Machine Learning 26, 213–226 (1997)

Generalization of Clauses Relative to a Theory

PETER IDESTAM-ALMQUIST pi@dsv.su.se
Department of Computer and Systems Sciences, Stockholm University, Electrum 230, S-164 40 Kista, Sweden

Editors: Stephen Muggleton and David Page

Abstract. Plotkin's notions of relative θ-subsumption and relative least general generalization of clauses are defined for full clauses, and they are defined in terms of a kind of resolution derivations called C-derivations. Techniques for generalization of clauses relative to a theory, based on the V-operators or saturation in its original form, have primarily been developed for Horn clauses. We show that these techniques are incomplete for full clauses, which is due to the restricted form of resolution derivations considered. We describe a technique for generalization of clauses relative to a theory, which is based on a generalization of the original saturation technique. We prove that our technique properly inverts C-derivations, and that it is complete for full clauses w.r.t. relative θ-subsumption.

Keywords: relative generalization, RLGG, full clauses

1. Introduction

Inductive logic programming (ILP) is a research area for inductive learning in first-order logic. The representation formalism usually used is Horn clauses. Every first-order formula can be represented by a set of full clauses that is equivalent to the formula w.r.t. satisfiability, but this is not the case for Horn clauses. It is therefore of interest to study inductive learning in a full clause representation language. Generalization is a main operation in inductive learning, and the use of background knowledge is crucial in inductive learning. Plotkin's notions of relative θ-subsumption and relative least general generalization w.r.t. a theory (background knowledge) are defined for full clauses, and they are defined in terms of a kind of resolution derivations called C-derivations (Plotkin, 1971a, b).

Techniques for generalization of clauses relative to a theory, based on the V-operators (Muggleton & Buntine, 1988; Muggleton, 1991) or saturation in its original form (Sammut, 1981; Rouveirol, 1990, 1992), have primarily been developed for Horn clauses. These techniques are incomplete for full clauses, which is due to the restricted form of resolution considered. Saturation in its original form is based on a procedure called elementary saturation (Rouveirol, 1992), which can be regarded as a V-operator. The V-operators invert what has been called "linear derivations", but in the resolution literature is known as input (resolution) derivations (Genesereth & Nilsson, 1987). More precisely techniques based on the V-operators invert input derivations in which the starting clause is used only once. We call this specific form of input derivations, C-start input derivations. In Section 3, we show that C-start input derivations and C-derivations are not equivalent w.r.t. derivability.

We assume the reader to be familiar with the basic notions and notations in Logic Programming (Lloyd, 1987) and/or Automatic Theorem Proving (Chang & Lee, 1973; Gallier, 1986). In the study of relative generalization of clauses it is convenient to generalize the concept of a clause. Usually a clause is defined as a finite set of literals. In the following we will let a *clause* denote a possibly infinite set of literals. Sometimes we say finite clause, possibly infinite clause or infinite clause to explicitly state if we mean a finite, possibly infinite or infinite set of literals. A *Horn clause* is a clause including at most one positive literal, and a *definite clause* is a clause including exactly one positive literal. When we say *full clauses* it is to emphasize that we mean clauses and not Horn clauses or definite clauses.

Both the techniques based on V-operators (see Section 3) and our technique (see Section 4) for generalization relative to a theory make an assumption on the clauses in the considered theories. We call this assumption the *variable assumption*, and it means that every variable in a clause occurs in at least two distinct literals in that clause. This should also be true for all clauses in the resolution derivations we consider. Theories that fulfil the variable assumption are in (Muggleton, 1991) called *strongly generative*.

The variable assumption may seem as a strong restriction, but every clause can be transformed into an equivalent clause that fulfils the assumption. This can simply be done by adding one literal $term(x)$ to a clause for each variable x that only occurs in one literal in that clause. Then to the background theory we also add one clause $(term(f(x_1, \ldots, x_n)) \leftarrow term(x_1), \ldots, term(x_n))$ for each n-ary function symbol f in the considered clausal language. Thus for each constant c (0-ary function symbol) we add a clause $(term(c) \leftarrow)$.

If we have a theory that fulfils the variable assumption (a strongly generative theory), we are not guaranteed that all clauses derivable by resolution from the theory fulfil the variable assumption. This is due to that a factor of a clause not necessarily fulfils the variable assumption although the clause does it. For this reason it is not enough to require the considered theory to fulfil the variable assumption, we also require all clauses in the resolution derivations we consider to fulfil the variable assumption. Therefore the kind of clause transformation described above, sometimes also needs to be performed on clauses obtained in the resolution derivation process.

Plotkin's framework for generalization of clauses relative to a theory is described in Section 2. In Section 3, we describe the techniques for generalization relative to a theory based on the V-operators, and show incompleteness of C-start input derivation. In Section 4, we present our technique for generalization of clauses relative to a theory, which is based on a generalization of the original saturation technique. We prove that our technique properly inverts C-derivations, and that it is complete for full clauses w.r.t. relative θ-subsumption. In Section 5, we summarize and discuss our results.

2. Generalization relative to a theory

There is a well-known framework for generalization of clauses developed by Plotkin (1970, 1971a, b). This framework is based on two relations between clauses called θ-subsumption and relative θ-subsumption, where relative θ-subsumption takes into account a *theory* (a finite set of finite clauses) of background knowledge. We first give the definition of θ-subsumption, and the related notion of least general generalization.

Definition. Let C and E be clauses. Then C *θ-subsumes* E, denoted $C \preceq E$, if and only if there exists a substitution θ such that $C\theta \subseteq E$. If $C \preceq E$ then we say that C is a *θ-subsumer* of E.

Definition. A clause C is a *generalization* of a set of clauses $S = \{E_1, \ldots, E_n\}$ if and only if, for every $1 \leq i \leq n$, $C \preceq E_i$. A generalization C of S is a *least general generalization* (LGG) of S if and only if, for every generalization C' of S, $C' \preceq C$.

Plotkin showed that for every finite set of finite clauses there exists a unique LGG (up to equivalence). This can be generalized to that for every finite set of possibly infinite clauses there exists a possibly infinite LGG. There is a well-known algorithm for computing LGGs, which was described by Plotkin (1971) and is frequently used in ILP.

Relative θ-subsumption is defined in terms of θ-subsumption and a particular kind of resolution derivations called C-derivations. Before we give a formal definition of C-derivations, we need some definitions concerning resolution.

