Theory Prime Implicates of First Order Formulas

Manoj K. Raut and Arindama Singh
Department of Mathematics
Indian Institute of Technology Madras
Chennai-600036, India

Abstract: We provide a knowledge compilation technique which deals with the computational intractability of first order reasoning problems. The notion of prime implicates is extended to theory prime implicates in first order case. We provide an algorithm to compute the theory prime implicates of a knowledge base $X$ with respect to a knowledge base $Y$ where both $X$ and $Y$ are assumed to be in Skolem Conjunctive Normal Form. Partial correctness of the algorithm is proved.

Keywords: Theory prime implicates, First order logic, Knowledge compilation

1 Introduction

Propositional entailment is a fundamental issue in artificial intelligence due to its high complexity. Checking whether a query is logically entailed by the knowledge base is intractable. To overcome the computational intractability, the computational effort is split into two phases such as off-line and on-line. In the off-line phase the knowledge base is compiled into a tractable form with respect to which queries can be answered in an on-line phase so that the latter would be tractable. In such type of compilation most of the computational overhead is shifted into the off-line phase which is amortized over the answers to queries. The off-line computation is known as knowledge compilation.

Several approaches for knowledge compilation have been suggested so far. The first kind of such approaches consists of an equivalence preserving compilation. In one such approach, the propositional theory $\Sigma$ is compiled into a logical equivalent theory $\Pi(\Sigma)$, the set of prime implicates [CM92, KI92, JP90, KT90, Ng93, SCL70, Str92, Tis67] with respect to which queries are answered by a subsumption test. Another equivalence preserving knowledge compilation consists of transforming the propositional database $\Sigma$ into a logical equivalent database $\text{FPI}(\Sigma)$ (see [dV94]) from which queries are answered by unit refutation. A generalized approach to equivalence preserving knowledge compilation uses the notion of the theory prime implicates [Mar95]. In this approach the theory prime implicates are computed from a knowledge base $\Sigma$ with respect to a tractable theory $Y$, so that queries can be answered from the set of theory prime implicates in polynomial time. Another kind of knowledge compilation deals with an approximation of the knowledge base [dV96], which is not related to the work in this paper.

Most of the research work in knowledge compilation have been restricted to propositional knowledge base in spite of the higher expressing capacity of first order knowledge base. Due to lack of expressive power in propositional theory, first order theory is required to represent knowledge in many problems. We exploit the clausal form of first order theory to store knowledge in a database. The clausal forms are obtained by skolemization and converting the formula into one of conjunctive or disjunctive normal forms. In this paper we assume the first order formulas are in SCNF (Skolem Conjunctive Normal Form) and computation of theory prime implicates is extended to first order logic where theory prime implicates of a knowledge base $X$ with respect to a theory $Y$ is computed. The theory prime implicates are computed using a resolution based algorithm for prime implicates computation for SCNF [RS02].

The paper is organized as follows. In Section 2, we introduce the definitions and notions for establishing the required results. Section 3 describes the properties of theory prime implicates and builds a relation with the set of prime implicates. We give an algorithm to compute the theory prime implicates and prove its partial correctness in Section 4 and Section 5 concludes the paper.
2 Preliminaries

