Hybrid Abductive Inductive Learning: a Generalisation of Progol

Oliver Ray, Krysia Broda and Alessandra Russo

Department of Computing, Imperial College London,
180 Queen's Gate, London SW7 2BZ
{or,kb,ar3}@doc.ic.ac.uk

Abstract. The learning system Progol5 and the underlying inference method of Bottom Generalisation are firmly established within Inductive Logic Programming (ILP). But despite their success, it is known that Bottom Generalisation, and therefore Progol5, are restricted to finding hypotheses that lie within the semantics of Plotkin's relative subsumption. This paper exposes a previously unknown incompleteness of Progol5 with respect to Bottom Generalisation, and proposes a new approach, called *Hybrid Abductive Inductive Learning*, that integrates the ILP principles of Progol5 with Abductive Logic Programming (ALP). A proof procedure is proposed, called HAIL, that not only overcomes this newly discovered incompleteness, but further generalises Progol5 by computing multiple clauses in response to a single seed example and deriving hypotheses outside Plotkin's relative subsumption. A semantics is presented, called *Kernel Generalisation*, which extends that of Bottom Generalisation and includes the hypotheses constructed by HAIL.

1 Introduction

Machine Learning is the branch of Artificial Intelligence that seeks to better understand and deploy learning systems through the analysis and synthesis of analogous processes in machines. The specific task of generalising from positive and negative examples relative to given background knowledge has been much studied in Machine Learning, and when combined with a first-order clausal representation is known as *Inductive Logic Programming* (ILP) [Mug91,MR94]. The *Progol* system of Muggleton [Mug95] is a state-of-the-art and widely applied ILP system that has been successful in significant real-world applications. *Progol5* [MB00] is the latest system in the Progol family, which is based on the inference method of *Bottom Generalisation* [Mug95,Yam99]. Given background knowledge B and seed example e, Bottom Generalisation constructs and generalises a clause, called the *BottomSet* [Mug95] of B and e, to return a hypothesis h that together with B entails e. Yamamoto [Yam99] has shown that Bottom Generalisation, and hence Progol5, are limited to deriving clauses h that *subsume e relative to B* in the sense of Plotkin [Plo71]. But while approaches have been proposed that do not suffer from this limitation, for example in [YF00,Ino01,FO00], these have yet to achieve the same degree of practical success as Progol.

This paper identifies a previously unknown incompleteness of Progol5 with respect to Bottom Generalisation, and attributes this incompleteness to the routine, called STARTSET responsible for computing positive literals in the *BottomSet*. A proof procedure is proposed, called HAIL, that not only overcomes this newly discovered incompleteness, but further generalises Progol5 by computing multiple clauses in response to a single seed example and deriving hypotheses outside Plotkin's relative subsumption. A semantics is presented, called *Kernel Generalisation*, which extends that of Bottom Generalisation and includes the hypotheses constructed by HAIL. The motivation is to develop an enhanced practical system by integrating the proven ILP principles of Progol5 with *Abductive Logic Programming* (ALP) [KKT92].

The relationship between abduction and Bottom Generalisation was first established in [MB00], where the authors view the Progol5 STARTSET routine as a form of abduction, and in [Yam00], where the author shows that positive literals in the *BottomSet* can be computed by an abductive proof procedure called SOLDR. But while existing approaches for integrating abduction and Bottom Generalisation have used abduction to compute single atom hypotheses, HAIL exploits the ability of ALP to compute multiple atom hypotheses. This enables HAIL to hypothesise multiple clauses not derivable by Bottom Generalisation.

The paper is structured as follows. Section 2 defines the relevant notation and terminology, and reviews Bottom Generalisation and Progol5. Section 3 discusses the STARTSET routine and considers its soundness with respect to Bottom Generalisation. Section 4 reveals an incompleteness of STARTSET with respect to Bottom Generalisation. Section 5 introduces the semantics of *Kernel Generalisation*, and a refinement called *Kernel Set Subsumption*. The HAIL proof procedure is described and illustrated with two worked examples. Section 6 compares this approach with related work and the paper concludes with a summary and a discussion of future work.

2 Background

This section defines the notation and terminology used in this paper and provides an introduction to Progol5 and Bottom Generalisation.

It is assumed that all clauses and formulae are expressed in a first-order language $\mathcal{L}$ based on a fixed signature Σ. In addition to the usual function and predicate symbols, this signature is assumed to contain a set of Skolem constants and a set of predicate symbols called *starred predicates*, such that every non-starred predicate p of arity $n \geq 0$ is associated with a unique starred predicate p^* of arity n. Informally, p^* represents the negation of p. Skolem symbols and starred predicates are reserved for the process of Skolemisation and for the formation of contrapositives, respectively. The notations $\mathcal{GA}$ and $\mathcal{GL}$ represent respectively the sets of *ground atoms* and *ground literals* in $\mathcal{L}$. The binary relations $\vdash$ and $\models$ and $\equiv$ denote respectively derivability under SLD resolution, classical logical entailment, and classical logical equivalence.

A clause is a *set* of literals $\{A_1, \ldots, A_m, \neg B_1, \ldots, \neg B_n\}$ and will often be written in the implicative form $A_1, \ldots, A_m \text{ :- } B_1, \ldots, B_n$. When a clause appears in a logical formula it denotes the universal closure of the disjunction of its literals. Where no confusion arises, a clause and a disjunction of literals will be treated interchangeably. The complement of a clause C, written $\overline{C}$, denotes the set of unit clauses obtained from the Skolemised negation of C. A (clausal) *theory* is a set of implicitly conjoined clauses. The symbols B, H, E^+, E^- will denote Horn theories representing background knowledge, hypothesis, positive and negative examples. The symbols h and e will be hypotheses and examples consisting of a single Horn clause. The theories and clauses denoted by these symbols are assumed to contain no Skolem symbols or starred predicates.

Given a theory B and a clause e, then $\mathcal{B}$ and ϵ denote the result of *normalising* B and e. The normalised theory $\mathcal{B}$ is obtained by adding to B the positive unit clauses in $\overline{e}$. The normalised clause ϵ is the Skolemised head of e, if it exists, or the empty clause $\Box$, otherwise. Note that $B \wedge H \models e$ iff $\mathcal{B} \wedge H \models \epsilon$ for any hypothesis H. A clause C is said to *θ-subsume* a clause D, written $C \succeq D$, if and only if $C\theta \subseteq D$ for some substitution θ. A clause is *reduced* if and only if it does not θ-subsume some proper subset of itself. The relation $\succeq$ induces a *lattice* ordering on the set of reduced clauses (up to renaming of variables), and the least element in this ordering is the empty-clause. A clausal theory S is said to *clausally subsume* a clausal theory T, written $S \sqsupseteq T$, if and only if every clause in T is θ-subsumed by at least one clause in S.

2.1 Bottom Generalisation

Bottom Generalisation [Mug95,Yam99] is an approach to ILP motivated by the principle of *Inverse Entailment*, which states $B \wedge H \models e$ iff $B \wedge \neg e \models \neg H$. Thus, the negations of inductive hypotheses may be deduced from the background knowledge together with the negation of a seed example. Given B and e, the task of finding such an H will be called the task of *inductive generalisation*, and H will be said to *cover* e. In Progol5 the principle of Inverse Entailment is realised through the technique of Bottom Generalisation, which is based on the *BottomSet* [Mug95], formalised in Definition 1 below.

Definition 1 (BottomSet [Mug95]). *Let B be a Horn theory and e be a Horn clause. Then the* BottomSet *of B and e, written $Bot(B,e)$, is the clause $Bot(B,e) = \{L \in \mathcal{GL} \mid B \wedge \overline{e} \models \neg L\}$.*

Given theory B and clause e, the *BottomSet* $Bot(B,e)$ is the clause containing all ground literals whose negations may be deduced from B and the complement of e. The sets $Bot^+(B,e)$ and $Bot^-(B,e)$ will denote respectively the positive (head) atoms and the negated (body) atoms of $Bot(B,e)$. A clause h is said to be *derivable* from B and e by Bottom Generalisation if and only if it θ-subsumes $Bot(B,e)$, as formalised in Definition 2 below.

