\Gamma = \{(P(d) \wedge Q(d)) \mid d \in \mathcal{C}\}$. Let $L$ be any consistent set of literals. If $L \cup \Gamma$ has no standard model, then $L$ must contain $\neg P(d)$ or $\neg Q(d)$ for some $d$, hence $L \cup \{(P(d) \wedge Q(d))\}$ is inconsistent for some $d$. Thus $\Gamma$ is logically separable, and so $\forall x(P(x) \wedge Q(x))$ is in $\mathcal{NF}$. Similarly, $\forall x(\neg P(x) \wedge \neg Q(x))$ is in $\mathcal{NF}$. Therefore, $\neg\forall x(\neg P(x) \wedge \neg Q(x))$, that is, $\exists x(P(x) \vee Q(x))$, is in $\mathcal{NF}$.

Note that the use of the expression *normal form* is somewhat non-standard. Unlike traditional normal forms like CNF, it is not even clear whether membership in $\mathcal{NF}$ is decidable. While Levesque pointed out a number of classes of first-order formulas where membership in $\mathcal{NF}$ can be decided syntactically, this is unlikely in general. For example, he showed that $\exists x.\neg R(a, x) \wedge R(x, b)$ is not in $\mathcal{NF}$.

It turns out that Levesque's original definition of logical separability (Def. 5) is a little too strong and rules out certain sentences from being in $\mathcal{NF}$ which we definitely would like to be in. The problem with the definition is that it mixes the use of standard and regular Tarskian interpretations. This has the peculiar effect that formulas like $\forall x(x = a \supset P(x))$, which make up proper KBs, are themselves not in $\mathcal{NF}$.[1] To see why, by the definition of $\mathcal{NF}$, for $\forall x(x = a \supset P(x))$, that is, $\forall x\neg(x = a \wedge \neg P(x))$, to be in $\mathcal{NF}$, we must have $(d = a \wedge \neg P(d)) \in \mathcal{NF}$ for every constant $d$, which requires that $\{d = a, \neg P(d)\}$ be logically separable. However, let $b$ be a constant distinct from $a$, then $\{b = a, \neg P(b)\}$ is not logically separable. The reason is that $\{b = a, \neg P(b)\}$ has no standard model because of the built-in unique names assumption, but both $\{b = a\}$ and $\{\neg P(b)\}$ are consistent in classical logic.

It turns out that the above anomaly is easy to fix by using the following, slightly weaker definition of logical separability, which only talk about standard interpretations.

**Definition 7** *A set $\Gamma$ of sentences is logically separable iff for every consistent set of ground literals $L$, if $L \cup \Gamma$ has no standard interpretation, then $L \cup \{\phi\}$ has no standard interpretation for some $\phi \in \Gamma$.*

1. This anomaly was first observed by Thomas Eiter (personal communication).

With the new definition, $\{b = a, \neg P(b)\}$ is now logically separable, since $\{b = a\}$ has no standard model. As a result, we can show that $\forall x(x = a \supset P(x))$ is now in $\mathcal{NF}$. Note that $\mathcal{NF}$ using the new definition of logical separability is strictly bigger than the original $\mathcal{NF}$, and we will use the new version from now on. Our main result, which says that there are sentences with no equivalent sentence in $\mathcal{NF}$, then trivially extends to Levesque's original definition.

Before we turn to that, let us briefly recall what $\mathcal{NF}$ is good for and point to some related work. When a user poses a query in $\mathcal{NF}$ to a proper KB, then all we need is $V$ to obtain a correct (sound and complete) answer with respect to logical entailment. Moreover, as we mentioned earlier, $V$ has been proven to be as efficient as database retrieval (Liu & Levesque, 2003), and standard database technology can be brought to bear for its implementation.