The alphabet of first order language contains the symbols $x, y, z, x_1, y_1, \ldots$, as variables, $f, g, h, f_1, g_1, \ldots$, as function symbols, $P, Q, R, P_1, Q_1, \ldots$, as predicates, $\neg, \lor, \land$ as connectives, $(), \{\}$ as punctuation marks and, $\forall$ and $\exists$ as universal and existential quantifiers. Let $FM$ contain the set of formulas built upon this alphabet. We assume the syntax and semantics of first order logic. Formulas are denoted by upper case letters. For an interpretation or a first order structure $i$ and a formula $X$, we write $i \models X$ if $i$ is a model of $X$. For a set of formulas $\Sigma$ (or a formula) and any formula $X$ we write $\Sigma \models X$ to denote that for every interpretation $i$ if $i$ is a model of every formula in $\Sigma$ then $i$ is a model of $X$. We call $X$ a logical consequence of $\Sigma$. If $X \models Y$ and $Y \models X$ then $X$ and $Y$ are said to be equivalent which is denoted by $X \equiv Y$. The quotient set of $FM$ are said to be equivalent which is denoted by $\equiv$. FM is a finite conjunction of literals. A quantifier-free formula is in conjunctive normal form (CNF) if it is a finite conjunction of literals. A disjunctive clause is a finite disjunction of literals which is also represented as a set of clauses. Each formula is represented as a set of clauses. Two literals $\ell_1, \ell_2$ are said to be complementary to each other with respect to the complementary substitution $\xi$ such that $\ell_1 \equiv \ell_2$. We call $\ell_1$ and $\ell_2$ are two companion literals of $\ell_1$. A disjunctive clause $C$ is an implicate of a finite set of formulas $X$ (assumed to be in SCNF) if $X \sigma \models C$ for a substitution $\sigma$. We write $I(X)$ as the set of all implicates of $X$. A clause $C$ is a prime implicate of $X$ if $C$ is an implicate of $X$ and there is no other implicate $C'$ of $X$ such that $C' \tau \models C$ for a substitution $\tau$ (i.e., if no other implicate $C'$ subsumes $C$). $I(X)$ denotes the set of prime implicates of $X$. It may be observed that if $C$ is not prime then there exists a prime implicate $D$ of $X$ such that $D \tau \models C$. A clause $C' \in I(X)$ is a minimal element of $I(X)$, if for all $C \in I(X)$ and for all $\sigma, C \sigma \models C'$ implies $C \sigma \equiv C'$. Equivalently, a clause $C' \in I(X)$ is a minimal element of $I(X)$ if there is no $C \in I(X)$ and there is no substitution $\sigma$ such that $C \sigma \equiv C'$ and $C \sigma \models C'$. Clearly, the prime implicates of a finite set of formulas $X$ are the minimal elements of $I(X)$ with respect to $\models$. Let $I(X) = min(I_M(X), \models)$.

Let $C_1, C_2$ be two clauses in $X$ and $r \in C_1, s \in C_2$ be a pair of complementary literals with respect to a most general unifier $\sigma$. The consensus of $C_1$ and $C_2$ is $C = CON(C_1, C_2) = [(C_1 - \{r\}) \cup (C_2 - \{s\})] \sigma$ provided that $C$ is fundamental. $C$ can also be written as $[\{C_1 \sigma - \{r\}\} \cup \{C_2 \sigma - \{s\}\}]$ for a literal $t$, provided $r \sigma = t$ and $s \sigma = -t$. If $C$ is the consensus of $C_1$ and $C_2$ with respect to a most general unifier $\sigma$ then $C$ is said to be associated with $\sigma$. By default, each clause $C$ of a set of formulas $X$ is associated with the empty substitution $\epsilon$. Let $C_1$ and $C_2$ be two resolvent clauses associated with substitutions $\sigma_1$ and $\sigma_2$, respectively. Their consensus with respect to $\sigma$ is defined provided $\sigma_1 \sigma = \sigma_2 \sigma$.

Let $Y$ be a finite set of formulas. We define the extension of $\models$ as $\models_Y$ over $FM \times FM$ by $X_1 \models_Y Y_2$ if $\{X_1\} \cup Y \models Y_2$ where $X_1$ and $Y_2$ are two formulas in $FM$. Similarly for a substitution $\sigma$ we define $X_1 \sigma \models_Y X_2$ if $\{X_1 \sigma\} \cup Y \sigma \models X_2$, (i.e., $(X_1 \cup Y) \sigma \models X_2$ if $X_1$ is a set of formulas) where both $X_1$ and $Y$ are associated with the same substitution $\sigma$. If $X_1 \sigma \models_Y X_2$ holds then we say that $X_2$ is a $Y$-logical consequence of $X_1$. We define the equivalence relation $\equiv_Y$ over $FM$ by $X_1 \equiv_Y X_2$ iff $X_1 \sigma_1 \models_Y X_2$ for all $\sigma_1, \sigma_2, \sigma_3$, respectively. Then their consensus with respect to $\sigma$ is defined provided $\sigma_1 \sigma = \sigma_2 \sigma$.

