# Which Hypotheses Can Be Found with Inverse Entailment?

–Extended Abstract*–
Akihiro YAMAMOTO†
Fachgebiet Intellektik, Fachbereich Informatik
Technische Hochschule Darmstadt
Alexanderstr. 10, D64283 Darmstadt, GERMANY
Phone : +49-6151-16-2863, Fax : +49-6151-16-5326
Email : yamamoto@intellektik.informatik.th-darmstadt.de

## Abstract

In this paper we give a completeness theorem of an inductive inference rule *inverse entailment* proposed by Muggleton. Our main result is that a hypothesis clause $H$ can be derived from an example $E$ under a background theory $B$ with inverse entailment iff $H$ subsumes $E$ relative to $B$ in Plotkin's sense. The theory $B$ can be any clausal theory, and the example $E$ can be any clause which is neither a tautology nor implied by $B$. The derived hypothesis $H$ is a clause which is not always definite. In order to prove the result we give declarative semantics for arbitrary consistent clausal theories, and show that SB-resolution, which was originally introduced by Plotkin, is complete procedural semantics. The completeness is shown as an extension of the completeness theorem of SLD-resolution. We also show that every hypothesis $H$ derived with saturant generalization, proposed by Rouveirol, must subsume $E$ w.r.t. $B$ in Buntine's sense. Moreover we show that saturant generalization can be obtained from inverse entailment by giving some restriction to its usage.

## 1. Introduction

In this paper we identify the class of hypotheses which can be generated with inverse entailment. *Inverse entailment* is an inductive inference rule proposed in [Muggleton, 1995]. It is used to learn a clause $H$ from a positive example $E$ under a background theory $B$. Any consistent clausal theory (conjunctions of clauses) is allowed to be the theory $B$, and any clause which is neither a tautology nor logically implied by $B$ can be given as the example $E$. The hypothesis $H$ generated with the rule is not always definite. Because the conditions on $B$, $E$, and $H$ are weaker than those for most other inference rules for ILP, we expect inverse entailment should be applied to areas to which ILP techniques have not been applied yet. However, inverse entailment is not given its logical foundations enough. In this paper we construct the logical foundations of inverse entailment.

We say a hypothesis $H$ is *correct* if $B \wedge H \models E$ and $B \wedge H$ is consistent. It was proposed in [Muggleton, 1995] that every hypothesis derived with inverse entailment is correct and that every correct hypothesis $H$ can be derived. In the terminology in [Rouveirol, 1991] he said inverse entailment is *complete* as an inductive inference rule. We showed in [Yamamoto, 1996] that his proposition does not hold in general, by giving a case in which some correct hypotheses cannot be derived with inverse entailment. We also show that every correct hypothesis $H$ can be derived if $B$ is a ground reduced definite program and $E$ is a ground unit clause. Another condition for $B$ was given by [Furukawa *et al.*, 1997] under the same assumptions on $E$ and $H$. These conditions are not sufficient when we apply inverse entailment to areas the knowledge of which is not represented as definite programs. In order to find more relaxed conditions, we clarify in this paper which hypotheses can be derived with inverse entailment in general. If the class of derivable hypotheses is identified, we could make inverse entailment complete by giving some conditions on hypothesis space so that it is subsumed by the identified class.

Our main result is that a hypothesis $H$ is derived with inverse entailment iff $H$ subsumes $E$ relative to $B$ whenever $E$ is not a tautology and $B \not\models E$. The relative subsumption was defined in [Plotkin, 1971]. If $H$ subsumes $E$ relative to $B$, it holds that $B \wedge H \models E$. This means that the definition of completeness in [Rouveirol, 1991] is too restricted to represent the ability of inverse entailment. In this paper we revise the definition so that we can represent the abilty in the form of a completeness theorem.

Every hypothesis $H$ derived with inverse entailment subsumes a ground clause $K$ which is a disjunction of

*The full paper is submitted to the ILP-97 Workshop.