In this regard, there is also an interesting connection to recent work on evaluating certain queries in description logics (Baader, Calvanese, McGuiness, Nardi, & Patel-Schneider, 2003). A description-logic KB consists of two parts, a TBOX with terminological definitions like "a mother is a female person with at least one child" and an ABOX, which is a set of atomic formulas. An ABOX is just a special case of a proper KB. Most importantly, an ABOX does not make the closed world assumption, just as proper KBs. Calvanese, de Giacomo, Lembo, Lenzerini, and Rosati (2006) showed that for conjunctive queries, which consist of conjunctions of atoms with existentially quantified variables, query answering can be reduced to database retrieval as well. It is interesting to note that the queries they consider are in $\mathcal{NF}$. While they only consider a small fragment of $\mathcal{NF}$, they go beyond proper KBs as they also perform terminological reasoning (using the TBOX). We remark that there have been other extensions of proper KBs by explicitly allowing disjunctions (Lakemeyer & Levesque, 2002; Liu, Lakemeyer, & Levesque, 2004), but reasoning there goes beyond database retrieval.

## 3. First-Order $\mathcal{NF}$ Is Not Expressive

Levesque showed that in the propositional case, every formula $\phi$ can be transformed into an equivalent one in $\mathcal{NF}$. His transformation is this. Convert the formula to CNF, and run resolution repeatedly on this set of clauses, deleting any tautologous or subsumed ones until no new clauses are generated. The resulting set of clauses is the set of prime implicates of $\phi$. The conjunction of these clauses is in $\mathcal{NF}$ and it is equivalent to $\phi$. However, this transformation cannot be extended to the first-order case. To see why, consider $\phi = \forall xyz[R(x, y) \wedge R(y, z) \supset R(x, z)]$, which says that $R$ is transitive. If we run the first-order version of Levesque's transformation on $\phi$, we would end up with an infinite set of clauses, each of the format $R(x_1, x_2) \wedge \ldots \wedge R(x_n, x_{n+1}) \supset R(x_1, x_{n+1})$, where $n \geq 2$.

In this section, we will prove that in the first-order case, not every formula is equivalent to one in $\mathcal{NF}$. For this purpose, we will need a first-order version of prime implicates. In the propositional case, prime implicates are defined as follows:

**Definition 8** *Let $\Gamma$ be a theory. An implicate of $\Gamma$ is a non-tautologous clause $c$ such that $\Gamma \models c$. A prime implicate of $\Gamma$ is an implicate $c$ of $\Gamma$ such that for no proper subset $c'$ of $c$ does $\Gamma \models c'$.*

Since we only consider standard interpretations, we can easily generalize prime implicates to the first-order case:

**Definition 9** *Let $\phi \in \mathcal{L}$. An implicate of $\phi$ is a non-tautologous ground clause $c$ such that $\phi \models_{\mathcal{E}} c$. A prime implicate of $\phi$ is an implicate $c$ of $\phi$ such that for no proper subset $c'$ of $c$ does $\phi \models_{\mathcal{E}} c'$. We use $\mathsf{PI}(\phi)$ to denote the set of prime implicates of $\phi$.*

If $\models_{\mathcal{E}} \phi \equiv \psi$, that is, $\mathcal{E} \models \phi \equiv \psi$, then both $\mathcal{E} \cup \{\phi\}$ and $\mathcal{E} \cup \{\psi\}$ entail the same sentences and, in particular, $\mathsf{PI}(\phi) = \mathsf{PI}(\psi)$.

The following is a basic property of prime implicates:

**Proposition 10** *Let $c$ be a non-tautologous ground clause. Then $\phi \models_{\mathcal{E}} c$ iff there exists a $c' \in \mathsf{PI}(\phi)$ such that $c' \subseteq c$.*

In this note, by the length of a formula, we mean the number of predicate symbols, variables, constants and logical connectives contained in the formula. By the length of a clause, we mean the length of the corresponding disjunctive formula. The key property in this note is defined as follows:

**Definition 11** *We say that $\mathsf{PI}(\phi)$ is bounded if there exists a number $n$ such that the length of every member in $\mathsf{PI}(\phi)$ is at most $n$. If $\mathsf{PI}(\phi)$ is bounded, we use $\mathsf{B}(\phi)$ to denote the maximum length of a member in $\mathsf{PI}(\phi)$.*

We will show that for every $\phi \in \mathcal{NF}$, $\mathsf{PI}(\phi)$ is bounded. Then for any $\phi$ which has an equivalent $\phi'$ in $\mathcal{NF}$, $\mathsf{PI}(\phi)$ is bounded too. However, there exist formulas $\phi$ such that $\mathsf{PI}(\phi)$ is not bounded. Thus not every formula can be transformed into an equivalent one in $\mathcal{NF}$.