Definition 2 (Bottom Generalisation [Mug95,Yam99]). *Let B be a Horn theory and e a Horn clause. A Horn clause h (containing no Skolem constant) is said to be* derivable by Bottom Generalisation *from B and e iff $h \succeq Bot(B,e)$.*

It is shown in [Yam99] that the class of hypotheses derivable by Bottom Generalisation can be characterised by Plotkin's relative subsumption, or the related notion of C-derivation, formalised in Definitions 3 and 4 below.

Definition 3 (Relative Subsumption [Plo71]). *Clause C* subsumes *a clause D* relative to *a theory T, iff $T \models \forall(C\phi \rightarrow D)$ for some substitution ϕ.*

Definition 4 (C-Derivation [Plo71]). *A* C-derivation *of a clause D from a clausal theory T* with respect to *a clause C, is a resolution derivation of the clause D from the clauses $T \cup \{C\}$, in which C is used at most once as an input clause (i.e. a leaf). A C-derivation of the empty-clause is called a* C-refutation.

Yamamoto [Yam99] shows that given a theory B and a clause e such that $B \not\models e$, a clause h is derivable by Bottom Generalisation from B and e if and only if h subsumes e relative to B, or equivalently, if and only if there is a C-refutation from $B \cup \overline{e}$ with respect to h. The C-refutation therefore characterises the hypotheses which are derivable by Bottom Generalisation.

2.2 Progol5

Progol5 [MB00] is an established ILP system based on the efficient realisation of Bottom Generalisation. Given Horn theories B, E^+ and E^-, Progol5 aims to return an augmented theory $B' = B \cup \{h_1, \ldots, h_n\}$ that entails E^+ and is consistent with E^-. Each clause h_i hypothesised by Progol5 is maximally *compressive* in the sense that it must cover the greatest number of remaining positive examples, while containing the fewest number of literals. Progol5 also takes as input a set M of *mode-declarations* [Mug95] that specifies a language bias with which hypothesised clauses must be *compatible*. Mode-declarations consist of *head-declarations* and *body-declarations* that impose syntactic constraints on the head and body atoms of hypothesised clauses. If p and t are predicates and X is a variable, then the head-declaration $modeh[p(+t)]$ states that the atom $p(X)$ may appear in the head of a hypothesis clause, and the body-declaration $modeb[p(+t)]$ states that the atom $p(X)$ may appear in the body of a hypothesis clause providing X appears in the head. The *type predicate* t is used internally by the Progol5 $BOTTOMSET$ routine.

Progol5 consists of a standard covering loop called COVERSET and three sub-routines called STARTSET, BOTTOMSET and SEARCH. These routines are described in [Mug95,MB00] and a brief overview is provided below. COVERSET constructs hypotheses incrementally by repeatedly performing three operations until all positive examples have been covered. First it selects a seed example e from among those remaining in E^+. Then it constructs a maximally compressive hypothesis h that together with the current B covers at least e. Finally, it adds h to B and removes all covered examples from E^+. The step of hypothesis formation is performed in two stages. A finite Horn subset of $Bot(B, e)$ is constructed by STARTSET and BOTTOMSET, and is generalised by SEARCH. The head atom is computed by STARTSET by reasoning with contrapositives [Sti86], then the body atoms are computed by BOTTOMSET using a Prolog interpreter, and finally the most compressive generalisation is determined by SEARCH using a general-to-specific search through the θ-subsumption lattice.

3 Soundness of StartSet

This section considers an idealisation of the contrapositive reasoning mechanism used by the Progol5 STARTSET, and discusses the soundness of this procedure.

Contrapositives are a means of propagating negative information backwards through program clauses. For example, consider the clause a :- b where a and b are propositions. Typically this clause would be used to conclude a from b. But by the classical equivalence $a \leftarrow b$ *iff* $\neg b \leftarrow \neg a$, it could equally be used to conclude $\neg b$ from $\neg a$. This latter inference can be simulated within a logic programming context by introducing the *starred* predicates a^* and b^* to represent the negations of predicates a and b, giving the new clause b^* :- a^*.

Contrapositives obtained in this way will be called *contrapositive variants*. Any Horn clause C with exactly n body atoms yields exactly n contrapositive variants, each of which is obtained by transposing the head atom C_0 (if it exists) with a body C_j atom (assuming one exists) and starring the transposed atoms. The contrapositive variants of a Horn theory T, written $Contra(T)$, are defined as the union of the contrapositive variants of the individual clauses of T.

Recall that $Bot^+(B,e)$ is the set of ground atoms whose negations are entailed by B and $\overline{e}$. To compute such negative consequences, STARTSET uses SLD-resolution on the theory obtained from B by first adding $\overline{e}$ and then adding the contrapositive variants of the resulting clauses. These theories will be called the *complementary* and *contrapositive extensions* of B, as formalised in Definitions 5 and 6 below.

Definition 5 (Complementary Extension). *Let B be a Horn theory and e be a Horn clause. Then the* complementary extension *of B with respect to e, written B_e, is the Horn theory $B \cup \overline{e}$.*

Definition 6 (Contrapositive Extension). *Let B be a Horn theory and e be a Horn clause. Then the* contrapositive extension *of B with respect to e, written B_e^*, is the Horn theory $B_e \cup Contra(B_e)$.*

Atoms in $Bot^+(B,e)$ may be computed by identifying those ground atoms that succeed under SLD-resolution as starred queries from the contrapositive extension. As formalised in Definition 7 below, the set of all ground atoms obtained in this way is called the *StartSet* of B with respect to e, and is a subset of $Bot^+(B,e)$, as stated in Proposition 1.

Definition 7 (StartSet). *Let B be a Horn theory, and e be a Horn clause. Then the* StartSet of B with respect to e, *denoted $StartSet(B,e)$ is the set of ground atoms $\{\alpha \in \mathcal{GA} \mid B_e^* \vdash \alpha^*\}$.*

Proposition 1 (Soundness of StartSet). *Let B be a Horn theory, and e be a Horn clause. Then $StartSet(B,e) \subseteq Bot^+(B,e)$. For proof see [Ray03].*

For reasons of efficiency, the STARTSET routine used by Progol5 is more complex than the idealised STARTSET described above. It is shown in [Ray03], however, that the Progol5 routine computes only a subset of the idealised *StartSet*, and so the soundness and incompleteness results presented in this section and the next apply to both procedures.