Definition. Let C be a clause, $\Gamma \subseteq C$ and γ an mgu of Γ. Then $C\gamma$ is a *factor* of C.

Definition. A clause R is a *resolvent* of two clauses C and D if and only if there are $C\gamma$, $D\mu$, A, B and θ such that:

a) $C\gamma$ is a factor of C and $D\mu$ is a factor of D,
b) $C\gamma$ and $D\mu$ have no variables in common,
c) A is a literal in $C\gamma$ and B is a literal in $D\mu$,
d) θ is an mgu of $\{A, \bar{B}\}$, and
e) R is the clause $((C\gamma - \{A\}) \cup (D\mu - \{B\}))\theta$.

The clauses C and D are called *parent clauses* of R.

Definition. A *resolution derivation* of a clause R from a theory T is a structure (a binary tree) recursively defined as follows:

a) $(-, -, R)$ is a resolution derivation of R from T if and only if $R \in T$, and
b) $((S_1, S_2, D), (S'_1, S'_2, E), R)$ is a resolution derivation of R from T if and only if (S_1, S_2, D) and (S'_1, S'_2, E) are resolution derivations of D and E from T, and R is a resolvent of D and E.

The *length* of a resolution derivation is the number of clauses included in the resolution derivation.

We write $T \vdash_{\mathcal{R}} R$ to denote that there is a resolution derivation of a clause R from a theory T. We also write $T \vdash_{\mathcal{R}} \{R_1, \ldots, R_n\}$ when, for every $1 \leq i \leq n$, there is a resolution derivation of R_i from T.

Definition. A *C-derivation* of a clause R from a clause C and a theory T is a specific kind of resolution derivation recursively defined as follows:

a) $(-, -, R)$ is a C-derivation of R from C and T if and only if $R = C$, and
b) $((S_1, S_2, D), (S_1', S_2', E), R)$ is a C-derivation of R from C and T if and only if R is a resolvent of D and E, and either:
   i) (S_1, S_2, D) is a resolution derivation of D from T, and (S_1', S_2', E) is a C-derivation of E from C and T, or
   ii) (S_1, S_2, D) is a C-derivation of D from C and T, and (S_1', S_2', E) is a resolution derivation of E from T.

We write $(C, T) \vdash_C R$ to denote that there is a C-derivation of a clause R from a clause C and a theory T. A C-derivation is a resolution derivation in which a specific clause is used exactly once. This definition slightly differs from Plotkin's definition, in which a resolution derivation is a C-derivation if and only if the specified clause is used *at most* once.

Definition. A clause C *θ-subsumes* a clause E *relative* to a theory T, denoted $C \preceq_T E$, if and only if there exists a C-derivation of a clause R from C and T such that $R \preceq E$. If $C \preceq_T E$ then we say that C is a *relative θ-subsumer* of E w.r.t. T.

Our definition is equivalent to Plotkin's original definition except for clauses that logically follow from only the theory. Strictly following Plotkin, any clause is a relative θ-subsumer of a clause that logically follows from a theory. In our opinion, our definitions of relative θ-subsumption and relative least general generalization (see below) are more analogous to the definitions of θ-subsumption and least general generalization. However, the difference only occurs for a kind of clauses usually not considered in inductive learning.

Definition. A clause C is a *relative generalization* of a set of clauses $S = \{E_1, \ldots, E_n\}$ w.r.t. a theory T if and only if, for every $1 \leq i \leq n$, $C \preceq_T E_i$. A relative generalization C of S w.r.t. T is a *relative least general generalization* (RLGG) of S w.r.t. T if and only if, for every relative generalization C' of S w.r.t. T, $C' \preceq C$.

Example. Consider the following clauses:

$$
\begin{aligned}
C &= (r(x) \leftarrow p(x), q(x)),\\
D_1 &= (p(x), q(x) \leftarrow s(x)),\\
D_2 &= (p(x) \leftarrow q(x)),\\
D_3 &= (q(x) \leftarrow p(x)),\\
R_1 &= (p(x) \leftarrow s(x)),\\
R_2 &= (r(x) \leftarrow p(x)),\\
R_3 &= (r(x) \leftarrow s(x)),\\
R_4 &= (r(x) \leftarrow q(x)),\\
E_1 &= (r(a) \leftarrow s(a)), \text{ and}\\
E_2 &= (r(b) \leftarrow p(b)).
\end{aligned}
$$

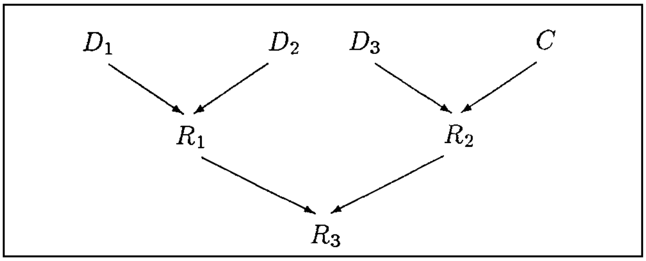

Figure 1. A C-derivation of R_3 from C and $\{D_1, D_2, D_3\}$.

Let $T = \{D_1, D_2, D_3\}$ be a theory. Then C θ-subsumes E_1 relative to T, since there exists a C-derivation (of length 7) of R_3 from C and T (see figure 1), namely

$$(((-, -, D_1), (-, -, D_2), R_1), ((-, -, D_3), (-, -, C), R_2), R_3),$$

and $R_3 \preceq E_1$. The clause C also θ-subsumes E_2 relative to T, since there exists a C-derivation (of length 3) of R_4 from C and T, namely

$$((-, -, D_3), (-, -, C), R_4),$$

and $R_4 \preceq E_2$. In fact C is an RLGG of $\{E_1, E_2\}$ w.r.t. T. □

In general there exists no finite RLGG of a finite set of finite clauses w.r.t. a theory. However, for every such set of clauses there exists a possibly infinite RLGG, and that is the reason why we generalized the concept of a clause to be a possibly infinite set of literals.