The proof that the prime implicates of formulas in $\mathcal{NF}$ are bounded will proceed by induction. The following lemma is useful to establish the induction for conjunctions and their negations.

**Lemma 12** *Let $\Gamma = \{\phi_1, \ldots, \phi_n\}$.*

1. *If $\bigwedge \Gamma \in \mathcal{NF}$, then $\mathsf{PI}(\bigwedge \Gamma) \subseteq \bigcup_i \mathsf{PI}(\phi_i)$.*
2. *$\mathsf{PI}(\neg \bigwedge \Gamma) \subseteq \{\bigcup_i c_i \mid c_i \in \mathsf{PI}(\neg\phi_i), i = 1, \ldots, n\}$.*

**Proof:**

1. Let $c \in \mathsf{PI}(\bigwedge \Gamma)$. Then $\bigwedge \Gamma \models_{\mathcal{E}} c$. By Theorem 2, $\Gamma \cup \overline{c}$ has no standard interpretation. Since $\bigwedge \Gamma \in \mathcal{NF}$, $\Gamma$ is logically separable. Since $c$ is non-tautologous, $\overline{c}$ is consistent. Thus $\{\phi_i\} \cup \overline{c}$ has no standard interpretation for some $i$, and so $\phi_i \models_{\mathcal{E}} c$. Let $c' \subseteq c$ such that $\phi_i \models_{\mathcal{E}} c'$. Then $\bigwedge \Gamma \models_{\mathcal{E}} c'$. Since $c \in \mathsf{PI}(\bigwedge \Gamma)$, $c' = c$. Thus $c \in \mathsf{PI}(\phi_i)$.

2. Let $c \in \mathsf{PI}(\neg \bigwedge \Gamma)$. Then $\neg \bigwedge \Gamma \models_{\mathcal{E}} c$. Thus $\neg\phi_i \models_{\mathcal{E}} c$ for all $i$. By Proposition 10, for each $i$, there exists $c_i \in \mathsf{PI}(\neg\phi_i)$ (and hence $\neg\phi_i \models_{\mathcal{E}} c_i$) such that $c_i \subseteq c$. Then $\neg \bigwedge \Gamma \models_{\mathcal{E}} \bigcup_i c_i$ and $\bigcup_i c_i \subseteq c$. Since $c \in \mathsf{PI}(\neg \bigwedge \Gamma)$, $c = \bigcup_i c_i$. □

Note that, if the prime implicates of the $\phi_i$ are bounded, it follows easily from the lemma that the prime implicates of $\bigwedge\Gamma$ and $\neg\bigwedge\Gamma$ are bounded as well. To obtain a similar result for quantified formulas is more complicated. The obvious generalization of Lemma 12, replacing $\Gamma$ by $\{\phi^x_{d_1}, \phi^x_{d_2}, \ldots\}$ and $\bigwedge\Gamma$ by $\forall x\phi$ does not work, as this would lead to an infinite union of sets in the first part and an infinite union of clauses in the second part. We can get around this by observing the similarity between $\mathsf{PI}(\phi^x_b)$ and $\mathsf{PI}(\phi^x_d)$, where $b$ and $d$ are constants not appearing in $\phi$, as shown by Proposition 14 below.

We begin with a property that will be useful in the proof. Let $*$ be a mapping from $\mathcal{C}$ to $\mathcal{C}$. We use $\phi^*$ to denote $\phi$ with every constant $d$ replaced by $d^*$. We use $\Gamma^*$ to denote $\{\phi^* \mid \phi \in \Gamma\}$.