The definition of prime implicate is extended to the theory prime implicates as follows. Let $X$ and $Y$ be finite sets of formulas. A clause $C$ is a theory implicate of $X$ with respect to $Y$ iff $X \models_Y C$. A clause $C$ is called a theory prime implicate of $X$ with respect to $Y$ if $C$ is a theory implicate of $X$ with respect to $Y$ and there is no other theory implicate $C'$ such that $C' \models_Y C'$ for some substitution $\tau$. We denote $\Theta(X, Y)$ as the set of theory prime implicates of $X$ with respect to $Y$. Thus, $\Theta(X, Y)$ contains the minimal elements of the set of theory implicates of $X$.
with respect to $Y$. The minimal clauses are considered up to $Y$-logical equivalence, i.e., $\Theta(X,Y)$ contains a clause from each equivalence class.

3 Properties

We describe some of the properties of theory prime implicates of $X$ with respect to $Y$. These results lead us to the computation of the set of theory prime implicates $\Theta(X,Y)$ from the set of prime implicates $\Pi(X)$ of $X$.

Lemma 3.1. Let $X$ and $Y$ be finite sets of formulas. Then $\Theta(X,Y) \subseteq \Pi(X \cup Y)$.

Proof: Let $C \in \Theta(X,Y)$. So $X \sigma \vdash_Y C$ holds and there is no theory implicate $C'$ of $X$ with respect to $Y$ such that $C' \tau \vdash_Y C$ holds for some $\tau$. This implies $X \sigma \cup Y \tau \vdash C$, i.e., $Y \tau \vdash \neg \sigma \rightarrow \tau \rightarrow C$. So $C$ is not an implicate of $X \cup Y$. If $C$ is not a prime implicate of $X \cup Y$ then there exists a prime implicate $C'$ of $X \cup Y$ such that $C' \tau \vdash C$, i.e., $C' \tau \vdash_Y C$. As $C'$ is a prime implicate of $X \cup Y$, $(X \cup Y) \tau \vdash C'$, i.e., $X \tau \vdash_{Y'} C'$. This implies $C'$ is a theory implicate of $X$ with respect to $Y$. So we get a theory implicate $C'$ such that $C' \tau \vdash_Y C$, i.e., $C$ is not a theory prime implicate of $X$ with respect to $Y$, which is a contradiction.

Lemma 3.2 If $C, C' \in \Pi(X \cup Y)$ and $C \tau \not\vdash_{Y} C'$ for some $\tau$ then $C' \not\in \Theta(X,Y)$.

Proof: Let $C' \in \Theta(X,Y)$. $X \sigma \vdash_Y C'$ and there is no theory implicate $C''$ such that $C'' \tau \vdash_Y C'$ for some $\tau$. In other words, $(X \cup Y) \sigma \vdash C'$ and there is no theory implicate $C''$ (i.e., $(X \cup Y) \tau \vdash C''$) such that $C'' \tau \vdash_Y C'$. Thus, $(X \cup Y) \sigma \vdash C'$ and for every $C''$ if $(X \cup Y) \tau \vdash C''$ then $C'' \not\vdash_{Y'} C'$. As $C$ is a prime implicate of $X \cup Y$, $C$ is also an implicate of $X \cup Y$. With $C'' = C$, $(X \cup Y) \sigma \vdash C'$ and if $(X \cup Y) \tau \vdash C$ then $C \not\vdash_{Y'} C'$, which contradicts the hypothesis of the Lemma. This implies that $C' \not\in \Theta(X,Y)$, completing the proof.

Lemma 3.3 If $C, C' \in \Theta(X,Y)$ then either $\sigma \vdash_Y C'$ for all $\sigma$ and $C' \not\vdash_{Y} C$ for all $\sigma_2$, or $C \not\vdash_{Y'} C'$ for all $\sigma_2$ and $C \not\vdash_{Y} C$ for all $\sigma_1$.