3. Relative generalization using V-operators

Inverting resolution derivations is a technique for learning missing clauses in a theory (Muggleton & Buntine, 1988; Rouveirol & Puget, 1989; Wirth, 1989; Rouveirol, 1990; Hume & Sammut, 1991; Idestam-Almquist, 1992), and it has strong connections to the definitions of relative θ-subsumption and RLGG (Muggleton, 1991). If we know the resolvent and one of the parent clauses in a resolution step, a V-operator can derive the other parent clause. Two such V-operators for first-order clauses are presented in (Muggleton & Buntine, 1988). To restrict the number of alternatives in the construction of the unknown parent clause, these V-operators used a number of different assumptions. A better way to restrict the number of alternatives is to only construct maximally specific alternatives of the unknown parent clause. Most specific V-operators and a function $\mathcal{V}^n$, which describes the set of maximally specific clauses that can be constructed by iterative application of the V-operators, are presented in (Muggleton, 1991).

Definition. Let E be a clause and T a theory. Then the function $\mathcal{V}^n(T, E)$ is recursively defined as follows:

a) $\mathcal{V}^0(T, E) = \{E\}$, and
b) $\mathcal{V}^n(T, E) = \mathcal{V}^{n-1}(T, E) \cup \{(F \cup \{\bar{L}\}\theta) \mid L \in D \in T$ and $F \in \mathcal{V}^{n-1}(T, E)$ and $(D - \{L\})\theta \subseteq F\}$ $(n > 0)$.

The closure $\mathcal{V}^*(T, E) = \mathcal{V}^0(T, E) \cup \mathcal{V}^1(T, E) \cup \cdots$

Note that the clause $(F \cup \{\bar{L}\}\theta)$ is only completely determined under the variable assumption.

Example. Consider the following clauses:

$D = (\, p \leftarrow q(x)\,)$, and
$E = (\, p \leftarrow)$.

The variable assumption does not hold, since the variable x only occurs in one literal in D. Let E be a resolvent of D and an unknown clause C. Then the most specific alternative C' of C is a clause $(E \cup \{\bar{L}\}\theta)$ such that $L \in D$ and $(D - \{L\})\theta \subseteq E$. We have $C' = (\,p, q(x) \leftarrow)\theta$ for any ground substitution θ. Thus C' can not be completely determined.

Saturation was originally presented in Rouveirol (1990) and further developed in Rouveirol (1992). Elementary saturation can be regarded as a V-operator, and then the process of saturation is included in the function $\mathcal{V}^n$.

The function $\mathcal{V}^n$ inverts a sequence of n resolution steps. Theorem 1 below is given in Muggleton (1991), and it describes a relationship between an RLGG of a set of Horn clauses w.r.t. a logic program and an LGG of a set of clauses obtained from the Horn clauses by the closure $\mathcal{V}^*$ of the function $\mathcal{V}^n$. A *logic program* is a finite set of finite definite clauses. Let $\cup\mathcal{V}^n(T, E)$ denote the clause formed by taking the union of the clauses in $\mathcal{V}^n(T, E)$.

Theorem 1 (Relationship between RLGG and LGG: Horn clauses). *Let P be a logic program, and let $\{H_1, \ldots, H_n\}$ be a set of Horn clauses. Then* (*under the variable assumption*) *an LGG of the set of clauses $\{\cup\mathcal{V}^*(P, H_1), \ldots, \cup\mathcal{V}^*(P, H_n)\}$ is an RLGG of $\{H_1, \ldots, H_n\}$. w.r.t. P.*

This theorem only consider Horn clauses, and it should be observed that an RLGG of two Horn clauses w.r.t. a logic program not necessarily is a Horn clause. Even when H_1 and H_2 are finite, the clauses $\cup\mathcal{V}^*(P, H_1)$ and $\cup\mathcal{V}^*(P, H_2)$ may be infinite, and thus the RLGG of H_1 and H_2 may also be infinite. However, if we replace $\mathcal{V}^*$ with $\mathcal{V}^k$, for some natural number k, we have a technique for computing approximate RLGGs, as described in Algorithm 1.

Algorithm 1 (Computation of approximate RLGGs: Horn clauses).
Input: *A finite set of finite Horn clauses $\{H_1, \ldots, H_n\}$, a logic program P, and a natural number k.*
Output: *An approximate RLGG of $\{H_1, \ldots, H_n\}$ w.r.t. P.*
1) *For every $1 \le i \le n$, compute $G_i = \cup\mathcal{V}^k(P, H_i)$.*
2) *Compute and return the LGG of $\{G_1, \ldots, G_n\}$.*

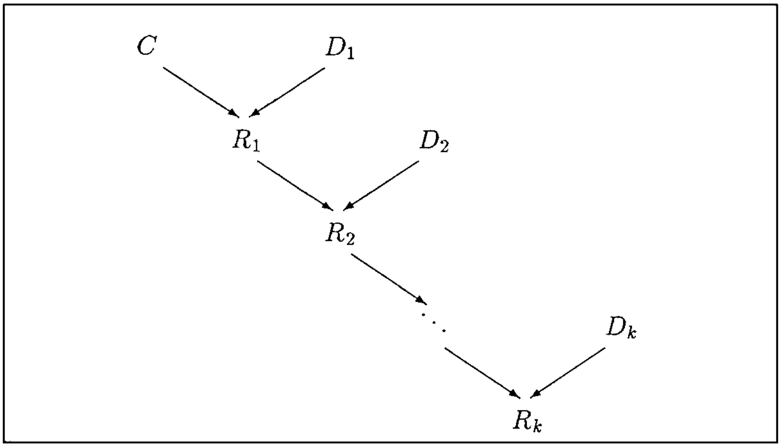

Figure 2. A C-start input derivation of R_k from C and $\{D_1, \ldots, D_k\}$.

Theorem 1 can not be extended to full clauses, because the function $\mathcal{V}^n$ only inverts resolution derivations, where in each resolution step one of the parent clauses is a clause in the theory. This specific kind of resolution derivations are called "linear derivations" in Muggleton (1991), but in the resolution literature it is known as input (resolution) derivations (Genesereth and Nilsson, 1987). More precisely the function $\mathcal{V}^n$ inverts input derivations where the starting clause is used only once. We call such input derivations, C-start input derivations.

Definition. A *C-start input derivation* of a clause R from a clause C and theory T is a specific form of resolution derivation recursively defined as follows:

a) $(-, -, R)$ is a C-start input derivation of R from C and T if and only if $R = C$, and
b) $((S_1, S_2, D), (-, -, E), R)$ is a C-start input derivation of R from C and T if and only if (S_1, S_2, D) is a C-start input derivation of D from C and T, $E \in T$ and R is a resolvent of D and E.