**Proposition 13 (Levesque)** *Let $*$ be a bijection from $\mathcal{C}$ to $\mathcal{C}$. If $\Gamma \models_{\mathcal{E}} \phi$, then $\Gamma^* \models_{\mathcal{E}} \phi^*$.*

**Proposition 14** *Let $\phi$ be a formula with a single free variable $x$. Let $b, d$ be constants in $\mathcal{C}$ not appearing in $\phi$. Let $*$ be the bijection that swaps $b$ and $d$ and leaves all other constants unchanged. Then $\mathsf{PI}(\phi^x_d) = \{c^* \mid c \in \mathsf{PI}(\phi^x_b)\}$.*

**Proof:** Let $c \in \mathsf{PI}(\phi^x_b)$. Then $\phi^x_b \models_{\mathcal{E}} c$. By Proposition 13, $(\phi^x_b)^* \models_{\mathcal{E}} c^*$, that is, $\phi^x_d \models_{\mathcal{E}} c^*$. Let $c' \subseteq c^*$ such that $\phi^x_d \models_{\mathcal{E}} c'$. Then $\phi^x_b \models_{\mathcal{E}} c'^*$. Since $c \in \mathsf{PI}(\phi^x_b)$ and $c'^* \subseteq c$, $c'^* = c$, and hence $c' = c^*$. Thus $c^* \in \mathsf{PI}(\phi^x_d)$. Similarly, if $c \in \mathsf{PI}(\phi^x_d)$, then $c^* \in \mathsf{PI}(\phi^x_b)$. Therefore, $\mathsf{PI}(\phi^x_d) = \{c^* \mid c \in \mathsf{PI}(\phi^x_b)\}$. □

Basically, the above proposition says that the prime implicates of $\phi^x_b$ and $\phi^x_d$ are the same modulo constant renaming.

**Lemma 15** *Let $\phi$ be a formula with a single free variable $x$.*

1. *If $\forall x\phi \in \mathcal{NF}$, and for all constants $d \in \mathcal{C}$, $\mathsf{PI}(\phi^x_d)$ is bounded, then $\mathsf{PI}(\forall x\phi)$ is also bounded.*

2. *If for all constants $d \in \mathcal{C}$, $\mathsf{PI}(\neg\phi^x_d)$ is bounded, then $\mathsf{PI}(\neg\forall x\phi)$ is also bounded.*

**Proof:**

1. Since for all $d \in \mathcal{C}$, $\mathsf{PI}(\phi^x_d)$ is bounded, we let $n = \max\{\mathsf{B}(\phi^x_d) \mid d \in H^+(\phi)\}$. By Proposition 14, for any $d \in \mathcal{C}$, $\mathsf{PI}(\phi^x_d)$ is a relabeling of $\mathsf{PI}(\phi^x_b)$ for some $b$ in $H^+(\phi)$. Thus for any $d \in \mathcal{C}$, $\mathsf{PI}(\phi^x_d)$ is bounded by $n$. We will show that $\mathsf{PI}(\forall x\phi)$ is also bounded by this $n$, by showing that every element of $\mathsf{PI}(\forall x\phi)$ is also an element of $\mathsf{PI}(\phi^x_d)$, for some $d \in \mathcal{C}$.

   So suppose that $c \in \mathsf{PI}(\forall x\phi)$. Then $\forall x\phi \models_{\mathcal{E}} c$. Thus $\{\forall x\phi\} \cup \overline{c}$ has no standard interpretation. So $\{\phi^x_d \mid d \in \mathcal{C}\} \cup \overline{c}$ has no standard interpretation. Since $\forall x\phi \in \mathcal{NF}$, $\{\phi^x_d \mid d \in \mathcal{C}\}$ is logically separable. Thus there exists $d \in \mathcal{C}$ such that $\{\phi^x_d\} \cup \overline{c}$ has no standard interpretation. So $\phi^x_d \models_{\mathcal{E}} c$. Let $c' \subseteq c$ such that $\phi^x_d \models_{\mathcal{E}} c'$. Then $\forall x\phi \models_{\mathcal{E}} c'$. Since $c \in \mathsf{PI}(\forall x\phi)$, $c' = c$. Thus $c \in \mathsf{PI}(\phi^x_d)$.