4 Incompleteness of StartSet

This section reveals that STARTSET, and therefore Progol5, are incomplete with respect to Bottom Generalisation. Proposition 2 shows that for B and e defined as follows, the atom $c \in Bot^{+}(B,e)$ but $c \notin StartSet(B,e)$. Therefore the hypothesis $h = c$ is derivable by Bottom Generalisation from B and e, but is not computed by Progol5. Let a, b and c be proposition symbols, and define:

$$B = \left\{ \begin{array}{l} a \text{:-} b, c \\ b \text{:-} c \end{array} \right\} \qquad\qquad e = a \qquad\qquad h = c$$

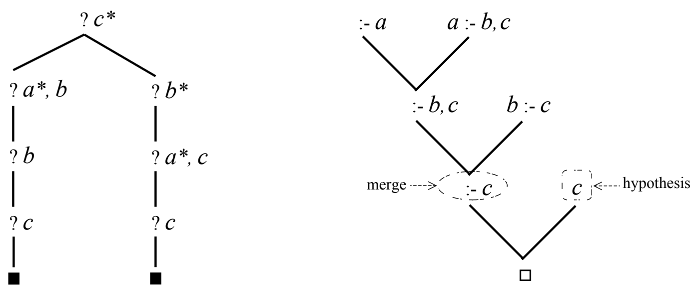
Proposition 2 (Incompleteness of StartSet). *Given B and e as defined above, then $c \in Bot^{+}(B,e)$ but $c \notin StartSet(B,e)$.*

Proof. First, referring to Definitions 5 and 6, observe that the complementary and contrapositive extensions are as follows:

$$B_e = \left\{ \begin{array}{l} a \text{:-} b, c \\ b \text{:-} c \\ \text{:-} a \end{array} \right\} \qquad\qquad B_e^{*} = \left\{ \begin{array}{l} a \text{:-} b, c \\ b \text{:-} c \\ \text{:-} a \end{array} \right\} \cup \left\{ \begin{array}{l} b^{*} \text{:-} a^{*}, c \\ c^{*} \text{:-} a^{*}, b \\ c^{*} \text{:-} b^{*} \\ a^{*} \end{array} \right\}$$

Then show that $c \in Bot^{+}(B,e)$. Observe that $B_e \wedge c \models \bot$ since c and the first two clauses of B_e entail a, and this contradicts the third clause of B_e. Therefore $B_e \models \neg c$ and so $c \in Bot^{+}(B,e)$ by Definitions 1 and 5.

Finally show that $c \notin StartSet(B,e)$. Observe that the query c^{*} fails under SLD-resolution from B_e^{*}, as shown by the SLD tree in Figure (a) below. Therefore $c \notin StartSet(B,e)$ by Definition 7.



(a) Failed SLD-computation (b) C-Derivation (with merge)

The incompleteness identified above is related to a refinement of C-refutations. A C-refutation for this example is shown in Figure (b) above. Recall that a clause as defined by Plotkin is a *set* of literals, and so identical literals are merged. Note that in this example every refutation that uses h only once, requires at least one merge. If a C-refutation with no merge of literals is called a *C*-refutation*, then it remains to show the conjecture that h is derivable by Progol5 from B and e only if there exists a C*-refutation from $B \cup \overline{e}$ with respect to h.

5 Hybrid Abductive-Inductive Learning

This section proposes a semantics that extends Bottom Generalisation, and introduces a corresponding proof procedure that generalises Progol5. The motivation underlying this approach is given in Proposition 3, which generalises a similar result in [Yam00] from definite clauses to Horn clauses, by reformulating the *BottomSet* in terms of a deductive and an abductive component.

Proposition 3. *Let $\mathcal{B}$ and ϵ be the result of normalising a Horn theory B and a Horn clause e where $B \not\models e$. Let $\alpha, \delta \in \mathcal{GA}$ denote ground atoms. As usual, let the operators $\bigwedge$ and $\bigvee$ denote respectively the conjunction and disjunction of a set of formulae (which are atoms in this case). Then*

$$Bot(B, e) \equiv \bigwedge\{\delta \in \mathcal{GA} \mid \mathcal{B} \models \delta\} \rightarrow \bigvee\{\alpha \in \mathcal{GA} \mid \mathcal{B} \wedge \alpha \models \epsilon\}$$

Proof. By Definition 1, $Bot(B, e) = \{L \mid B \wedge \overline{e} \models \neg L\}$. Partitioning into positive literals α and negative literals $\neg\delta$, this set can be rewritten as the union of the two sets: (i) $\{\alpha \mid B \wedge \overline{e} \models \neg\alpha\}$ and (ii) $\{\neg\delta \mid B \wedge \overline{e} \models \delta\}$. The proof is then by cases, according to whether e is a definite clause or a negative clause.

Case 1: Let e be the definite clause $e = \{E_0, \neg E_1, \ldots, \neg E_n\}$. Then set (i) is equal to $\{\alpha \mid B \wedge \neg E_0\sigma \wedge E_1\sigma \wedge \ldots \wedge E_n\sigma \models \neg\alpha\}$, which can be written $\{\alpha \mid \mathcal{B} \wedge \neg\epsilon \models \neg\alpha\}$, and is equal to $\{\alpha \mid \mathcal{B} \wedge \alpha \models \epsilon\}$. Set (ii) is equal to $\{\neg\delta \mid B \wedge \neg E_0\sigma \wedge E_1\sigma \wedge \ldots \wedge E_n\sigma \models \delta\}$, which can be written as $\{\neg\delta \mid \mathcal{B} \wedge \neg\epsilon \models \delta\}$, and this is now shown to be equal to $\{\neg\delta \mid \mathcal{B} \models \delta\}$ using the following argument. If δ is any atom such that $\mathcal{B} \models \delta$ then $\mathcal{B} \wedge \neg\epsilon \models \delta$ by monotonicity. Therefore $\{\neg\delta \mid \mathcal{B} \wedge \neg\epsilon \models \delta\} \supseteq \{\neg\delta \mid \mathcal{B} \models \delta\}$. If $\mathcal{B} \wedge \neg\epsilon \models \delta$ then $\mathcal{B} \wedge \neg\epsilon \wedge \neg\delta \models \bot$. Now, by the completeness of Hyper-resolution [CL73], there is a Hyper-resolution refutation from the clauses of $\mathcal{B} \cup \{\neg\epsilon\} \cup \{\neg\delta\}$, in which the electrons are $E_1\sigma, \ldots, E_n\sigma$ and any facts in $\mathcal{B}$. And since the nuclei $\neg\epsilon$ and $\neg\delta$ are negative unit clauses, they can be used only once (if at all) to derive the empty-clause in the very last step of the refutation. But suppose $\neg\epsilon$ is used, then $\neg\delta$ cannot be used, and so there is a Hyper-resolution refutation from the clauses of $\mathcal{B} \cup \{\neg\epsilon\}$, which means that $\mathcal{B} \wedge \neg\epsilon \models \bot$ by the soundness of Hyper-resolution, and so $\mathcal{B} \models \epsilon$. But this is equivalent to $B \models e$, which is a contradiction. Therefore $\neg\epsilon$ is not used, and so there is a Hyper-resolution refutation from the clauses of $\mathcal{B} \cup \{\neg\delta\}$, which means that $\mathcal{B} \models \delta$ by the soundness of Hyper-resolution. Therefore $\{\neg\delta \mid \mathcal{B} \wedge \neg\epsilon \models \delta\} \subseteq \{\neg\delta \mid \mathcal{B} \models \delta\}$. Hence $\{\neg\delta \mid \mathcal{B} \wedge \neg\epsilon \models \delta\} = \{\neg\delta \mid \mathcal{B} \models \delta\}$.

Case 2: Let e be the negative clause $e = \{\neg E_1, \ldots, \neg E_n\}$. Then set (i) is equal to $\{\alpha \mid B \wedge E_1\sigma \wedge \ldots \wedge E_n\sigma \models \neg\alpha\}$, which is equal to $\{\alpha \mid B \wedge E_1\sigma \wedge \ldots \wedge E_n\sigma \wedge \alpha \models \bot\}$ and can be written $\{\alpha \mid \mathcal{B} \wedge \alpha \models \epsilon\}$ as $\epsilon = \square$ whenever e is negative. Set (ii) is equal to $\{\neg\delta \mid B \wedge E_1\sigma \wedge \ldots \wedge E_n\sigma \models \delta\}$, which can be written $\{\neg\delta \mid \mathcal{B} \models \delta\}$.