A C-start input derivations can graphically be described as in figure 2. Every C-start input derivation is a C-derivation, but not the converse. For example the C-derivation in figure 1 is not a C-start input derivation. More important to notify is that C-start input derivations and C-derivations are not equivalent w.r.t. derivability. In Theorem 2, we show that there are C-derivations for which there are no corresponding C-start input derivations. It follows from this theorem that techniques for generalization relative to a theory, based on the V-operators or saturation in its original form, are not complete for full clauses. For example the RLGG in the example on page 5 can not be found by such techniques.

Theorem 2 (Incompleteness of C-start input derivation). *There exist clauses C and R, and a theory T, such that there exists a C-derivation of R from C and T, but there exists no C-start input derivation of R from C and T.*

Proof: Let $C = \{p, q\}$, $D_1 = \{p, \neg q\}$, $D_2 = \{\neg p, q\}$, $D_3 = \{r, \neg p, \neg q\}$, $T = \{D_1, D_2, D_3\}$ and $R = \{r\}$. Then there exists a C-derivation of R from C and T, namely $(((C, -, -), (D_1, -, -), R_1), ((D_2, -, -), (D_3, -, -), R_2), R)$, where $R_1 = \{p\}$ and $R_2 = \{r, \neg p\}$.

The clause R includes none of the literals $p, \neg p, q$ or $\neg q$. However, every clause derivable by C-start input derivation from C and T includes at least one of these literals. If we resolve away a p or $\neg p$ by using a clause in T then the resolvent will include either q or $\neg q$, and vice versa. Consequently, there exists no C-start input derivation of R from C and T. □

4. Relative generalization using complementation and resolution

In this section we present a technique for inverting C-derivations and computing RLGGs, which is complete for full clauses. The technique is based on a generalization of the original saturation technique (Rouveirol, 1990, 1992), and the saturations of the considered clauses are computed by using complementation and resolution.

The complement of a formula is the negation of the formula. Since all variables in a clause implicitly are universally quantified, all variables in the complement of a clause should be existentially quantified. However, existentially quantified variables can not be expressed in a clausal language. We therefore replace each existentially quantified variable by a new constant (a Skolem constant). Then we obtain a formula that can be represented by a set of clauses, and that is equivalent to the original formula w.r.t. satisfiability.

Definition. The *complement* $\bar{A}$ of a positive literal A is $\neg\bar{A}$, and the *complement* $\neg A$ of a negative literal $\neg A$ is A.

Definition. Let σ be a substitution, C be a clause, T a set of clauses and F the set of function symbols occurring in $T \cup \{C\}$. Then σ is a *Skolem substitution* for C w.r.t. T if and only if $\{x_1/a_1, \ldots, x_k/a_k\} \subseteq \sigma$ where $\{x_1, \ldots, x_k\}$ is the set of variables occurring in C, $a_1, \ldots, a_k$ are distinct constants, and $F \cap \{a_1, \ldots, a_k\} = \emptyset$.

Definition. Let $C = \{L_1, \ldots, L_m\}$ be a clause, T a set of clauses, and $\sigma = \{x_1/a_1, \ldots, x_k/a_k\}$ a Skolem substitution for C w.r.t. T. Then the set of ground unit clauses $\bar{C} = \{\{\overline{L_1}\}\sigma, \ldots, \{\overline{L_m}\}\sigma\}$, is the *complement* of C by σ w.r.t. T.

Definition. Let C be a clause, T a set of clauses, $\sigma = \{x_1/a_1, \ldots, x_k/a_k\}$ a Skolem substitution for C w.r.t. T, and $\bar{D}$ a set of ground unit clauses derivable by resolution from $T \cup \bar{C}$, where $\bar{C}$ is the complement of C by σ w.r.t. T. Then $\sigma^{-1} = \{a_1/x_1, \ldots, a_k/x_k\}$ is an *inverse Skolem substitution* for $\bar{D}$ w.r.t. $T \cup \bar{C}$.

Definition. Let D be a clause, and $\sigma^{-1} = \{a_1/x_1, \ldots, a_k/x_k\}$ an inverse Skolem substitution. Then $D\sigma^{-1}$, the *anti instance* of D by σ^{-1}, is the clause obtained from D by for every $1 \leq i \leq k$, simultaneously replacing each occurrence of the constant a_i in D by the variable x_i.

Definition. Let $\bar{D} = \{\{L_1\}, \ldots, \{L_m\}\}$ be a set of ground unit clauses, T a set of clauses, and $\sigma^{-1} = \{a_1/x_1, \ldots, a_k/x_k\}$ an inverse Skolem substitution for $\bar{D}$ w.r.t. T. Then the clause $D = \{\overline{L_1}, \ldots, \overline{L_m}\}\sigma^{-1}$ is the *complement* of $\bar{D}$ by σ^{-1} w.r.t. T.

Example. Consider the following clause C and set of clauses $\bar{C}$:

$$C = (\,p(x) \leftarrow q(x), r(a)\,), \text{ and}$$
$$\bar{C} = \{(\leftarrow p(b)\,), (\,q(b) \leftarrow), (\,r(a) \leftarrow)\}.$$

Then the substitution $\sigma = \{x/b\}$ is a Skolem substitution for C, and the set of clauses $\bar{C}$ is the complement of C by σ. We also have that $\sigma^{-1} = \{b/x\}$ is an inverse Skolem substitution for $\bar{C}$, and that C is the complement of $\bar{C}$ by σ^{-1}.

When inverting a C-derivation, we know a clause R that follows by C-derivation from an unknown clause C and a known theory T. A least general alternative for the clause C can then be found by adding all literals to R that can be resolved away by some clauses in T. This can be accomplished by the saturation technique described below.

Definition. Let E be a clause and T a theory. Then a clause F is a *saturation* of E w.r.t. T if and only if:

a) $\bar{E}$ is the complement of E by a Skolem substitution σ w.r.t. T,
b) $\bar{F}$ is the set of all unit clauses derivable by resolution from $T \cup \bar{E}$, and
c) F is the complement of $\bar{F}$ by σ^{-1} w.r.t. T.

If we only consider resolution derivations of a maximal length of k then we say that F is an *k-saturation* of E w.r.t. T.

Note that it is only under the variable assumption that all unit clauses in $\bar{F}$ are ground. Our definition of saturation is a generalization of the original definition in (Rouveirol, 1990), which is only defined for definite clauses and requires the head of saturation to be equal to the head of the original clause.