2. As in Part 1, we let $n = \max\{\mathsf{B}(\neg\phi^x_d) \mid d \in H^+(\phi)\}$. Then for any $d \in \mathcal{C}$, $\mathsf{PI}(\neg\phi^x_d)$ is bounded by $n$. We will show that $\mathsf{PI}(\neg\forall x\phi)$ is bounded by $(m+n+1)\cdot n$, where $m$ is the length of $\phi$. This is done by showing that for any $c \in \mathsf{PI}(\neg\forall x\phi)$, there exists a set $D$ of no more than $(m+n+1)$ constants such that for all $d \in D$, there exists $c_d \in \mathsf{PI}(\neg\phi^x_d)$ such that $c = \bigcup_{d\in D} c_d$.

   So suppose that $c \in \mathsf{PI}(\neg\forall x\phi)$. Then $\neg\forall x\phi \models_{\mathcal{E}} c$, *i.e.*, $\exists x\neg\phi \models_{\mathcal{E}} c$. Let $d$ be an arbitrary constant in $\mathcal{C}$. Then $\neg\phi^x_d \models_{\mathcal{E}} c$ because $\neg\phi^x_d \models_{\mathcal{E}} \exists x\neg\phi$. By Proposition 10, there exists $c_d \in \mathsf{PI}(\neg\phi^x_d)$ such that $c_d \subseteq c$. Now let $b$ be a constant that appears in neither $\phi$ nor $c$ with $c_b \in \mathsf{PI}(\neg\phi^x_b)$ and $c_b \subseteq c$. Then $b$ does not appear in $c_b$, and the length of $c_b$ is at most $n$. Let $D = H(\phi) \cup H(c_b) \cup \{b\}$. Then $D$ has no more than $(m+n+1)$ elements. We will show that $c = \bigcup_{d\in D} c_d$. To do so, let $a$ be an arbitrary constant not in $D$. Let $*$ be the bijection that swaps $b$ and $a$ and leaves all other constants unchanged. Since $c_b \in \mathsf{PI}(\neg\phi^x_b)$ we have $\neg\phi^x_b \models_{\mathcal{E}} c_b$, and, by Proposition 13, $(\neg\phi^x_b)^* \models_{\mathcal{E}} (c_b)^*$. Since neither $b$ nor $a$ appears in $\phi$ or $c_b$, $(\neg\phi^x_b)^* = \neg\phi^x_a$ and $(c_b)^* = c_b$. Hence $\neg\phi^x_a \models_{\mathcal{E}} c_b$. So we have that for all $d \in D$, $\neg\phi^x_d \models_{\mathcal{E}} c_d$; and for all $a \notin D$, $\neg\phi^x_a \models_{\mathcal{E}} c_b$. Thus $\exists x\neg\phi \models_{\mathcal{E}} \bigcup_{d\in D} c_d$, which is a subset of $c$. Since $c \in \mathsf{PI}(\exists x\neg\phi)$, $c = \bigcup_{d\in D} c_d$. □

We now have all the pieces in hand to prove the main theorem:

**Theorem 16** *Let $\psi \in \mathcal{NF}$. Then $\mathsf{PI}(\psi)$ is bounded.*

**Proof:** For technical reasons, it is easier to prove a slightly more general statement, namely that both $\mathsf{PI}(\psi)$ and $\mathsf{PI}(\neg\psi)$ are bounded provided that $\psi \in \mathcal{NF}$. The proof is by induction on $\psi$.

1. $\psi$ is a ground atom or ewff. If $\psi$ is a ground atom, then $\mathsf{PI}(\psi) = \{\psi\}$ and $\mathsf{PI}(\neg\psi) = \{\neg\psi\}$, hence they are bounded. If $\psi$ is a ground ewff that is true, then $\psi$ does not entail any ground clause, and hence $\mathsf{PI}(\psi)$ is the empty set; if $\psi$ is a ground ewff that is false, then $\psi$ entails the empty clause, and hence $\mathsf{PI}(\psi)$ is the set consisting of the empty clause. Therefore, if $\psi$ is a ground ewff, both $\mathsf{PI}(\psi)$ and $\mathsf{PI}(\neg\psi)$ are bounded.