In both cases $Bot(B, e) = \{L \mid B \cup \overline{e} \models \neg L\} = \{\neg\delta \mid B \wedge \overline{e} \models \delta\} \cup \{\alpha \mid B \wedge \overline{e} \models \neg\alpha\} = \{\neg\delta \mid \mathcal{B} \models \delta\} \cup \{\alpha \mid \mathcal{B} \wedge \alpha \models \epsilon\}$. Since the clause $Bot(B, e)$ represents the disjunction of its literals, it is therefore logically equivalent to the formula $Bot(B, e) \equiv \bigwedge\{\delta \in \mathcal{GA} \mid \mathcal{B} \models \delta\} \rightarrow \bigvee\{\alpha \in \mathcal{GA} \mid \mathcal{B} \wedge \alpha \models \epsilon\}$.

Proposition 3 shows that the atoms $\delta \in Bot^{-}(B,e)$ are those ground atoms that may be *deduced* from the normalised background $\mathcal{B}$, and that the atoms $\alpha \in Bot^{+}(B,e)$ are those ground atoms that may be *abduced* from $\mathcal{B}$ given as goal the normalised example ϵ. This has two important implications. First, the incompleteness of Progol5 identified in Section 4 can be avoided by replacing the STARTSET routine with an abductive procedure for deriving single atom hypotheses α. Second, the semantics of Bottom Generalisation can be extended, and the Progol5 proof procedure can be further generalised, by exploiting abductive hypotheses with multiple atoms, as shown in the next two subsections.

5.1 Semantics

This subsection introduces a new semantics called *Kernel Generalisation*, and a refinement of this semantics called *Kernel Set Subsumption*. The underlying notion, called a *Kernel*, is a logical formula that generalises the *BottomSet* by replacing the single atoms α in Proposition 3 by sets of (implicitly conjoined) atoms $\Delta = \{\alpha_1, \dots, \alpha_n\}$, as formalised in Definition 8 below.

Definition 8 (Kernel). *Let $\mathcal{B}$ and ϵ be the result of normalising a Horn theory B and a Horn clause e such that $B \not\models e$. Then the* Kernel *of B and e, written $\mathcal{K}er(B,e)$, is the formula defined as follows:*

$$\mathcal{K}er(B,e) = \bigwedge\{\delta \in \mathcal{GA} \mid \mathcal{B} \models \delta\} \rightarrow \bigvee\{\Delta \subseteq \mathcal{GA} \mid \mathcal{B} \wedge \Delta \models \epsilon\}$$

As formalised in Definition 9 below, any formula that logically entails the Kernel is said to be *derivable* by Kernel Generalisation, and as shown in Proposition 4 below, all such formulae are correct inductive generalisations.

Definition 9 (Kernel Generalisation). *Let B be a Horn theory and e be a Horn clause such that $B \not\models e$. Then a Horn theory H is said to be* derivable by Kernel Generalisation *from B and e iff $H \models \mathcal{K}er(B,e)$.*

Proposition 4 (Soundness of Kernel Generalisation). *Let H and B be Horn theories and let e be a Horn clause such that $B \not\models e$. Then $H \models \mathcal{K}er(B,e)$ only if $B \wedge H \models e$, for any Horn theory H.*

Proof. Assume $H \models \mathcal{K}er(B,e)$. For convenience, let P and S abbreviate the following formulae: let $P = \bigwedge\{\delta \in \mathcal{GA} \mid \mathcal{B} \models \delta\}$ be the conjunction of all ground atoms entailed by $\mathcal{B}$, and let $S = \bigvee\{\Delta \subseteq \mathcal{GA} \mid \mathcal{B} \wedge \Delta \models \epsilon\}$ be the disjunction of the conjunctions of ground atoms that together with $\mathcal{B}$ entail ϵ. Then observe that (i) $\mathcal{B} \models P$ as each conjunct δ of P is individually entailed by $\mathcal{B}$, and (ii) $\mathcal{B} \wedge S \models \epsilon$ as together with $\mathcal{B}$, each individual conjunct Δ of S entails ϵ, and (iii) $H \models P \rightarrow S$ by Definition 8 and the assumption above. Let $\mathcal{M}$ be a model of $\mathcal{B}$ and H. Then $\mathcal{M}$ is a model of P using (i), and of S using (iii), and of ϵ using (ii). Therefore $\mathcal{B} \wedge H \models \epsilon$, which is equivalent to $B \wedge H \models e$.

To remain within Horn clause logic, it is convenient to introduce a refinement of the Kernel, called a *Kernel Set*. Informally, a Kernel Set $\mathcal{K}$ of B and e, is a partial representation of the $Ker(B, e)$. Comparing Definition 10 below, with Definition 8 above, the set of head atoms $\{\alpha_1, \ldots, \alpha_n\}$ of $\mathcal{K}$ is seen to be an element of the consequent $\{\Delta \mid \mathcal{B} \wedge \Delta \models \epsilon\}$ of $\mathcal{K}er(B, e)$. The set of body atoms $\{\delta_1^1, \ldots, \delta_n^{m(n)}\}$ of $\mathcal{K}$ is seen to be a subset of the antecedent $\{\delta \mid \mathcal{B} \models \delta\}$.

Definition 10 (Kernel Set). *Let $\mathcal{B}$ and ϵ be the result of normalising a Horn theory B and a Horn clause e. Then a Horn theory $\mathcal{K}$ is said to be* a Kernel Set *of B and e iff*

$$\mathcal{K} = \left\{ \begin{array}{l} \alpha_1 \mathrel{:\!-} \delta_1^1, \ldots\ldots\ldots, \delta_1^{m(1)} \\ \qquad\qquad \vdots \\ \alpha_i \mathrel{:\!-} \delta_i^1, \ldots, \delta_i^j, \ldots, \delta_i^{m(i)} \\ \qquad\qquad \vdots \\ \alpha_n \mathrel{:\!-} \delta_n^1, \ldots\ldots\ldots, \delta_n^{m(n)} \end{array} \right\}$$

where $0 \leq m(i)$ denotes the number of body atoms in the i^{th} clause, and $\alpha_i \in \mathcal{GA}$ denotes the head atom of the i^{th} clause, and $\delta_i^j \in \mathcal{GA}$ denotes the j^{th} body atom of the i^{th} clause, and $\mathcal{B} \cup \{\alpha_1, \ldots, \alpha_n\} \models \epsilon$ and $\mathcal{B} \models \delta_i^j$ for all i, j such that $1 \leq i \leq n$ and $1 \leq j \leq m(i)$.

As formalised in Definition 11 below, any formula that clausally subsumes a Kernel Set is said to be *derivable* by Kernel Set Subsumption, and as shown in Proposition 5 below, all such formulae are correct inductive generalisations.

Definition 11 (Kernel Set Subsumption). *Let $\mathcal{K}$ be a Kernel Set of a Horn theory B and a Horn clause e such that $B \not\models e$. Then a Horn theory H is said to be* derivable by Kernel Set Subsumption *from B and e iff $H \sqsupseteq \mathcal{K}$.*

Proposition 5 (Soundness of Kernel Set Subsumption). *Let $\mathcal{K}$ be a Kernel Set of a Horn theory B and a Horn clause e such that $B \not\models e$. Then $H \sqsupseteq \mathcal{K}$ only if $B \wedge H \models e$.*

Proof. Assume $H \sqsupseteq \mathcal{K}$. For convenience, let P, Q, R and S abbreviate the following formulae: let $P = \bigwedge\{\delta \in \mathcal{GA} \mid \mathcal{B} \models \delta\}$ be the conjunction of all ground atoms entailed by $\mathcal{B}$, let $Q = \bigwedge\{\delta_1^1, \ldots, \delta_i^j, \ldots, \delta_n^{m(n)}\}$ be the conjunction of all body atoms of $\mathcal{K}$, let $R = \bigwedge\{\alpha_1, \ldots, \alpha_n\}$ be the conjunction of all head atoms of $\mathcal{K}$, and let $S = \bigvee\{\Delta \subseteq \mathcal{GA} \mid \mathcal{B} \wedge \Delta \models \epsilon\}$ be the disjunction of the conjunctions of ground atoms that together with $\mathcal{B}$ entail ϵ. Then observe that (i) $P \models Q$ as the conjuncts δ_i^j of Q are included among the conjuncts δ of P, and (ii) $R \models S$ as the conjunction R is one of the disjuncts Δ in S, and (iii) $\mathcal{K} \models Q \rightarrow R$, as any model of $\mathcal{K}$ that satisfies every body atom, must also satisfy every head atom, and (iv) $H \models \mathcal{K}$ by definition of θ-subsumption and the assumption above. Let $\mathcal{M}$ be a model of H. If $\mathcal{M}$ is a model of P, then $\mathcal{M}$ is a model of Q using (i), and of $\mathcal{K}$ using (iv), and of R using (iii), and of S using (ii). Therefore $H \models P \rightarrow S$, and so $H \models \mathcal{K}er(B, e)$ by Definition 8, and thus $B \wedge H \models e$ by Proposition 4.