†The author is a visiting researcher from Hokkaido University until May 31, 1997. The author's address from June 1, 1997 is : Division of Electronics and Information Engineering, Hokkaido University, N 13 W 8 Kitha-ku, Sapporo 060 JAPAN, phone : +81-11-706-7250, fax : +81-11-706-7808, email : yamamoto@huee.hokudai.ac.jp and yamamoto@meme.hokudai.ac.jp.

literals in a set

$$\mathrm{Bot}(E,B) = \{\neg L \mid L \text{ is a ground literal and } B \wedge \overline{E} \models L\}.$$

The formula $\overline{E}$ is obtained from $\neg E$ by replacing each variable in it with a Skolem constant symbol. The set $\mathrm{Bot}(E,B)$ is called the *bottom set* of $E$ under $B$[1]. In our theory we use the set $LC(B \wedge \overline{E}) = \{L \mid B \wedge \overline{E} \models L\}$ in stead of $\mathrm{Bot}(E,B)$, because $LC(B \wedge \overline{E})$ can be regarded as *declarative semantics* of $B \wedge \overline{E}$. As procedural semantics we adopt SB-resolution, which was introduced in [Plotkin, 1971] with the name $C$-derivation. We prove a theorem between the two semantics by extending the completeness theorem of SLD-resolution, with the subsumption theorem [Kowalski, 1970] and a property on relative subsumption given in [Plotkin, 1971].

Inverse entailment is not the first inference rule based on the strategy that every hypothesis is generated by generalizing highly specified clauses. Another inference rule based on the strategy was given in [Rouveirol, 1992] which we call the *saturant generalization* rule. She assumed that the background theory $B$ is a definite program and the positive examples $E$ are definite clauses. Every hypothesis $H$ generated with the rule is a definite clause. Saturant generalization derives hypotheses by generalizing clauses called saturants. We show that saturant generalization can be obtained from inverse entailment if we represent inverse entailment with SOLDR-resolution [Yamamoto, 1997]. SOLDR-resolution is an extension of SLD-resolution and is also a restricted version of SOL-derivation [Inoue, 1992]. We also prove the completeness of saturant generalization w.r.t. the generalized subsumption relation [Buntine, 1988].

This paper is organized as follows: In the next section we make some preparation on logic, inverse entailment, and saturation. We give a new definition of completeness of inductive rules in Section 3. In Section 4 we introduce SB-derivation and give its properties, give a relation between inverse entailment and relative subsumption, and prove the completeness of inverse entailment. In Section 5 we compare saturant generalization with inverse entailment. We also show the completeness of saturant generalization. In Section 6 we give some concluding remarks.

## 2. Preliminaries

### 2.1. Terminology in Logic

We assume the readers are familiar with the concepts on first-order logic and logic programming. We fix a first-order language $\mathcal{L}$. $HB(\mathcal{L})$ and $GL(\mathcal{L})$ denote the set of all ground atoms of $\mathcal{L}$ and the set of all ground literals, respectively.

A *clausal theory* is a finite conjunction of clauses. In this paper we define a *clause* as a formula of the form

$$F = \forall x_1 \ldots x_k (A_1 \vee A_2 \vee \ldots \vee A_n \vee \neg B_1 \vee \ldots \vee \neg B_m)$$

where $n \geq 0$, $m \geq 0$, $A_i$'s and $B_j$'s are all atoms, and $x_1, \ldots, x_k$ are all variables occurring in the atoms. We represent the formula in the form of implication:

$$A_1, A_2, \ldots, A_n \leftarrow B_1, \ldots, B_m.$$

$F^+$ and $F^-$ respectively denote $A_1, A_2, \ldots, A_n \leftarrow$ and $\leftarrow B_1, B_2, \ldots, B_m$. The *complement* of $F$ is a clausal theory

$$\overline{F} = (\neg A_1 \wedge \neg A_2 \wedge \ldots \wedge \neg A_m \wedge B_1 \wedge \ldots \wedge B_m)\sigma_F$$

where $\sigma_F$ is a substitution which replaces each variable in $F$ with a Skolem constant symbol.

A *definite clause* is a clause of the form $A_0 \leftarrow A_1, \ldots, A_n$ and a *goal clause* is a clause of the form $\leftarrow A_1, \ldots, A_n$. A clausal theory consisting of definite clauses is called a *definite program*.