2. $\psi$ is $\neg\phi$. By induction, $\mathsf{PI}(\phi)$ and $\mathsf{PI}(\neg\phi)$ are bounded. Since $\mathsf{PI}(\phi) = \mathsf{PI}(\neg\neg\phi)$, both $\mathsf{PI}(\neg\phi)$ and $\mathsf{PI}(\neg\neg\phi)$ are bounded.

3. $\psi$ is $\bigwedge\Gamma$. By induction, for all $\phi \in \Gamma$, $\mathsf{PI}(\phi)$ and $\mathsf{PI}(\neg\phi)$ are bounded. By Lemma 12, $\mathsf{PI}(\bigwedge\Gamma)$ is bounded by $\max\{\mathsf{B}(\phi) \mid \phi \in \Gamma\}$, and $\mathsf{PI}(\neg\bigwedge\Gamma)$ by the sum of $\mathsf{B}(\neg\phi)$ for $\phi \in \Gamma$.

4. $\psi$ is $\forall x\phi$. By induction, for any constant $d$, $\mathsf{PI}(\phi^x_d)$ and $\mathsf{PI}(\neg\phi^x_d)$ are bounded. By Lemma 15, both $\mathsf{PI}(\forall x\phi)$ and $\mathsf{PI}(\neg\forall x\phi)$ are bounded. □

As an easy corollary, we have:

**Corollary 17** *Not every sentence has an equivalent one in $\mathcal{NF}$.*

**Proof:** Let $\phi{=}\forall xyz[R(x,y) \wedge R(y,z) \supset R(x,z)]$, which says that $R$ is transitive. Then for all $n$, the following is in $\mathsf{PI}(\phi)$:

$$R(d_1, d_2) \wedge \ldots \wedge R(d_n, d_{n+1}) \supset R(d_1, d_{n+1}).$$

Thus $\mathsf{PI}(\phi)$ is not bounded. Suppose that there exists $\phi' \in \mathcal{NF}$ such that $\phi$ and $\phi'$ are equivalent. Then $\mathsf{PI}(\phi) = \mathsf{PI}(\phi')$. By Theorem 16, $\mathsf{PI}(\phi')$ is bounded, a contradiction. □

Moreover, it is easy to generalize our inexpressiveness result:

**Theorem 18** *There does not exist a class $\mathcal{F}$ of sentences with these properties:*

1. *every formula has an equivalent one in $\mathcal{F}$;*
2. *$V$ is logically complete for $\mathcal{F}$ (i.e., for every proper KB $\Sigma$ and every $\phi \in \mathcal{F}$, if $\Sigma \models_{\mathcal{E}} \phi$ then $V[\Sigma, \phi] = 1$, and if $\Sigma \models_{\mathcal{E}} \neg\phi$ then $V[\Sigma, \phi] = 0$);*
3. *if $\neg\phi \in \mathcal{F}$ then $\phi \in \mathcal{F}$; if $\bigwedge \Gamma \in \mathcal{F}$, then $\Gamma \subseteq \mathcal{F}$; and if $\forall x\phi \in \mathcal{F}$, then $\phi^x_d \in \mathcal{F}$ for all constants $d$.*

Note that the theorem does not require logical separability, only that $V$ be complete for $\mathcal{F}$. We call any set of formulas that satisfies the third requirement *downward saturated*, which, besides being a desirable property of a normal form, is needed for technical reasons.

**Proof:** Suppose, to the contrary, that there exists a class $\mathcal{F}$ of sentences which satisfies the three properties stated above. As in the case of $\mathcal{NF}$, we can show that $\mathsf{PI}(\phi)$ is bounded for every $\phi \in \mathcal{F}$ and use the same sentence as in the proof of Corollary 17 to obtain a contradiction.

The boundedness proof is almost identical to the argument before for $\mathcal{NF}$, and we will not repeat it here. Instead we just note the necessary changes. In fact, the only changes needed are in the proofs of Item 1 in Lemma 12 and 15, where we appeal to logical separability to show that $\phi_i \models_{\mathcal{E}} c$ for some $i$ respectively $\phi^x_d \models_{\mathcal{E}} c$ for some $d$. Here we show that the same conclusions can be drawn using the assumption that $V$ is complete for $\mathcal{F}$.