A saturation of a clause E w.r.t. a theory T is a least general alternative for the clause C in a C-derivation of E from C and T. Moreover, a saturation of a clause E w.r.t. a theory T is also a least general relative θ-subsumer of E w.r.t. T, which is stated in the Theorem 3. In Theorem 4 we show that an LGG of saturations of a set of clauses is an RLGG of the original set of clauses, which is an extension of Theorem 1 to full clauses. Proofs of Theorem 3 and Theorem 4 can be found in appendix.

Theorem 3 (Inverting relative θ-subsumption: full clauses). *Let E be a clause, T a theory, and F a saturation of E w.r.t. T. Then (under the variable assumption) $F \preceq_T E$, and for every relative θ-subsumer F' of E w.r.t. T, $F' \preceq F$.*

Theorem 4 (Relationship between RLGG and LGG: full clauses). *Let T be a theory, and $\{E_1, \ldots, E_n\}$ and $\{F_1, \ldots, F_n\}$ be two sets of clauses, such that for every $1 \leq i \leq n$, F_i is* *a saturation of* E_i *w.r.t.* T. *Then* (*under the variable assumption*) *an LGG of* $\{F_1, \ldots, F_n\}$ *is an RLGG of* $\{E_1, \ldots, E_n\}$ *w.r.t.*T.

Example. Consider the following clauses and sets of clauses:

$$\begin{aligned}
D_1 &= (\, p(x) \leftarrow q(x)\,),\\
D_2 &= (\, r(x) \leftarrow s(x)\,),\\
E_1 &= (\, p(a) \leftarrow r(b)\,),\\
E_2 &= (\, q(y) \leftarrow s(y)\,),\\
\overline{E_1} &= \{(\leftarrow p(a)\,), (\,r(b) \leftarrow)\},\\
\overline{E_2} &= \{(\leftarrow q(c)\,), (\,s(c) \leftarrow)\},\\
\overline{F_1} &= \{(\leftarrow p(a)\,), (\leftarrow q(a)), (r(b) \leftarrow)\},\\
\overline{F_2} &= \{(\leftarrow q(c)\,), (\,r(c) \leftarrow), (s(c) \leftarrow)\},\\
F_1 &= (\, p(a), q(a) \leftarrow r(b)\,),\\
F_2 &= (\, q(y) \leftarrow r(y), s(y)\,), \text{ and}\\
G &= (\, q(z) \leftarrow r(w)\,).
\end{aligned}$$

Let $T = \{D_1, D_2\}$ be a theory. The complement of E_1 and E_2 by the Skolem substitution $\sigma = \{y/a, z/b\}$ is the sets of clauses $\overline{E_1}$ and $\overline{E_2}$. Then $\overline{F_1}$ is the set of all unit clauses derivable by resolution from $T \cup \overline{E_1}$, and $\overline{F_2}$ is the set of all unit clauses derivable by resolution from $T \cup \overline{E_2}$. The complement of $\overline{F_1}$ and $\overline{F_2}$ by the inverse Skolem substitution $\sigma^{-1} = \{a/y, b/z\}$ are the clauses F_1 and F_2. Thus F_1 and F_2 are saturations of E_1 and E_2. We have that the clause G is an LGG of $\{F_1, F_2\}$, and thus an RLGG of $\{E_1, E_2\}$ w.r.t. T.

A saturation of a finite clause is in general an infinite clause, but a k-saturation for some specified positive integer k is always finite. This leads us to our technique for computation of approximate RLGGs, which is summarized in Algorithm 2.

Algorithm 2 (Computation of approximate RLGGs: full clauses).
Input: *A finite set of finite clauses* $\{E_1, \ldots, E_n\}$, *a theory* T, *and a positive integer* k.
Output: *An approximate RLGG of* $\{E_1, \ldots, E_n\}$ *w.r.t.* T.
1) *For every* $1 \leq i \leq n$, *compute the* k-*saturation* F_i *of* E_i *w.r.t.* T.
2) *Compute and return the LGG of* $\{F_1, \ldots, F_n\}$.

5. Concluding remarks

We have studied the problem of generalization of a set of full clauses relative to a theory of full clauses. We first described Plotkin's framework for relative generalization w.r.t. a theory. We then noted that techniques for relative generalization, based on the V-operators or saturation in its original form, primarily have been developed for Horn clauses, and showed that they are incomplete for full clauses. We therefore presented a generalization of the original saturation technique, and proved that it is complete for full clauses, which is our main contribution. Similar ideas have earlier been presented in (Idestam-Almquist, 1992; Muggleton, 1993), but without proofs.

A relative generalization of a set of clauses w.r.t. a theory is defined in terms of C-derivations, which means that the generalization only need to be used once in a proof of any of the clauses in the considered set of clauses. By such a definition we only consider relative generalizations under θ-subsumption, and not relative generalizations under implication. However, to find relative generalizations under implication that are not relative generalizations under θ-subsumption, we can combine our saturation technique with the technique for generalization of full clauses under implication found in (Idestam-Almquist, 1995).

Appendix

In this appendix we give the proofs of Theorems 3 and 4. Lemmas 1 and 3 are used in the proof of Theorem 3, and Lemma 2 is used in the proof of Lemma 3. In the proof of Theorem 4 we use Theorem 3 and Lemma 4.

Lemma 1. *Let T be a theory, E a clause, $\bar{E}$ the complement of E by a Skolem substitution σ w.r.t. T, and $\bar{F}$ a set of ground unit clauses such that $T \cup \bar{E} \vdash_{\mathcal{R}} \bar{F}$ and $\bar{E} \subseteq \bar{F}$. Then $F \preceq_T E$, where F is the complement of $\bar{F}$ by σ^{-1} w.r.t. T.*

Proof: The proof is by mathematical induction on the number n of clauses in $\bar{F} - \bar{E}$ (the set of clauses in $\bar{F}$ that are not in $\bar{E}$). It should be noted that $\bar{F}$, in the statement of the lemma, in the proof is indexed by n.