**Definition.** Let $A$ be a ground atom and $I$ be an Herbrand interpretation. A definite clause $A_0 \leftarrow A_1, \ldots, A_m$ *covers* $A$ in $I$ if there is a substitution $\theta$ such that $A_0\theta = A$ and $A_i\theta$ is true in $I$ for every $i = 1, \ldots, n$.

Covering is used in defining the $T_P$ operator [Lloyd, 1987].

**Definition.** For a definite program $P$ and an Herbrand interpretation $I$, we define $T_P(I)$ as

$$T_P(I) = \{A \in HB(\mathcal{L}) \mid A \text{ is covered by a clause in } P\}.$$

We will often use the subsumption relation between two clauses.

**Definition.** A clause $D$ *subsumes* a clause $C$ if there is a substitution $\theta$ such that every literal in $D\theta$ occurs in $C$.

Two well-known extensions of the subsumption relation have been proposed: the relative subsumption relation in [Plotkin, 1971], and the generalized subsumption relation defined in [Buntine, 1988].

**Definition ([Plotkin, 1971]) .** Let $H$ and $E$ be two clauses. $H$ *subsumes* $E$ *relative to* $B$, written as $H \succeq_P E$ $(B)$, if there is a clause $F$ such that $B \models \forall(E \leftrightarrow F)$ and $H$ subsumes $F$.

**Definition ([Buntine, 1988]) .** Let $H$ and $E$ be two definite clauses. $H$ *subsumes* $E$ *w.r.t.* $B$, written as $H \succeq_B E$ $(B)$, if $T_H(M) \supseteq T_E(M)$ for every Herbrand model $M$ of $B$.

### 2.2. Inverse entailment and saturation

We give a formal definition of the inverse entailment rule in [Muggleton, 1995].

**Definition.** Let $B$ be a clausal theory and $E$ be a clause. The *bottom set* of $E$ under $B$ is a set of literals $\mathrm{Bot}(E,B) = \{L \in GL(\mathcal{L}) \mid B \wedge \overline{E} \models \neg L\}$.

**Definition.** A clause $H$ is *derived by the inverse entailment rule* from $E$ under $B$ if $H$ subsumes some ground clause $K$ which is a disjunction of literals in $\mathrm{Bot}(E,B)$.

[1] In [Muggleton, 1995] a disjunction of all literals in $\mathrm{Bot}(E,B)$ is used and called *the bottom clause*, but we do not use it because it may consists of infinitely many literals.

**Example 1 ([Muggleton, 1995]).** Let us define $B_1$ and $E_1$ as follows:

$$B_1 = (\text{pet}(x) \leftarrow \text{cat}(x)) \wedge (\text{cuddly-pet}(x) \leftarrow \text{small}(x), \text{fluffy}(x), \text{pet}(x)),$$
$$E_1 = \text{cuddly-pet}(x) \leftarrow \text{fluffy}(x), \text{cat}(x).$$

The complement of $E_1$ is

$$\overline{E_1} = (\text{fluffy}(c_x) \leftarrow) \wedge (\text{cat}(c_x) \leftarrow) \wedge (\leftarrow \text{cuddly-pet}(c_x))$$

where $c_x$ is a Skolem constant symbol for the variable $x$. The bottom set of $E_1$ under $B_1$ is

$$\text{Bot}(E_1, B_1) = \{\text{small}(c_x), \neg\text{fluffy}(c_x), \neg\text{cat}(c_x), \neg\text{pet}(c_x)\}.$$

A clause $H_1 = \text{small}(x) \leftarrow \text{fluffy}(x), \text{cat}(x)$ is derived with inverse entailment because $H_1$ subsumes a clause $K_1 = \text{small}(c_x) \leftarrow \text{fluffy}(c_x), \text{cat}(c_x)$.

No procedure with which we can generate all elements of the bottom set is given in [Muggleton, 1995], but we showed in [Yamamoto, 1997] they can be generated with SOLDR-resolution when $B$ is a definite program and $E$ is a definite clause.

For the saturant generalization rule proposed in [Rouveirol, 1992] it is assumed that every background theory $B$ is a definite program and that every example $E$ is a definite clause. The rule consists of two sub-rules: saturation and generalization. Every hypothesis generated by the rule is a definite clause.