First, note the following: Let $\phi \in \mathcal{F}$, and let $c$ be a non-tautologous ground clause. Then $\overline{c}$ is essentially a proper KB. Thus if $\phi \models_{\mathcal{E}} c$, then $\overline{c} \models_{\mathcal{E}} \neg\phi$, and hence $V[\overline{c}, \phi] = 0$, by completeness of $V$ for $\mathcal{F}$.

- Change in Lemma 12, Item 1:
  Let $c \in \mathsf{PI}(\bigwedge \Gamma)$. Then $\bigwedge \Gamma \models_{\mathcal{E}} c$. Since $\bigwedge \Gamma \in \mathcal{F}$, $V[\overline{c}, \bigwedge \Gamma] = 0$. By the definition of $V$, $V[\overline{c}, \phi_i] = 0$ for some $\phi_i \in \Gamma$. By soundness of $V$, $\overline{c} \models_{\mathcal{E}} \neg\phi_i$. Thus $\phi_i \models_{\mathcal{E}} c$.
- Change in Lemma 15, Item 1:
  Let $c \in \mathsf{PI}(\forall x\phi)$. Then $\forall x\phi \models_{\mathcal{E}} c$. Since $\forall x\phi \in \mathcal{F}$, $V[\overline{c}, \forall x\phi] = 0$. By the definition of $V$, $V[\overline{c}, \phi^x_d] = 0$ for some constant $d$. By soundness of $V$, $\overline{c} \models_{\mathcal{E}} \neg\phi^x_d$. Thus $\phi^x_d \models_{\mathcal{E}} c$.

With these small changes the proofs of the two lemmas go through for $\mathcal{F}$ instead of $\mathcal{NF}$. Finally, the proof of Theorem 16 carries over without any change, since the induction works for any downward-saturated set. □

## 4. Propositional $\mathcal{NF}$ Is Not Succinct

In the propositional case, Levesque's transformation to $\mathcal{NF}$, that is, taking the conjunction of the prime implicates of the formula, may cause an exponential blowup in the size of the formula. This is because the number of prime implicates of a formula with $n$ propositions is exponential in $n$ in the worst case (Chandra & Markowsky, 1978). In this section, we show that in the propositional case, under a certain complexity assumption, not every formula has a polynomial-size equivalent one in $\mathcal{NF}$. This is done by relating $\mathcal{NF}$ to an existing result in knowledge compilation.

Knowledge compilation (Selman & Kautz, 1996; Darwiche & Marquis, 2002) has been proposed as one of the main techniques to deal with the computational intractability of general propositional reasoning. In this approach, a tractable language, which usually means a language such that whether a clause is entailed by a formula from the language can be decided in polynomial time, is identified as the target compilation language. A propositional theory is first compiled off-line into the target language, and the result is then used on-line to answer multiple queries. The main motivation here is to shift most of the computational cost into the off-line phase, which is amortized over all on-line queries. As shown in the following, $\mathcal{NF}$ can serve as a knowledge compilation language. The reason is that in the propositional case, answering an arbitrary query against a proper KB is equivalent to answering a clausal query against an arbitrary KB. As mentioned in Section 2, in the propositional case, a proper KB is simply a consistent set of literals.

**Proposition 19** *Clausal entailment on $\mathcal{NF}$ can be decided in polynomial time.*

**Proof:** Let $\phi \in \mathcal{NF}$, and let $c$ be a non-tautologous clause. Then $\overline{c}$ is a proper KB. Thus $\phi \models c$ iff $\overline{c} \models \neg\phi$ iff $V[\overline{c}, \phi] = 0$, by soundness and completeness of $V$ for $\mathcal{NF}$. Clearly, in the propositional case, $V$ runs in polynomial time. $\square$

The following is a well-known result in knowledge compilation:

**Theorem 20** (Selman & Kautz, 1996) *Unless $NP \subseteq P/poly$, there does not exist a class $\mathcal{F}$ of formulas such that every propositional formula has a polynomial-size equivalent one in $\mathcal{F}$, and clausal entailment on $\mathcal{F}$ can be decided in polynomial time.*

The complexity class *P/poly*, also known as *non-uniform P*, originated in circuit complexity (Boppana & Sipser, 1990). Roughly, a problem is in *P/poly* if for every integer $n$ there exists a circuit of size polynomial in $n$ that solves instances of size $n$. Without going into further details, $NP \subseteq P/poly$ implies the collapse of the polynomial hierarchy at the second level, which is considered very unlikely.