Proposition 5 above, shows that Kernel Set Subsumption is a sound method of inductive generalisation. Proposition 6 below, shows that Kernel Set Subsumption is a strict extension of Bottom Generalisation for Horn clause logic.

Proposition 6 (Kernel Set Subsumption extends Bottom Generalisation). *Let B be a Horn theory and e a Horn clause such that $B \not\models e$. Then the set of hypotheses KSS derivable by Kernel Set Subsumption strictly includes the set of Horn clause hypotheses BG derivable by Bottom Generalisation.*

Proof. First show that $KSS \supseteq BG$. If the Horn clause h is derivable from B and e by Bottom Generalisation, then $h \succcurlyeq Bot(B, e)$ by Definition 2, and therefore $h\sigma \subseteq Bot(B, e)$ for some substitution σ. By Proposition 3 it follows $h\sigma = \alpha \text{:-} \delta_1, \ldots, \delta_n$ where $\mathcal{B} \wedge \alpha \models \epsilon$ and $\mathcal{B} \models \delta_j$ for all $0 \leq j \leq n$. Therefore the Horn theory $H = \{h\}$ is derivable by Kernel Set Subsumption using the Kernel Set $\mathcal{K} = \{\alpha \text{:-} \delta_1, \ldots, \delta_n\}$. Thus $KSS \supseteq BG$.

Now show that $KSS \neq BG$. Let $p/0$ and $q/1$ be predicates, let a and b be constants, and define $B = \{p \text{:-} q(a), q(b)\}$, $e = p$, and $h = q(X)$. Then the hypothesis $h = q(X)$ is *not* derivable by Bottom Generalisation, as it does not θ-subsume $Bot(B, e) = \{p\}$. But the hypothesis $h = q(X)$ *is* derivable by Kernel Set Subsumption, as it clausally subsumes the Kernel Set $\mathcal{K}$ consisting of the two clauses $q(a)$ and $q(b)$. Thus $KSS \neq BG$.

The notion of Kernel Set introduced above is related to an extension of Plotkin's C-refutation. Let a *K-derivation of* a clause D *from* a clausal theory T *with respect to* a clausal theory K be defined as a resolution derivation of D from $T \cup K$ in which any clause in K is used at most once. Then it remains to show the conjecture that a theory $\mathcal{K}$ is a Kernel Set of B and e only if there exists a K-refutation from $B \cup \overline{e}$ with respect to $\mathcal{K}$. Note that C-derivations are a special case of K-derivations in which K consists of a single clause C.

5.2 Proof Procedure

This subsection introduces a proof procedure for Kernel Set Subsumption, called HAIL, that integrates abductive, deductive and inductive reasoning within a cycle of learning that generalises Progol5. This cycle is illustrated in Figure 1.

HAIL, like Progol5, consists of a CoverSet loop (Steps 1 and 5) with abductive (Step 2), deductive (Step 3) and inductive (Step 4) phases. Given Horn theories B, E^+ and E^-, and a set of mode-declarations M, HAIL aims to return an augmented background knowledge $B' = B \cup H_1 \cup \ldots \cup H_m$ that entails E^+, is consistent with E^-, and such that each theory H_i for $1 \leq i \leq m$ is maximally compressive and compatible with M. On every iteration of the cycle, at least one clause is removed from E^+, and a non-empty theory H_i is added to B. It is assumed that initially E^+ is non-empty and consistent with B and E^+.

The CoverSet loop begins (Step 1) by selecting from E^+ a seed example e, which is normalised with B, giving theory $\mathcal{B}$ and atom ϵ. An abductive procedure is then used (Step 2) to find explanations $\Delta_i = \{\alpha_1, \ldots, \alpha_n\}$ of goal ϵ from theory $\mathcal{B}$. By definition, each explanation is a set of implicitly conjoined ground atoms such that $\mathcal{B} \wedge \Delta_i \models \epsilon$. Any abductive procedure can be used, but for the purposes of illustration Figure 1 depicts a tree-like computation representing the ASLD procedure of Kakas and Mancarella [KM90]. Abduced atoms α_j are shown as tapered squares, the goal ϵ is shown as an oval, and the theory $\mathcal{B}$ is implicit.

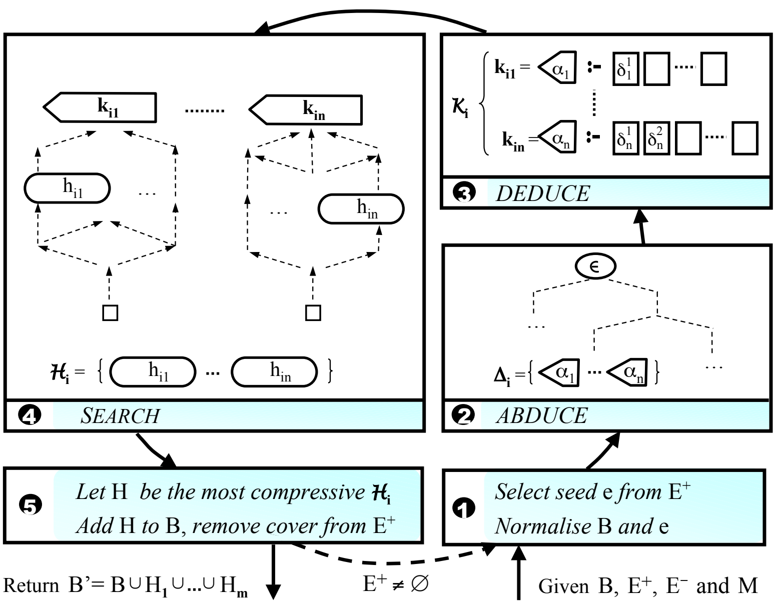


Fig. 1. Conceptual View of the HAIL Learning Cycle

Every n-atom hypothesis $\Delta_i = \{\alpha_1, \ldots, \alpha_n\}$ abduced in Step 2 is used in Step 3 to form an n-clause Kernel Set $\mathcal{K}_i = \{k_{i1}, \ldots, k_{in}\}$, with each atom α_j becoming the head of exactly one clause k_{ij}. To every head atom α_j is adjoined a set of body atoms δ_i^j, shown as squares in Figure 1, each of which is determined by a deductive procedure that computes ground atomic consequences of $\mathcal{B}$. The resulting Kernel Set $\mathcal{K}_i$ is then generalised (Step 4) by constructing a Horn theory $\mathcal{H}_i$ that includes at least one clause h_{ij} from the θ-subsumption lattice of each Kernel clause k_{ij}. Figure 1 shows the clauses h_{i1} and h_{in} (rounded rectangles) selected from the θ-subsumption lattices (dotted arrows) of the Kernel clauses k_{i1} and k_{in} (tapered rectangles). In general, the same clause may be selected from several lattices, as in the example used in Proposition 6.