Base step ($n = 0$): We have $\overline{F_0} = \bar{E}$, and then $F_0 = E$, where F_0 is the complement of $\overline{F_0}$ by σ^{-1} w.r.t. T. Then $(F_0, T) \vdash_C F_0$ and $F_0 \preceq E$, and consequently $F_0 \preceq_T E$.
Induction hypothesis ($n = k$): If $\overline{F_k}$ includes k clauses that are not in $\bar{E}$, then $F_k \preceq_T E$, where F_k is the complement of $\overline{F_k}$ by σ^{-1} w.r.t. T.

Induction step ($n = k + 1$): By the induction hypothesis $T \cup \bar{E} \vdash_{\mathcal{R}} \overline{F_k}$. Let $\overline{F_{k+1}} = \overline{F_k} \cup \{\{L\}\}$, where $\{L\}$ is a ground unit clause such that $T \cup \bar{E} \vdash_{\mathcal{R}} \{L\}$ and $\{L\} \notin \overline{F_k}$. Then there must exist a clause D such that $T \vdash_{\mathcal{R}} D$ and $(D, \overline{F_k}) \vdash_C \{L\}$. Then we have $D \subseteq F_k \cup \{L\}\sigma^{-1}$ and $\{L\}\sigma^{-1} \subseteq D$. Then F_k is a resolvent of F_{k+1} and D. By the induction hypothesis there exists a clause R such that $(F_k, T) \vdash_C R$ and $R \preceq E$. Thus we have $(F_{k+1}, T) \vdash_C R$, and consequently $F_{k+1} \preceq_T E$. □

Lemma 2. *Let C and D be clauses, $R\gamma$ an instance of a resolvent R of C and D, and $\overline{R\gamma}$ the complement of $R\gamma$ by a Skolem substitution σ. Then (under the variable assumption) there exists a unit clause H such that $C \preceq R\gamma \cup H$ and $\{D\} \cup \overline{R\gamma} \vdash_{\mathcal{R}} \bar{H}$, where $\bar{H}$ is the complement of H by σ.*

Proof: Let $C' = \{A_1, \ldots, A_m\}$ be a factor of C, $D' = \{B_1, \ldots, B_n\}$ a factor of D, θ an mgu of $\{\overline{A_k}, B_p\}$, and $R = \{A_1, \ldots, A_{k-1}, A_{k+1}, \ldots, A_m, B_1, \ldots, B_{p-1}, B_{p+1}, \ldots, B_n\}\theta$.

Then $\overline{R\gamma} = \{\{\bar{A}_1\}\theta\gamma\sigma, \ldots, \{\bar{A}_{k-1}\}\theta\gamma\sigma, \{\bar{A}_{k+1}\}\theta\gamma\sigma, \ldots, \{\bar{A}_m\}\theta\gamma\sigma, \{\bar{B}_1\}\theta\gamma\sigma, \ldots, \{\bar{B}_{p-1}\}\theta\gamma\sigma, \{\bar{B}_{p+1}\}\theta\gamma\sigma, \ldots, \{\bar{B}_n\}\theta\gamma\sigma\}$.

Let $H = \{\bar{B}_p\}\theta\gamma$. Since $\bar{B}_p\theta = A_k\theta$, we have $\{A_1, \ldots, A_m\}\theta\gamma \subseteq R\gamma \cup H$. Then $C'\theta\gamma \subseteq R\gamma \cup H$, and consequently $C \preceq R\gamma \cup H$.

We have $\bar{H} = \{\{B_p\}\theta\gamma\sigma\}$. We also have $\{\{B_1, \ldots, B_n\}, \{\bar{B}_1\}\theta\gamma\sigma, \ldots, \{\bar{B}_{p-1}\}\theta\gamma\sigma, \{\bar{B}_{p+1}\}\theta\gamma\sigma, \ldots, \{\bar{B}_n\}\theta\gamma\sigma\} \vdash_{\mathcal{R}} \{B_p\}\delta$, where $\delta \subseteq \theta\gamma\sigma$ include bindings of all variables in B_p that also occur in $\{B_1, \ldots, B_{p-1}, B_{p+1}, \ldots, B_n\}$. By the variable assumption $\{B_p\}\delta = \{B_p\}\theta\gamma\sigma$, and consequently $\{D\} \cup \overline{R\gamma} \vdash_{\mathcal{R}} \bar{H}$. □

Lemma 3. *Let C be a clause, T a theory, $R\gamma$ an instance of a clause R such that $(C, T) \vdash_C R$, and let $\overline{R\gamma}$ be the complement of $R\gamma$ by σ. Then (under the variable assumption) there exist a set of unit clauses $\{H_1, \ldots, H_n\}$ such that $C \preceq R\gamma \cup H_1 \cup \cdots \cup H_n$ and $T \cup \overline{R\gamma} \vdash_{\mathcal{R}} \overline{H_1} \cup \cdots \cup \overline{H_n}$, where for every $1 \le i \le n$, $\overline{H_i}$ is the complement of H_i by σ.*

Proof: The proof is by mathematical induction on the number of resolution steps n in the branch from C to R in the C-derivation of R from C and T. It should be noted that R, in the statement of the lemma, in the proof is indexed by n.

Base step $(n = 0)$: We have $R_0 = C$, and thus $C\gamma_0 = R_0\gamma_0$. Then there exists an empty set of unit clauses $\{H_1, \ldots, H_0\}$ such that trivially $C \preceq R_0\gamma_0 \cup H_1 \cup \cdots \cup \overline{H_0}$ and $T \cup \overline{R_0\gamma} \vdash_{\mathcal{R}} \overline{H_1} \cup \cdots \cup \overline{H_0}$, where for every $1 \le i \le 0$, $\overline{H_i}$ is the complement of H_i by σ.

Induction hypothesis $(n = k)$: There exists a set of unit clauses $\{H_1, \ldots, H_k\}$ such that $C \preceq R_k\gamma \cup H_1 \cup \cdots \cup H_k$ and $T \cup \overline{R_k\gamma} \vdash_{\mathcal{R}} \overline{H_1} \cup \cdots \cup \overline{H_k}$, where for every $1 \le i \le k$, $\overline{H_i}$ is the complement of H_i by σ.

Induction step $(n = k + 1)$: One of the parent clauses of R_{k+1} is R_k. Let D be the other parent clause, and we have $T \vdash_{\mathcal{R}} D$. By Lemma 2 (under the variable assumption) there exists a unit clause H_{k+1} such that $R_k \preceq R_{k+1}\gamma_{k+1} \cup H_{k+1}$ and $\overline{R_{k+1}\gamma_{k+1}} \cup \{D\} \vdash_{\mathcal{R}} \overline{H_{k+1}}$, where $\overline{H_{k+1}}$ is the complement of H_{k+1} by σ. Hence, there exists a substitution γ_k such that $R_k\gamma_k \subseteq R_{k+1}\gamma_{k+1} \cup H_{k+1}$.