As an easy corollary of the above proposition and theorem, we have:

**Corollary 21** *Unless $NP \subseteq P/poly$, not every propositional formula has a polynomial-size equivalent one in $\mathcal{NF}$.*

In other words, it is very unlikely that we can obtain compact $\mathcal{NF}$ representations for arbitrary propositional formulas.

## 5. Conclusion

Levesque remarked in his paper that he did not envision the use of $\mathcal{NF}$ in the sense of query optimization by taking an arbitrary query and converting it into $\mathcal{NF}$ before handing it to $V$. His main argument was the high computational cost, which usually cannot be afforded on-line. Besides, except for special cases it is not even clear how to convert a formula into $\mathcal{NF}$, if one exists. Instead he suggested that $\mathcal{NF}$ could be a guideline for users to formulate "good" queries which can be evaluated efficiently.

The contribution of this technical note is to point out some of the limits even of this use of $\mathcal{NF}$. In the propositional case our result says that most likely there will be queries which cannot be both in $\mathcal{NF}$ and compactly representable. In some sense, this is not all bad news, since in practice queries tend to be very small compared to the knowledge base. In the first-order case our result is more serious as we showed that there are queries which do not have a normal form at all. In other words, no matter how ingenious a user might be, there will always be queries which have no easy-to-answer form, at least if we insist on a form which is independent of the knowledge base as in $\mathcal{NF}$. Indeed, it may still be possible to find another notion of normal form which depends in some way on the knowledge base, for example, the constants it contains. But that is future work.

## Acknowledgments

We thank Hector Levesque for many helpful discussions on the topic of this paper and reading an earlier version of the paper. We also thank the anonymous reviewers for their detailed comments on improving the presentation of this paper.

## References

Baader, F., Calvanese, D., McGuiness, D., Nardi, D., & Patel-Schneider, P. (2003). *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press.

Boppana, R. B., & Sipser, M. (1990). The complexity of finite functions. In van Leeuwen, J. (Ed.), *Handbook of Theoretical Computer Science*, Vol. A, pp. 757–804. Elsevier.

Calvanese, D., de Giacomo, G., Lembo, D., Lenzerini, M., & Rosati, R. (2006). Data complexity of query answering in description logics. In *Proc. of the Tenth International Conference on Principles of Knowledge Representation and Reasoning (KR-06)*, pp. 260–270.

Chandra, A. K., & Markowsky, G. (1978). On the number of prime implicants. *Discrete Mathematics*, *24*, 7–11.

Darwiche, A., & Marquis, P. (2002). A knowledge compilation map. *Journal of Artificial Intelligence Research*, *17*, 229–264.

Lakemeyer, G., & Levesque, H. J. (2002). Evaluation-based reasoning with disjunctive information in first-order knowledge bases. In *Proc. of the Eighth International Conference on Principles of Knowledge Representation and Reasoning (KR-02)*, pp. 73–81.

Levesque, H. J. (1998). A completeness result for reasoning with incomplete first-order knowledge bases. In *Proc. of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR-98)*, pp. 14–23.

Liu, Y., Lakemeyer, G., & Levesque, H. J. (2004). A logic of limited belief for reasoning with disjunctive information. In *Proc. of the Ninth International Conference on Principles of Knowledge Representation and Reasoning (KR-04)*, pp. 587–597.

Liu, Y., & Levesque, H. J. (2003). A tractability result for reasoning with incomplete first-order knowledge bases. In *Proc. of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03)*, pp. 83–88.

Selman, B., & Kautz, H. (1996). Knowledge compilation and theory approximation. *Journal of the ACM*, *43*(2), 193–224.