Proof: As $\sigma \vdash_Y C'$ for all $\sigma$, $C'$ is a $Y$-logical consequence of $C$. If $C \not\vdash_{Y'} C$ for all $\sigma_1$ then $C$ is a $Y$-logical consequence of $C$. So $C \not\vdash_Y C$. Thus either $C$ or $C'$ belongs to $\Theta(X,Y)$ but not both which is a contradiction to the hypothesis. Similarly we can prove the other part. This completes the proof.

The following result is obtained by using Lemmas 3.1 – 3.3.

Theorem 3.1 $\Theta(X,Y) = \min(\Pi(X \cup Y), [\vdash_Y])$.

Theorem 3.2 Let $X$ and $Y$ be finite sets of formulas and $C$ be any clause. $X \sigma \vdash_Y C$ holds iff there exists a theory prime implicate $C'$ of $X$ with respect to $Y$ such that $C' \tau \vdash_Y C$ holds.

Proof: Suppose $X \sigma \vdash_Y C$ holds, i.e., $(X \cup Y) \sigma \vdash C$. So $C$ is an implicate of $X \cup Y$. If $C$ is not prime then there is an implicate $C''(\neq C)$ of $X \cup Y$ and a substitution $\tau$ such that $C'' \tau \vdash C$. Let $A'' = \{C'' : C'' \not\vdash_X C\}$ is an implicate of $X \cup Y$ and $C'' \tau \vdash_Y C$ for some $\tau$. We can obtain a finite subset $A = \{c_1, \ldots, c_n\}$ of $A''$ such that for each $C'' \in A''$ there is a $C_i \in A$ and a substitution $\eta$ such that $C'' = C_i \eta$ since there are only a finite number of variables in $C''$ and $C$ is finite. Now $A$, being a finite set, has a strict partial order as logical consequence with respect to $Y$. Each element of $A$ is an implicate of $X \cup Y$. Any minimal element $C'$ of $A$ is a prime implicate of $X \cup Y$, i.e., a theory prime implicate of $X$ with respect to $Y$.

Conversely, there exists a prime implicate $C'$ of $X \cup Y$ such that $C' \tau \vdash_Y C$, i.e., $C' \tau \cup Y \tau \vdash C$ holds. As $(X \cup Y) \sigma \vdash C'$ and $C' \vdash C' \tau$, $(X \cup Y) \sigma \vdash Y \tau \cup C' \tau$, since $C' \tau$ and $Y \tau$ are disjunctive. Thus, $(X \cup Y) \sigma \vdash C$, showing that $X \sigma \vdash_Y C$. This completes the proof.

Lemma 3.4 If $X \equiv X'$ then $\Pi(X) = \Pi(X')$.

Proof: Let $C \in \Pi(X)$, i.e., $X \sigma \vdash C$ and there exists no implicate $C'$ of $X$ such that $C' \sigma \vdash C$. Since $X \equiv X'$, $X' \sigma \vdash C$ and there exists no implicate $C'$ such that $C' \sigma \vdash C$. This implies $C \in \Pi(X')$. Thus, $\Pi(X) \subseteq \Pi(X')$. Similarly, $\Pi(X') \subseteq \Pi(X)$.

Theorem 3.3 If $X \equiv_Y X'$ and $Y \equiv_Y Y'$ then $\Theta(X', Y') = \Theta(X,Y)$, upto $Y$-equivalence.

Proof: Let $C \in \Theta(X', Y')$, i.e., for all $C' \in \Pi(X' \cup Y')$, if $C' \sigma \vdash_{Y'} C$ then $\sigma C \equiv_{Y'} C$. By hypothesis, $X' \cup Y' \equiv X \cup Y'$; by Lemma 3.4, $\Pi(X' \cup Y') = \Pi(X \cup Y)$. So, for all $C' \in \Pi(X \cup Y)$ if $C' \sigma \vdash_{Y'} C$ then...
$C' \sigma \equiv_Y C$. As $C' \in \Theta(X,Y)$, $\Theta(X',Y') \subseteq \Theta(X,Y)$. Similarly, the other inclusion $\Theta(X,Y) \subseteq \Theta(X',Y')$ follows.