**Definition.** Let $E$ be a definite clauses, and $B$ a definite program. A definite clause $K$ is a *saturant* of $E$ under $B$ if

1. $K^+ = E^+$, and
2. $K^-$ is a disjunction of literals in a set

$$\text{Bot}^-(E, B) = \{\neg A \mid A \in HB(\mathcal{L}) \text{ and } B \wedge \overline{E^-} \models A\}.$$

Note that, for a definite clause $E$, $\overline{E^-}$ is a definite program.

**Definition.** Let $B$ be a definite program and $E$ be a definite clause. A hypothesis $H$ is *derived by the saturant generalization rule* from $E$ under $B$ if $H$ subsumes some saturant $K$ of $E$.

Our definition is different from the original one in two points: In the original definition it is assumed that $B$ and $E$ are flattened before saturation, while we do not assume it. The set $\text{Bot}^-(E, B)$ is defined with the operator $T_{B \wedge \overline{E^-}}$ in the orignal definition, while we define it with logical consequence. We need not mind the first difference because flattening is used to make the implementation of generalization easier and have no theoretical effect. The second change is justified by the well known fact that, for any $A \in HB(\mathcal{L})$, $B \wedge \overline{E^-} \models A$ iff there is $m \geq 0$ such that $A \in T_{B \wedge \overline{E^-}}{}^m(\emptyset)$ [Lloyd, 1987].

## 3. Completeness of Inductive Inference Rules

Generally speaking an inductive inference rule is a rule with which hypotheses are derived from a given example $E$ under a background theory $B$. Each hypothesis $H$ must be correct, that is, it explains $E$ with the support of $B$. We define the correctness in logical terminology.

**Definition.** Let $B$ be a background theory and $E$ an example. A hypothesis $H$ is a *correct* for $E$ under $B$ if $B \wedge H$ is consistent and $B \wedge H \models E$.

**Definition.** Let $B$ be a background theory, $E$ an example. An inductive inference rule $\mathcal{R}$ is *correct* if every hypothesis derived from $E$ under $B$ with $\mathcal{R}$ is correct for $E$ under $B$.

In order to define the completeness of an inductive inference rule, we introduce generalization relations.

**Definition.** Let $\mathcal{B}$ and $\mathcal{H}$ be sets of formulas. A triary relation $\succeq \in \mathcal{B} \times \mathcal{H} \times \mathcal{H}$ is a *generalization relation* on $\mathcal{H}$ parameterized with $\mathcal{B}$ if $\succeq (B, H, E)$ implies $B \wedge H \models E$. In the following we write $H \succeq E$ $(B)$ instead of $\succeq (B, H, E)$.

Directly from the definition, the *implication relation* $\succeq_I$ defined as $H \succeq_I E$ $(B) \iff B \wedge H \models E$ is a generalization relation.

**Definition.** An inductive inference rule $\mathcal{R}$ is *complete* w.r.t. a generalization relation $\succeq$ if $\mathcal{R}$ is correct and every hypothesis $H$ such that $H \succeq E$ $(B)$ can be derived from $E$ under $B$ with $\mathcal{R}$.

The completeness defined in [Rouveirol, 1991] can be regarded as the completeness w.r.t. the implication relation $\succeq_I$ in our definition. We can show, with an example used in [Yamamoto, 1996], that neither inverse entailment nor saturant generalization is complete w.r.t. the implication relation.

**Example 2.** Let us consider the following $B_2$ and $E_2$:

$$B_2 = (even(0) \leftarrow) \wedge (even(s(x)) \leftarrow odd(x)),$$
$$E_2 = odd(s(s(s(0)))) \leftarrow .$$

We can show that $Bot(B_2, E_2) = \{odd(s(s(s(0)))), \neg even(0)\}$. A correct hypothesis $H_2 = odd(s(y)) \leftarrow even(y)$ cannot be derived with inverse entailment because $H_2$ subsumes none of the clauses consisting of elements in $Bot(B_2, E_2)$. We can also show that $H_2$ subsumes none of the saturants of $E_2$.

## 4. Completeness of Inverse Entailment

In this section we show that the inverse entailment rule is complete w.r.t. Plotkin's relative subsumption.

At first we introduce SB-resolution and give a useful theorem on it proved by Plotkin.