The hypotheses constructed by HAIL should be compatible with the given language bias, and they should be maximally compressive in the sense of covering the greatest number of remaining positive examples while containing the fewest number of literals. Therefore, the abductive and search procedures are required return hypotheses that are minimal in the sense that no subset is also a hypothesis, and, in practice, all three procedures will make use of the mode-declarations M. In this way, the most compressive hypothesis $\mathcal{H}_i$ is determined for each Kernel Set $\mathcal{K}_i$ resulting from some explanation Δ_i. In step 5, the most compressive such hypothesis, H, is then asserted into B, and any covered examples are removed from E^+. The cycle is repeated until E^+ is empty, whereupon the augmented background B' is returned.

Begin HAIL	
Input	given B, E^+, E^-, M
remove cover	let $E^+ = E^+ - \{e \in E^+ \mid B \models e\}$
CoverSet Loop	while $E^+ \neq \emptyset$
select seed	select seed example $e \in E^+$
normalise	let $\langle \mathcal{B}, \epsilon \rangle = Normalise(B, e)$
Abduction	let $\mathcal{A} = ABDUCE(\mathcal{B}, \epsilon, M_h)$
Deduction	for each abduced hypothesis $\Delta_i \in \mathcal{A}$
	for each abduced atom $\alpha_j \in \Delta_i$
	let $k_{ij} = DEDUCE(\mathcal{B}, \alpha_j, M_b)$
	let $\mathcal{K}_i = \bigcup_j \{k_{ij}\}$
Induction	let $\mathcal{H}_i = SEARCH(\mathcal{K}_i, B, E^+, E^-, M)$
best hypothesis	let $H = \mathcal{H}_i$ with greatest $Compression$
assert hypothesis	let $B = B \cup H$
remove cover	let $E^+ = E^+ - \{e \in E^+ \mid B \models e\}$
Output	return B
End HAIL	

Fig. 2. HAIL Proof Procedure

The high-level operation of the HAIL learning cycle is shown in Figure 2, in which the abductive, deductive and search procedures are referred to generically as $ABDUCE$, $DEDUCE$ and $SEARCH$. Given as input B, E^+, E^- and M, HAIL first removes from E^+ any examples already covered by B – as these require no hypothesis. The first seed example is then selected and normalised, giving $\mathcal{B}$ and ϵ. Given theory $\mathcal{B}$ and goal ϵ, $ABDUCE$ computes a set $\mathcal{A} = \{\Delta_1, \ldots, \Delta_p\}$ of explanations, each of which is an implicitly conjoined set $\Delta_i = \{\alpha_1, \ldots, \alpha_n\}$ of ground atoms compatible with the head-declarations M_h in M, and is such that $B \wedge \Delta_i \models e$.

In the outer for-loop, each explanation $\Delta_i \in \mathcal{A}$ is processed in turn. In the inner for-loop, each atom $\alpha_j \in \Delta_i$ becomes the head of a clause k_{ij} to which $DEDUCE$ adjoins a set body atoms, each of which is a ground atomic consequence of $\mathcal{B}$ compatible with the body-declarations M_b in M. The Kernel Set $\mathcal{K}_i$ formed of the the union of the clauses k_{ij} is then generalised by SEARCH, which determines the most compressive theory $\mathcal{H}_i$ clausally subsuming the Kernel Set and compatible with M. The most compressive theory obtained in this way is then added to B, and any newly covered examples are removed from E^+.

A concrete instance of Figure 2 is proposed in [Ray03] that instantiates $ABDUCE$, $DEDUCE$ and $SEARCH$ with ASLD, BOTTOMSET and a new search algorithm called M-SEARCH. Very briefly, the language bias M_h is encoded within the ALSD procedure as additional abductive integrity constraints, the Progol5 BOTTOMSET routine is used to compute the body atoms of each individual Kernel clause, and M-SEARCH performs a recursive specific to general search through the collection subsumption lattices obtained from the given Kernel Set. These concrete procedures are now used informally in Examples 1 and 2 below, to illustrate the HAIL proof procedure in Figure 2 above.

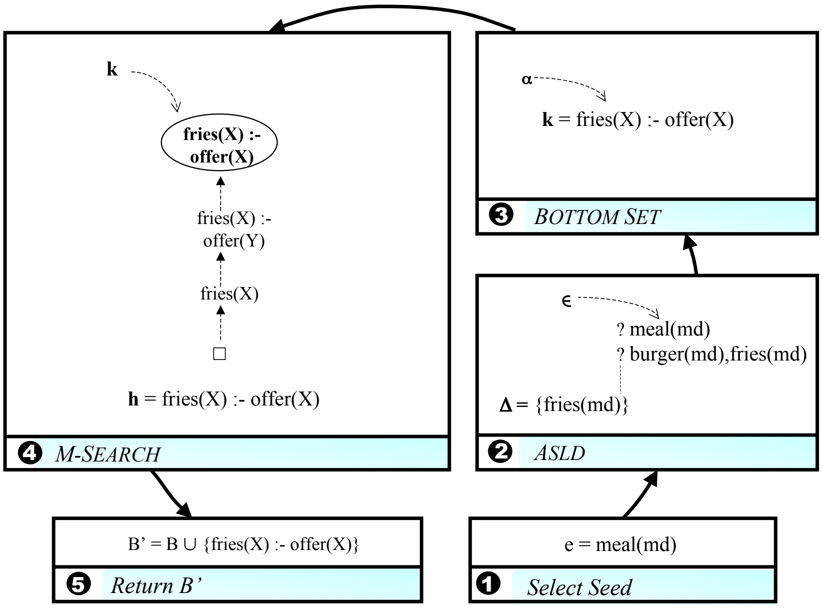


Fig. 3. Fast Food Example - Solved by HAIL (but not by Progol5)

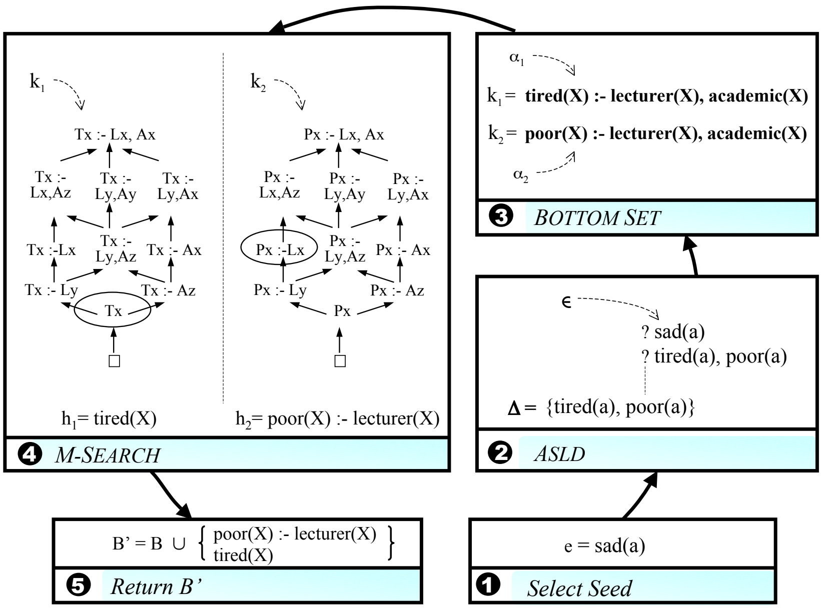


Fig. 4. Academic Example - Solved by HAIL (but not by Bottom Generalisation)

Example 1 (Fast Food).