By the induction hypothesis there exists a set of unit clauses $\{H_1, \ldots, H_k\}$ such that $C \preceq R_k\gamma_k \cup H_1 \cup \cdots \cup H_k$. Since $R_k\gamma_k \subseteq R_{k+1}\gamma_{k+1} \cup H_{k+1}$, we have $R_k\gamma_k \cup H_1 \cup \cdots \cup H_k \subseteq R_{k+1}\gamma_{k+1} \cup H_1 \cup \cdots \cup H_{k+1}$. Consequently, $C \preceq R_{k+1}\gamma_{k+1} \cup H_1 \cup \cdots \cup H_{k+1}$.

By the induction hypothesis we also have $T \cup \overline{R_k\gamma_k} \vdash_{\mathcal{R}} \overline{H_1} \cup \cdots \cup \overline{H_k}$, where for every $1 \le i \le k$, $\overline{H_i}$ is the complement of H_i by σ. Since $R_k\gamma_k \subseteq R_{k+1}\gamma_{k+1} \cup H_{k+1}$, we have $\overline{R_k\gamma_k} \subseteq \overline{R_{k+1}\gamma_{k+1}} \cup \overline{H_{k+1}}$, and thus trivially $\overline{R_{k+1}\gamma_{k+1}} \cup \overline{H_{k+1}} \vdash_{\mathcal{R}} \overline{R_k\gamma_k}$. Then since $\overline{R_{k+1}\gamma_{k+1}} \cup \{D\} \vdash_{\mathcal{R}} \overline{H_{k+1}}$, we have $\overline{R_{k+1}\gamma_{k+1}} \cup \{D\} \vdash_{\mathcal{R}} \overline{R_k\gamma_k} \cup \overline{H_{k+1}}$. Then since $T \vdash_{\mathcal{R}} D$, we have $T \cup \overline{R_{k+1}\gamma_{k+1}} \vdash_{\mathcal{R}} \overline{R_k\gamma_k} \cup \overline{H_{k+1}}$. Then since $T \cup \overline{R_k\gamma_k} \vdash_{\mathcal{R}} \overline{H_1} \cup \cdots \cup \overline{H_k}$, we have $T \cup \overline{R_{k+1}\gamma_{k+1}} \vdash_{\mathcal{R}} \overline{H_1} \cup \cdots \cup \overline{H_{k+1}}$.

Consequently (under the variable assumption) there exists a set of unit clauses $\{H_1, \ldots, H_{k+1}\}$ such that $C \preceq R\gamma \cup H_1 \cup \cdots \cup H_{k+1}$ and $T \cup \overline{R\gamma} \vdash_{\mathcal{R}} \overline{H_1} \cup \cdots \cup \overline{H_{k+1}}$, where for every $1 \le i \le k + 1$, $\overline{H_i}$ is the complement of H_i by σ. □

Theorem 3 (Inverting relative θ-subsumption: full clauses). *Let E be a clause, T a theory, and F a saturation of E w.r.t. T. Then (under the variable assumption) $F \preceq_T E$, and for every relative θ-subsumer F' of E w.r.t. T, $F' \preceq F$.*

Proof: Let $\bar{E}$ be the complement of E by a Skolem substitution σ w.r.t. T, and $\bar{F}$ the complement of F by σ. By the definition of saturation, $\bar{F}$ is the set of all unit clauses derivable by resolution from $T \cup \bar{E}$, and thus $\bar{E} \subseteq \bar{F}$. By the variable assumption, all clauses in $\bar{F}$ are ground. Then by Lemma 1, we have $F \preceq_T E$.

Let F' be an arbitrary relative θ-subsumer of E w.r.t. T. Then there exists a clause R such that $(F', T) \vdash_C R$ and $R \preceq E$. Then there exists a substitution γ such that $R_\gamma \subseteq E$. Let $\overline{R\gamma}$ be the complement of $R\gamma$ by σ. Then by Lemma 3 (under the variable assumption), there exists a set of unit clauses $\{H_1, \ldots, H_n\}$ such that $F' \preceq R\gamma \cup H_1 \cup \cdots \cup H_n$ and $T \cup \overline{R\gamma} \vdash_{\mathcal{R}} \overline{H_1} \cup \cdots \cup \overline{H_n}$, where for every $1 \leq i \leq n$, $\overline{H_i}$ is the complement of H_i by σ.

Since $R\gamma \subseteq E$, we have $\overline{R\gamma} \subseteq \bar{E}$. Thus $T \cup \bar{E} \vdash_{\mathcal{R}} \overline{R\gamma} \cup \overline{H_1} \cup \cdots \cup \overline{H_n}$, and $\overline{R\gamma} \cup \overline{H_1} \cup \cdots \cup \overline{H_n} \subseteq \bar{F}$. Then we have $R\gamma \cup H_1 \cup \cdots \cup H_n \subseteq F$, and $R\gamma \cup H_1 \cup \cdots \cup H_n \preceq F$. Consequently $F' \preceq F$. □

Lemma 4. *Let T be a theory, and C, R and G clauses such that $(C, T) \vdash_C R$ and $G \preceq C$. Then there exists a clause H such that $(G, T) \vdash_C H$ and $H \preceq R$.*

Proof: The proof is by mathematical induction on the number of resolution steps n in the branch from C to R in the C-derivation of R from C and T. It should be noted that R, in the statement of the lemma, in the proof is indexed by n.

Base step $(n = 0)$: We have $R_0 = C$. Then let $H_0 = G$, and we have $(G, T) \vdash_C H_0$ and $H_0 \preceq R_0$.

Induction hypothesis $(n = k)$: There exists a clause H_k such that $(G, T) \vdash_C H_k$ and $H_k \preceq R_k$.

Induction step $(n = k + 1)$: We have that R_{k+1} is a resolvent of R_k and a clause D such that $T \vdash_{\mathcal{R}} D$. Let R'_k be a factor of R_k, D' a factor of D, A a literal in R', B a literal in D', and θ an mgu of $\{A, \bar{B}\}$. Then $R_{k+1} = ((R'_k - \{A\}) \cup (D' - \{B\}))\theta$. By the induction hypothesis $H_k \preceq R_k$, and then there exists a substitution γ such that $H_k\gamma \subseteq R_k$. Then there exists a substitution δ such that $H_k\delta \subseteq R'_k$.