**Definition.** Let $T$ be a clausal theory and $C$ a clause. An *SB-derivation* of $(T, C)$ is a sequence of clauses $C_0 = C, C_1, \ldots, C_n$ such that

1. each $C_i$ $(i \geq 1)$ is a binary resolvent of $D_{i1}$ and $D_{i2}$ where each $D_{ij}$ is $C_k$ for some $k < i$ or is a clause in $T$, and
2. $C_0$ is used exactly once as $D_{ij}$.

If $C_n$ is an empty clause, the sequence is called an *SB-refutation* of $(T, C)$.

**Theorem 1 ([Plotkin, 1971]).** *Let $H$ and $E$ be clauses and $B$ a clausal theory. Then $H \succeq_{\mathrm{P}} E$ $(B)$ iff one of the following three holds:*

1. *$E$ is a tautology.*
2. *$B \models E$.*
3. *There is an SB-refutation of $(B \wedge \overline{E}, H)$.*

**Corollary 1.** *The relation $\succeq_{\mathrm{P}}$ is a generalization relation.*

We can show that SLD-resolution, which is famous in the logic programming theory, is a special type of the SB-resolution. Let $P$ be a definite clause and $G$ be a goal clause. Without loss of generality, we can assume that $G$ is used exactly once in an SLD-refutation of $P \wedge G$. It is easy to construct an SB-derivation of $(P, G)$ from such an SLD-refutation.

Now we give *declarative semantics* of an arbitrary clausal theory w.r.t. which SB-resolution is complete, by referring the completeness theorem of SLD-resolution.

**Theorem 2 ([Lloyd, 1987]).** *For a definite program $P$ and a goal clause $G = \leftarrow A_1, \ldots, A_n$, the following three are equivalent:*

1. *There is an SLD-refutation of $P \wedge G$.*
2. *There is a substitution $\theta$ such that $\{A_1\theta, A_2\theta, \ldots, A_n\theta\} \subseteq M(P)$ where $M(P)$ is the least Herbrand model of $P$.*
3. *$P \wedge G$ is unsatisfiable.*

Since the least Herbrand model $M(P)$ coincides with the set $LC(P) = \{A \in HB(\mathcal{L}) \mid P \models A\}$, the condition 2 is equivalent to the following condition 2':

2' There is a substitution $\theta$ such that $\{A_1\theta, A_2\theta, \ldots, A_n\theta\} \subseteq LC(P)$.

We extend the definition of $LC(T)$ so that it may be defined for any clausal theory $T$.

**Definition.** For a clausal theory $T$ we define $LC(T) = \{L \in GL(\mathcal{L}) \mid T \models L\}$.

For a definite program $P$, the two definitions of $LC(P)$ are identical because no negative ground literal is a logical consequence of $P$. If a clausal theory $T$ is unsatisfiable, $LC(T) = GL(\mathcal{L})$. Otherwise, no pair of literals in $LC(T)$ is complementary. Since no ground literal is a tautology, $LC(T) = \emptyset$ iff $T$ is a tautology. When $T$ is a consistent and $C$ is an arbitrary clause, we get a theorem which is an extension of the equivalence of 1 and 2' in Theorem 2.

**Theorem 3.** *Let $T$ be a consistent clausal theory and $C = L_1 \vee L_2 \vee \ldots \vee L_n$ be a clause. There is an SB-refutation of $(T, C)$ iff there is substitution $\theta$ such that $\{\neg L_1\theta, \neg L_2\theta, \ldots, \neg L_n\theta\} \subseteq LC(T)$.*

Now we show that the relation between inverse entailment and Plotkin's relative subsumption.

**Lemma 1.** *Let $H$ and $E$ be clauses and $B$ be a clausal theory. Assume that $E$ is not a tautology and that $B \not\models E$. Then $H$ is derived from $E$ under $B$ with inverse entailment iff $H \succeq_{\mathrm{P}} E$ $(B)$.*

By Corollary 1 and Lemma 1, we get the completeness of inverse entailment.