This example shows how HAIL is able to overcome the incompleteness of Progol5 identified in Section 4. The background knowledge B describes a domain with three $bistro$s: md, bk and rz ($mcDonalds$, $burgerKing$ and $theRitz$). To have a $meal$ in a $bistro$ it is sufficient to have $burger$ and $fries$, and, a free $burger$ comes with every $fries$ at $bistro$s in a special $offer$. The positive examples E^+ state that a $meal$ has been eaten at both md and bk. The negative example(s) E^- state that a $meal$ has *not* been eaten at rz. The mode-declarations state that atoms of the form $fries(X)$ may appear in the heads of hypothesised clauses, and atoms of the form $offer(X)$ may appear in the bodies. It can be verified that hypothesis H is derivable by Bottom Generalisation using $e = meal(md)$ or $e = meal(bk)$ as the seed example. However, it is not computed by Progol5, as the queries $meal^*(md)$ and $meal^*(bk)$ fail from the contrapositive extension B_e^*. Therefore, STARTSET computes no atoms and Progol5 computes no hypothesis.

As illustrated in Figure 3, HAIL solves Example 1 in the following way. In Step 1, the seed $e = meal(md)$ is selected and normalised, trivially giving $\mathcal{B} = B$ and $\epsilon = e$. In Step 2, given theory $\mathcal{B}$ and goal ϵ, ASLD abduces the hypothesis $\Delta = \{fries(md)\}$ containing the single atom $\alpha = fries(md)$. In Step 3 α becomes the head of a clause k, to which the body atom $offer(md)$ is added by BOTTOMSET. For efficiency BOTTOMSET replaces the constant md with the variable X, as required by the mode-declarations. In Step 4 the θ-subsumption lattice, bounded from above by the newly computed clause k, and from below by the empty clause $\Box$, is searched. The most compressive hypothesis is k itself – as all more general clauses are inconsistent with the negative example $:\!-\, meal(rx)$. In Step 6 the clause $h = fries(X) :\!-\, offer(X)$ is added to B, and, because both positive examples are now covered, they are removed from E^+. The cycle terminates, returning the augmented background B'.

Background Knowledge
$B = \left\{ \begin{array}{l} meal(X) :\!-\, fries(X),\ burger(X) \\ burger(Y) :\!-\, fries(Y),\ offer(Y) \end{array} \right\} \cup \left\{ \begin{array}{l} bistro(md) \\ bistro(bk) \\ bistro(rz) \end{array} \right\} \cup \left\{ \begin{array}{l} offer(md) \\ offer(bk) \\ burger(rz) \end{array} \right\}$

Positive Examples	**Negative Examples**
$E^+ = \left\{ \begin{array}{l} meal(md) \\ meal(bk) \end{array} \right\}$	$E^- = \{ :\!-\, meal(rz) \}$

Head-Declarations	**Body-Declarations**
$M^+ = \{ \mathsf{modeh}[fries(+bistro)] \}$	$M^- = \{ \mathsf{modeb}[offer(+bistro)] \}$

Hypothesis
$H = \{ fries(Z) :\!-\, offer(Z) \}$

Example 2 (Academic).

This example shows how HAIL is able to compute more than one clause in response to a single seed example, and to derive hypotheses outside the semantics of Bottom Generalisation. The background knowledge describes a domain with three *academics*: *oli*, *ale* and *kb*. It can be verified that hypothesis H is *not* derivable by Bottom Generalisation using $e = sad(ale)$ or $e = sad(kb)$ as the seed example, since no literal with the predicates *tired* or *poor* is entailed by the complementary extension B_e. Therefore no literal with these predicates is contained in the *BottomSet* $Bot(B, e)$, nor any clause derivable by Bottom Generalisation.

As illustrated in Figure 4, HAIL solves Example 2 in the following way. In Step 1, the seed $e = sad(ale)$ is selected and normalised, again giving $\mathcal{B} = B$ and $\epsilon = e$. In Step 2, ASLD abduces the hypothesis Δ containing the two atoms $\alpha_1 = tired(ale)$ and $\alpha_2 = poor(ale)$. In Step 3, α_1 and α_2 become the heads of two clauses k_1 and k_2, to which the body atoms $lecturer(ale)$ and $academic(ale)$ are added by BOTTOMSET. Note that for efficiency BOTTOMSET then replaces the constant *ale* with the variable X, as required by the mode-declarations. Note also that, in general, different body atoms will be added to different clauses. Note finally that the two clauses k_1 and k_2 constitute a Kernel Set of B and e. In Step 4, one clause is chosen from each of the θ-subsumption lattices resulting from this Kernel Set. For ease of presentation the clauses in the θ-subsumption lattices have been written without brackets and only the first letter of each predicate symbol is shown. In Step 6 the most compressive hypothesis H consisting of the two clauses $tired(X)$ and $poor(X)$:- $lecturer(X)$ is added to B, and, because both positive examples are now covered, they are removed from E^+ and the cycle terminates returning the augmented background B'.

Background Knowledge
$B = \{ sad(X) \text{:-} tired(X), poor(X) \} \cup \left\{ \begin{array}{l} academic(oli) \\ academic(ale) \\ academic(kb) \end{array} \right\} \cup \left\{ \begin{array}{l} student(oli) \\ lecturer(ale) \\ lecturer(kb) \end{array} \right\}$

Positive Examples	**Negative Examples**
$E^+ = \left\{ \begin{array}{l} sad(ale) \\ sad(kris) \end{array} \right\}$	$E^- = \left\{ \begin{array}{l} \text{:-}\, sad(oli) \\ \text{:-}\, poor(oli) \end{array} \right\}$

Head-Declarations	**Body-Declarations**
$M^+ = \left\{ \begin{array}{l} \mathsf{modeh}[tired(+academic)] \\ \mathsf{modeh}[poor(+academic)] \end{array} \right\}$	$M^- = \left\{ \begin{array}{l} \mathsf{modeb}[lecturer(+academic)] \\ \mathsf{modeb}[academic(+academic)] \end{array} \right\}$

Hypothesis
$H = \left\{ \begin{array}{l} tired(X) \\ poor(X) \text{:-}\, lecturer(X) \end{array} \right\}$

6 Related Work

The importance of abductive inference in the context of Bottom Generalisation was first realised in [MB00] and [Yam00]. In [MB00], Muggleton and Bryant suggest that Progol5 can be seen as a procedure for efficiently generalising the atoms computed by the STARTSET routine, which they view as implementing a form of abduction based on contrapositive reasoning. This paper confirms the view of Muggleton and Bryant by showing that STARTSET performs abduction from *normalised* inputs, but reveals that STARTSET is *incomplete* with respect to Bottom Generalisation. In [Yam00] it is shown that given definite clauses B and e, then $Bot^{-}(B,e)$ is the set of atoms in the least Herbrand model of the definite theory consisting of B and the Skolemised body of e, and $Bot^{+}(B,e)$ is the set of atoms abducible by SOLDR-resolution from this program given as goal the Skolemised head of e. The Kernel semantics presented in this paper can be seen as a generalisation of these results that exploits multiple atom abductive hypotheses. In [Yam00], Yamamoto describes a procedure that incorporates explicit abduction within Bottom Generalisation. Atoms in the head and body of the *BottomSet* are computed by separate abductive and deductive procedures, and hypotheses are formed by generalising the computed atoms. However, this procedure is non-deterministic and is restricted to definite clause logic. Yamamoto shows that his procedure is able to induce a single clause or a set of facts for each seed example, but he conjectures that it would be difficult to extend the procedure to induce conjunctions of definite clauses. The proof procedure and semantics described in this paper can be seen as generalising those in [MB00] and [Yam00] by constructing Horn theories not derivable by Bottom Generalisation. But still, not all hypotheses can be found with this new approach, as can be seen using the following example due to Yamamoto [Yam97]. If $B = \{even(0)\} \cup \{even(s(X)) \,{:}\!-\, odd(X)\}$ and $e = odd(s(s(s(0))))$, then the hypothesis $h = odd(s(X)) \,{:}\!-\, even(X)$ is not derivable by Kernel Set Subsumption, or by Kernel Generalisation, as $\mathcal{K}er(B,e) = \{odd(s(s(s(0)))) \,{:}\!-\, even(0)\}$ and $h \not\models \mathcal{K}er(B,e)$. Note that in this example $\mathcal{K}er(B,e) = Bot(B,e)$.