Let $X$ and $Y$ be finite sets of formulas and $Z_1 = X \cup Y$. Define $L^i(Z_1) = Z_1 \cup \{C^i : C^1 \in \text{the consensus of a pair of clauses from } Z_1\}$. Construct $Z_2$ by deleting those clauses from $L^i(Z_1)$ which are Y-logical consequences of $C^1$. $L^2(Z_1) = Z_2 \cup \{C^2 : C^2 \in \text{the consensus of a pair of clauses from } Z_2\}$. Construct $Z_3$ like $Z_2$, but from $L^2(Z_1)$. Similarly, write $L^n(Z_1) = Z_n \cup \{C^n : C^n \in \text{the consensus of two clauses from } Z_n\}$. At one stage for some $m > n$, $L^m(Z_1) = L^n(Z_1) = Z_{n+1}$, happens in the propositional case. Unfortunately this need not be so in a first order knowledge base. This is because, the process may not terminate for some inputs. This goes along well with the undecidability of first order logic and the best we may hope is a partial correctness of the algorithm. See the following example of transitivity axioms.

**Example 3.1** Let $Z_1 = \{\neg Pxy \lor \neg Pyz \lor Pxz, \neg Pst \lor \neg Ptu \lor \neg Puw \lor Psx\}$. The consensus closure of both the clauses is infinite. $L(Z_1) = \{\neg Pxy \lor \neg Pyz \lor Pxz, \neg Pst \lor \neg Ptu \lor \neg Puw \lor Psx\}$. The consensus closure of both the clauses is infinite. $L(Z_1) = \{\neg Pxy \lor \neg Pyz \lor Pxz, \neg Pst \lor \neg Ptu \lor \neg Puw \lor Psx\}$. We can see there is no $m > n$ such that $L^m(Z_1) = L^n(Z_1)$, i.e., none of the clauses subsumes any of the others. Hence the process does not terminate for transitivity axioms as input.

However each clause in every set $L^n(X \cup Y)$ is a Y-logical consequence of $X$, as the following result shows.

**Lemma 3.5** $X \models_Y L^n(X \cup Y)$.

**Proof:** First, we show that $X \cup Y \models L^i(X \cup Y)$. Let $i$ be a model of $X \cup Y$. $i \models C$ for all $C \in X \cup Y$. If $X \cup Y = L^i(X \cup Y)$ then it is through. If not, there exists a clause $D \in L^i(X \cup Y)$ such that $D \models \text{the consensus of two clauses } C_j \text{ and } C_k \text{ of } X \cup Y$. As $i \models C_j \land C_k, \text{so } i \models D, \text{thus } i \models L^i(X \cup Y)$. This implies $X \cup Y \models L^i(X \cup Y)$, i.e., $X \models_Y L^i(X \cup Y)$. Similarly we can show that $L^i(X \cup Y) \models_Y L^j(X \cup Y) \models_Y \ldots \models_Y L^n(X \cup Y)$. By induction, it follows that $X \models_Y L^n(X \cup Y)$.

This implication can be strengthened to equivalence as in the following.

**Theorem 3.4** $X \equiv_Y L^n(X \cup Y)$.

**Proof:** Due to Lemma 3.5, we only show that $L^n(X \cup Y) \models_Y X$. For this, let $i$ be a model of $L^i(X \cup Y)$ for all $C \in L^i(X \cup Y)$ if $L^i(X \cup Y)$ holds for some $m$, $\Theta(X,Y)$ is finite as $L^n(X \cup Y)$ is finite, for each $n$.
Definition 3.1 Let $X$ and $Y$ be finite sets of formulas such that $X \models Y$. The theory prime implicate compilation of $X$ with respect to $Y$ is defined by $\Omega_Y(X) = \Theta(X, Y) \cup Y$.

Theorem 3.7 $\Omega_Y(X) \equiv X$, i.e., $\Theta(X, Y) \cup Y \equiv X$, provided $\Theta(X, Y)$ is finite.