**Theorem 4.** *Inverse entailment is complete w.r.t. Plotkin's relative subsumption if $E$ is not a tautology and that $B \not\models E$.*

From the soundness of resolution principle, it holds that $T \wedge C$ is unsatisfiable if there is an SB-refutation of $(T, C)$. However, the converse does not hold in general, that is, we cannot extend the implication $3 \Rightarrow 1$ of Theorem 2. This is the reason why the inverse entailment rule is not complete w.r.t. $\succeq_{\mathrm{I}}$.

## 5. Comparing Saturant Generalization with Inverse Entailment

In order to compare saturant generalization with inverse entailment, we analyze the structure of $LC(P \wedge G)$ for a definite clause $P$ and a goal $G$.

At first we put $LC^+(T) = LC(T) \cap HB(\mathcal{L})$ and $LC^-(T) = LC(T) - LC^+(T)$. The following lemma can be proved easily.

**Lemma 2.** *Let $P$ be a definite program and $G$ a goal. If $P \wedge G$ is consistent, $LC^+(P \wedge G) = LC(P) = M(P)$.*

Since $\overline{E^-}$ is a definite program and $\overline{E^+}$ is a goal clause, we get the following.

**Theorem 5.** *Let $B$ be a definite program and $E$ be a definite clause. Assume that $E$ is not a tautology and that $B \not\models E$. Then it holds that $\mathrm{Bot}^-(E, B) = \{\neg A \mid A \in LC^+(B \wedge \overline{E})\}$ where $\mathrm{Bot}^-(E, B)$ is the set used in the definition of saturants.*

Since $\overline{E^+}$ belongs to the set $LC^-(B \wedge \overline{E})$, it is clear that every hypothesis generated with saturant generalization can be generated with inverse entailment.

By using SOLDR-resolution introduced in [Yamamoto, 1997], we give a more precise relation between the two rules. We have shown in [Yamamoto, 1997] that, if $P \wedge G$ is consistent, every element in $LC^-(P \wedge G)$ is a ground instance of some consequence of SOLDR-derivation of $(P, G)$. The converse also holds. In other words, every positive literals in $\mathrm{Bot}(E, B)$ can be generated by using SOLDR-resolution.

Let us consider an SOLDR-derivation of $(B \wedge \overline{E^-}, \overline{E^+})$. The literal $\overline{E^+}$ can be obtained if the skip operation at the first step of the SOLDR-derivation. If we represent a procedure of inverse entailment with SOLDR-resolution, a procedure of saturant generalization can be obtained by restricting the usage of the skip operation in SOLDR-resolution.

The difference of the two rules can also be represented by the completeness theorem of saturant generalization.

The following theorem proved by [Buntine, 1988] is helpful.

**Theorem 6 ([Buntine, 1988]).** *Let $H$ and $E$ be definite clauses, and $B$ be a definite program. Then $H \succeq_{\mathrm{B}} E$ ($B$) iff, for some substitution $\theta$, $H^+\theta = E\sigma_E$ and $B \wedge \overline{E^-} \models \neg(H^-\theta)$.*

**Corollary 2.** *The relation $\succeq_{\mathrm{B}}$ is a generalization relation.*

From the discussion above and the logic programming theory, $B \wedge \overline{E^-} \models \neg(H^-\theta)$ iff $H^-$ subsumes some disjunction of literals in $\mathrm{Bot}^-(E, B)$. This is followed by the completeness of saturant generalization.

**Theorem 7.** *Saturant generalization is complete w.r.t. Buntine's generalized subsumption.*

## 6. Concluding Remarks

In this paper we showed that inverse entailment is complete w.r.t. Plotkin's relative subsumption and saturant generalization is complete w.r.t. Buntine's generalized subsumption. We also showed that saturant generalization can be obtained from inverse entailment. In order to get the results, we used the set $LC(T)$ as declarative semantics of a consistent clausal theory $T$, and showed that SB-resolution, which is an extension of SLD-resolution, can be a complete procedural semantics for $T$.

Some learnability of definite programs was shown in [Cohen, 1995a; 1995b] by using the same strategy that inverse entailment is founded on. He gave a class of definite clauses which is learnable in polynomial time. For every hypothesis $H$ in the class, the predicate symbol of $H^+$ does not occur in $H^-$. He also showed that it is difficult to learn any class of definite clauses if the predicate symbol of $H^+$ is allowed to occur in $H^-$. Because the first condition implies that $H$ subsumes an example $E$ relative to $B$, we are now interested in extending his result so that we can treat non-definite clauses as hypotheses.