If $H_k\delta \subseteq (R'_k - \{A\})$ then $H_k\delta\theta \subseteq R_{k+1}$, and thus $H_k \preceq R_{k+1}$. By the induction hypothesis $(G, T) \vdash_C H_k$. Let $H_{k+1} = H_k$, and consequently there exists a clause H_{k+1} such that $(G, T) \vdash_C H_{k+1}$ and $H_{k+1} \preceq R_{k+1}$.

If $H_k\delta \not\subseteq (R'_k - \{A\})$ then there exists a literal $L \in H_k$ such that $L\delta = A$. Then there exists an mgu of μ of $\{L, \bar{B}\}$. Thus $H_{k+1} = ((H_k - \{L\}) \cup (D' - \{B\}))\mu$ is a resolvent of H_k and D. By the induction hypothesis $(G, T) \vdash_C H_k$, and thus $(G, T) \vdash_C H_{k+1}$. Since $H_k\gamma \subseteq R'_k$, there exists a substitution ρ such that $(H_k - \{L\})\mu\rho \subseteq (R'_k - \{A\})\theta$ and $(D' - \{B\})\mu\rho = (D' - \{B\})\theta$, and thus $H_{k+1}\rho \subseteq R_{k+1}$. Consequently there exists a clause H_{k+1} such that $(G, T) \vdash_C H_{k+1}$ and $H_{k+1} \preceq R_{k+1}$. □

Theorem 4 (Relationship between RLGG and LGG: full clauses). *Let T be a theory, and $\{E_1, \ldots, E_n\}$ and $\{F_1, \ldots, F_n\}$ be two sets of clauses, such that for every $1 \leq i \leq n$, F_i is a saturation of E_i w.r.t. T. Then (under the variable assumption) an LGG of $\{F_1, \ldots, F_n\}$ is an RLGG of $\{E_1, \ldots, E_n\}$ w.r.t. T.*

Proof: By Theorem 3, for every $1 \leq i \leq n$, $F_i \preceq_T E_i$. Thus, for every $1 \leq i \leq n$, there exists a clause R_i such that $(F_i, T) \vdash_C R_i$ and $R_i \preceq E_i$. Let G be an LGG of $\{F_i, \ldots, F_n\}$. Then by Lemma 4, for every $1 \leq i \leq n$, there exists a clause H_i such that $(G, T) \vdash_C H_i$

and $H_i \preceq R_i$. Thus, for every $1 \leq i \leq n H_i \preceq E_i$ and $G \preceq_T E_i$. Hence, G is a relative generalization of $\{E_1, \ldots, E_n\}$ w.r.t. T.

Let G' be an arbitrary relative generalization of $\{E_1, \ldots, E_n\}$ w.r.t. T. Then for every $1 \leq i \leq n$, $G' \preceq_T E_i$. Then by Theorem 3, for every $1 \leq i \leq n$, $G' \preceq F_i$. Then by the definition of an LGG, we have $G' \preceq G$. Consequently G is an RLGG of $\{E_1, \ldots, E_n\}$. □

Acknowledgments

This work has been supported by the Swedish Research Council for Engineering Sciences (TFR) and the European Community ESPRIT BRA 6020 Inductive Logic Programming.

References

Chang, C., & Lee, R. (1973). *Symbolic Logic and Mechanical Theorem Proving*, London: Academic Press.

Gallier, J.H. (1986). *Logic for Computer Science—Foundations of Automatic Theorem Proving*, Harper & Row Publishers.

Genesereth, M.R., & Nilsson, N.J. (1987). *Logical Foundations of Artificial Intelligence*, Morgan Kaufmann.

Hume, D., & Sammut, C. 1991. Using inverse resolution to learn relations from experiments. In *Proceedings of the Eighth International Workshop on Machine Learning*, Morgan Kaufmann.

Idestam-Almquist, P. (1992). Learning missing clauses by inverse resolution. In *Proceedings of the International Conference on Fifth Generation Computer Systems 1992*, Tokyo: Ohmsha Publishers.

Idestam-Almquist, P. (1995). Generalization of clauses under implication. *Journal of Artificial Intelligence Research*, 3:467–489.

Lloyd, J.W. (1987). *Foundations of Logic Programming*, Springer-Verlag. Second edition.

Muggleton, S. (1991). Inductive logic programming. *New Generation Computing Journal*, 8(4):295–318.

Muggleton, S. (1993). Inductive logic programming: Derivations, successes and shortcomings. In *Proceedings of the Sixth European Conference on Machine Learning*, Springer-Verlag.

Muggleton, S. & Buntine, W. (1988). Machine invention of first-order predicates by inverting resolution. In *Proceedings of the Fifth International Conference on Machine Learning*, Morgan Kaufmann.

Plotkin, G.D. (1970). A note on inductive generalization. In B. Meltzer & D. Michie (Eds.), *Machine Intelligence*, Edinburgh University Press, 5:153–163.

Plotkin, G.D. (1971). *Automatic Methods of Inductive Inference*. Ph.D. thesis, Edinburgh University.

Plotkin, G.D. (1971). A further note on inductive generalization. In B. Meltzer & D. Michie (Eds.), *Machine Intelligence*, Edinburgh University Press, 6:101–124.

Rouveirol, C. (1990). Saturation: Postponing choices when inverting resolution. In *Proceedings of the Ninth European Conference on Artificial Intelligence*, Pitman.

Rouveirol, C. (1992). Extensions of inversion of resolution applied to theory completion. In S. Muggleton (Ed.), *Inductive Logic Programming*, San Diego, California: Academic Press.

Rouveirol, C. & Puget, J.-F. (1989). A simple solution for inverting resolution. In *Proceedings of the Fourth European Working Session on Learning*, Pitman.

Sammut, C. (1981). Concept learning by experiment. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*. Loss Altos, California: William Kaufmann.

Wirth, R. (1989). Completing logic programs by inverse resolution. In *Proceedings of the Fourth European Working Session on Learning*, Pitman.

Received November 15, 1995
Accepted June 25, 1996
Final Manuscript August 1, 1996