Proof: By Theorems 3.4 and 3.6, $\Theta(X, Y) \cup Y \models X$. Conversely, let $i$ be a model of $X$. Since $X \models Y$ (as implicitly assumed in Definition 3.1), $X$ entails each clause obtained by taking consensus of any pair of clauses from $X \cup Y$, i.e., $X \models L^n(X \cup Y)$. Thus, $X \models \Theta(X, Y)$, again by Theorem 3.6, showing that $i \models \Theta(X, Y) \cup Y$. This implies $X \models \Theta(X, Y) \cup Y$. □

4 Computing Theory Prime Implicates

We compute the theory prime implicates $\Theta$ of a set of formulas $X$ with respect to $Y$ by computing the implicates of $X \cup Y$. The latter computation uses a resolution based primitive implicate algorithm [KT90, Ki92, RS02] and only the representatives of $Y$-logical equivalent clauses are kept in the set $\Theta$.

Algorithm TPI

Input: $X \cup Y$, set of clauses in $X$ and $Y$
Output: $\Theta(X, Y)$, theory prime implicates of $X$ with respect to $Y$
begin
$\Theta := X \cup Y$;
if $X \cup Y := \emptyset$
$\Theta := \emptyset$;
else
$Z_0 := \emptyset$;
$Z_1 := \Theta$;
i := 1;
while $Z_i \neq Z_{i-1}$
do
compute $L^i(Z_1)$;
compute $Z_{i+1}$;
i := i + 1;
end
$\Theta := Z_{i+1}$;
end
return $\Theta(X, Y)$;
end

In the above algorithm we compute the set of theory prime implicates of $X$ with respect to $Y$ by computing the prime implicates of $X \cup Y$ and keeping only the $Y$-logical equivalent clauses in the set. We know apart from the set of clauses $Z_i$, that $L^i(Z_1)$ contains the consensus $CON$ of a pair of clauses from $Z_i$, $Z_{i+1}$ contains the clauses remaining after discarding the $Y$-logical consequence of $CON$. The algorithm runs till $L^n(Z_1) = Z_{n+1} = L^{n+1}(Z_1)$ holds. The set $L^n(Z_1)$ contains the set of theory prime implicates. Note there is no guarantee that the algorithm really terminates.

Theorem 4.1 The algorithm TPI correctly computes the set of theory prime implicates of an SCNF formula, provided it terminates.

Proof: Let $X \cup Y$ be the given set of clauses. If the set $X \cup Y$ is empty, then obviously there are no theory prime implicates. If $X \cup Y$ is non-empty, assign $X \cup Y$ to $Z_1$. Then, $L^1(Z_1)$ is computed. Those clauses $D$ of $L^1(Z_1)$ will be discarded from the set which are $Y$-logical consequences of the newly added clauses $CON$ to get $Z_2$ because anything derivable from $D$ can be derived from $CON$. By Theorem 3.1 and Theorem 3.6, $L^1(Z_1) \subseteq \Pi(X \cup Y)$. Similarly, $L^n(Z_1)$ is computed and for some $m > n$, $L^n(Z_1) = L^{m}(Z_1)$ due to the assumption that TPI terminates. Hence all the minimal elements of $\Pi(X \cup Y)$ has been computed in $L^n(Z_1)$. So by Theorem 3.1, $\Theta(X, Y)$ has been computed. □

We remark that partial correctness of TPI is the best possible. In fact any such algorithm can not be totally correct due to the undecidability of first order logic.

5 Conclusion

In this paper, the notion of prime implicates is generalized to the theory prime implicates and an algorithm for computing the theory prime implicates has been discussed. The algorithm computes the set of theory prime implicates $\Theta(X, Y)$ of a knowledge base $X$ with respect another knowledge base $Y$. It is only partially correct as its termination can not be guaranteed, in general.

Since the compilation takes a long time to obtain $\Theta$, it is desirable to ask queries at any time while the compilation goes on. Though all the queries can not be answered before the off-line computation is completed, the possibility of answering the number of queries increases. The off-line computation could be avoided partially but how it can be done efficiently is not yet known.
The size of the compilation is exponential in the size of the original knowledge base. If we consider $Y = \emptyset$, then the theory prime implicate compilation coincides with the prime implicates compilation.

References