From the new proof of the subsumption theorem in [Nienhuys-Cheng and de Wolf, 1994], the set $LC(T)$ can be represent with an same operator as $T_P$. Let us define $GR^n$ as follows:

$$
\begin{aligned}
GR^0(T) &= \left\{ C \;\middle|\; \begin{array}{l} C \text{ is a ground instance} \\ \text{of a clause in } T \end{array} \right\}, \\
GR^n(T) &= GR^{n-1}(T) \cup \\
&\quad \left\{ C \;\middle|\; \begin{array}{l} C \text{ is a resolvent of} \\ C_1 \text{ and } C_2 \in GR^{n-1}(T) \end{array} \right\}, \\
GR^\omega(T) &= \cup_{n \geq 0} GR^n(T).
\end{aligned}
$$

Then it holds that $LC(T) = GR^\omega(T) \cap GL(\mathcal{L})$. For a definite program $P$, $GR^\omega(P) \cap GL(\mathcal{L}) = {T_P}^\omega(\emptyset)$ since ${T_P}^\omega(\emptyset) = LC(P)$. From this observation we conjecture that covering for non-definite clauses should be defined with some three-valued logic.

## References


[Buntine, 1988] W. Buntine. Generalized Subsumption and Its Applications to Induction and Redundancy. *Artificial Intelligence*, 36:149–176, 1988.

[Cohen, 1995a] W. W. Cohen. Pac-learning Recursive Logic Programs: Efficient Algorithms. *J. of Artificial Intelligence Research*, 2:501–539, 1995.

[Cohen, 1995b] W. W. Cohen. Pac-learning Recursive Logic Programs: Negative Results. *J. of Artificial Intelligence Research*, 2:541–573, 1995.

[Furukawa *et al.*, 1997] K. Furukawa, T. Murakami, K. Ueno, T. Ozaki, and K. Shimazu. On a Sufficient Condition for the Exisitence of Moset Specific Hypothesis in Progol. SIG-FAI-9603, 56–61, Resarch Reprot of JSAI, 1997.

[Inoue, 1992] K. Inoue. Linear Resolution for Consequence Finding. *Artificial Intelligence*, 56:301–353, 1992.

[Kowalski, 1970] R. A. Kowalski. The Case for Using Equality Axioms in Automatic Demonstration. In *Proceedings of the Symposium on Automatic Demonstaration (Lecture Notes in Mathematics 125)*, pages 112–127. Springer-Verlag, 1970.

[Lloyd, 1987] J. W. Lloyd. *Foundations of Logic Programming : Second, Extended Edition.* Springer - Verlag, 1987.

[Muggleton, 1995] S. Muggleton. Inverse Entailment and Progol. *New Generation Computing*, 13:245–286, 1995.

[Nienhuys-Cheng and de Wolf, 1994]
S.-H. Nienhuys-Cheng and Ronald de Wolf. The Subsumption Theorem in Inductive Logic Programming: Facts and Fallacies. In L. de Raedt, editor, *Proceedings of the 5th International Workshop on Inductive Logic Programming*, pages 147–160, 1994.

[Plotkin, 1971] G. D. Plotkin. *Automatic Methods of Inductive Inference.* PhD thesis, Edinburgh University, 1971.

[Rouveirol, 1991] C. Rouveirol. Completeness for Inductive Procedures. In *Proceedings of the 8th International Workshop on Machine Learning*, pages 452–456. Morgan Kaufmann, 1991.

[Rouveirol, 1992] C. Rouveirol. Extentions of Inversion of Resolution Applied to Theory Completion. In S. Muggleton, editor, *Inductive Logic Programming*, pages 63–92. Academic Press, 1992.

[Yamamoto, 1996] A. Yamamoto. Improving Theories for Inductive Logic Programming Systems with Ground Reduced Programs. Submitted to New Generation Computing, 1996.

[Yamamoto, 1997] A. Yamamoto. Representing Inductive Inference with SOLD-Derivation. Submitted to the IJCAI'97 Workshop on Abduction and Induction in AI, 1997.