Complete methods of hypothesis finding for full clausal logic are proposed in [YF00] and [Ino01]. In [YF00], Yamamoto and Fronhöfer describe a technique based on *Residue Hypotheses*. Very briefly, the *Residue* of a ground theory G, written $Res(G)$, is the ground theory consisting of all non-tautological clauses that contain the negation of one literal from each clause in G. A *Residue Hypothesis* of two clausal theories B and E is defined as the Residue of a subset of the ground instances of clauses in B and clauses in the Residue of the Skolemisation of E. A hypothesis H is derived by the Residue Procedure from B and E iff H generalises a Residue Hypothesis of B and E. If the example consists of a single clause e, then a theory H is derived by the Residue Procedure from B and e iff $H \models Res(Gnd(B_e))$ where $Gnd(B_e)$ denotes the ground instances of the complementary extension $B_e = B \cup \overline{e}$. Compare this with Kernel Set Subsumption, which derives a theory H iff $H \sqsupseteq \mathcal{K}$ where $\mathcal{K}$ is a Kernel Set of B and e. Both procedures derive hypotheses by generalising a ground theory constructed from B and e. For example, if $B = \{p \,{:}\!-\, q(a), q(b)\}$ and $e = p$ then $H = \{q(X)\}$ is derived by the Residue Procedure with $Res(Gnd(B_e)) = \{q(a), p\} \cup \{q(b), p\}$ and is derivable by Kernel Set Subsumption with $\mathcal{K} = \{q(a)\} \cup \{q(b)\}$, but not by Bottom Generalisation, as shown in Proposition 6. In [Ino01], Inoue describes a technique called *Consequence Finding Induction* or *CF-Induction*, which is based on the concepts of *Production Fields* and *Characteristic Clauses*. Very briefly, a Production Field defines a syntactic language bias on the hypothesis space, and a Characteristic Clause of two clausal theories B and E, is a non-tautological clause entailed $B \wedge \overline{E}$ that is expressed in the language of some Production Field P, and is not properly subsumed by any other such clause. A hypothesis H is derived by CF-Induction iff H generalises the complement of a theory $CC(B, E)$ containing a set of Characteristic Clauses. For the example above, $H = \{q(X)\}$ is derived by CF-Induction with $CC = \{p \text{:-} q(a), q(b)\} \cup \{\text{:-} p\}$ since $\overline{CC}$ is equivalent to the theory $\{q(a), p\} \cup \{q(b), p\}$, and $q(X) \models \overline{CC}$. But because the Residue Procedure and CF-Induction are more general than HAIL, they must search a correspondingly larger hypothesis space, which makes them nondeterministic and computationally expensive. It is believed, however, that practical systems can be developed for HAIL that will build on the success of Progol by overcoming some of limitations described in this paper.

7 Conclusion

This paper has identified an incompleteness of the ILP proof procedure Progol5 with respect to the semantics of Bottom Generalisation, and has proposed a new approach, called *Hybrid Abductive Inductive Learning*, that integrates abductive and inductive reasoning within a learning cycle that exploits multiple atom abductive hypotheses. A proof procedure has been presented, called *HAIL*, that overcomes this newly identified incompleteness and further generalises Progol5 by computing multiple clauses in response to a single seed example, and by finding hypotheses not derivable by Bottom Generalisation. A semantics for this proof procedure, called Kernel Generalisation, has been defined, and a refinement of this semantics, called Kernel Set Subsumption, was shown to extend that of Bottom Generalisation.

To better characterise the hypotheses derivable by HAIL, precise completeness results are required for the semantics and proof procedures presented in this paper. It is believed that K-derivations, introduced in this paper as an extension of C-derivations, will serve as the basis of such a characterisation. Although Kernel Set Subsumption is an extension of Bottom Generalisation, it is not complete with respect to the general task of inductive generalisation. Therefore, the possibility of enlarging the class of derivable hypotheses by interleaving the abductive, deductive and inductive phases will be investigated. In addition, a prototype implementation of the HAIL proof procedure needs to be developed in order to evaluate the approach. One possibility would be to generalise the current Progol5 implementation and combine this with an existing abductive system, such as the A-System of Van Nuffelen, Kakas and Denecker [KND01].

References

[CL73] C. Chang and R. C. Lee. *Symbolic Logic and Mechanical Theorem Proving*. Academic Press, 1973.

[FO00] K. Furukawa and T. Ozaki. On the completion of inverse entailment for mutual recursion and its application to self recursion. In J. Cussens and A. Frisch, editors, *Proceedings of the Work-in-Progress Track, 10th International Conference on Inductive Logic Programming*, pages 107–119, 2000.

[Ino01] K. Inoue. Induction, abduction, and consequence-finding. In C. Rouveirol and M. Sebag, editors, *Proceedings 11th International Conference on Inductive Logic Programming*, volume 2157 of *Lecture Notes in AI*, pages 65–79. Springer Verlag, 2001.

[KKT92] A.C. Kakas, R.A. Kowalski, and F. Toni. Abductive logic programming. *Journal of Logic and Computation*, 2(6):719–770, 1992.

[KM90] A.C. Kakas and P. Mancarella. Database updates through abduction. In *16th International Conference on Very Large Databases (VLDB)*, pages 650–661. Morgan Kaufmann, 1990.

[KND01] A. Kakas, B. Van Nuffelen, and M. Denecker. A-system : Problem solving through abduction. In *Proceedings of IJCAI'01 - Seventeenth International Joint Conference on Artificial Intelligence*, pages 591–596, 2001.

[MB00] S.H. Muggleton and C.H. Bryant. Theory completion using inverse entailment. *Lecture Notes in Computer Science*, 1866:130–146, 2000.

[MR94] S.H. Muggleton and L. De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19/20:629–679, 1994.

[Mug91] S.H. Muggleton. Inductive logic programming. *New Generation Computing*, 8(4):295–318, 1991.

[Mug95] S.H. Muggleton. Inverse entailment and progol. *New Generation Computing, Special issue on Inductive Logic Programming*, 13(3-4):245–286, 1995.

[Plo71] G.D. Plotkin. *Automatic Methods of Inductive Inference*. PhD thesis, Edinburgh University, 1971.

[Ray03] O. Ray. Hail: Hybrid abductive-inductive learning. Technical Report 2003/6, Imperial College, 2003.

[Sti86] M.E. Stickel. A Prolog technology theorem prover: Implementation by an extended Prolog compiler. In J. H. Siekmann, editor, *Proceedings of the Eighth International Conference on Automated Deduction*, volume 230, pages 573–587, Berlin, 1986. Springer-Verlag.

[Yam97] A. Yamamoto. Which hypotheses can be found with inverse entailment? In S. Džeroski and N. Lavrač, editors, *Proceedings of the 7th International Workshop on Inductive Logic Programming*, volume 1297, pages 296–308. Springer-Verlag, 1997.

[Yam99] A. Yamamoto. An inference method for the complete inverse of relative subsumption. *New Generation Computing*, 17(1):99–117, 1999.

[Yam00] A. Yamamoto. *Abduction and Induction: essays on their relation and integration*, volume 18 of *Applied Logic Series*, chapter Using Abduction for Induction based on Bottom Generalisation. Kluwer Academic Publishers, first edition, 2000.

[YF00] A. Yamamoto and B. Fronhöfer. Hypothesis finding via residue hypotheses with the resolution principle. In *Proceedings of the 7th International Workshop on Inductive Logic Programming*, volume 1968 of *Lecture Notes in Computer Science*, pages 156–165. Springer-Verlag, 2000.