Generalised Kernel Sets for Inverse Entailment

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 task of inverting logical entailment is of central importance to the disciplines of Abductive and Inductive Logic Programming (ALP & ILP). Bottom Generalisation (BG) is a widely applied approach for Inverse Entailment (IE), but is limited to deriving single clauses from a hypothesis space restricted by Plotkin's notion of C-derivation. Moreover, known practical applications of BG are confined to Horn clause logic. Recently, a hybrid ALP-ILP proof procedure, called HAIL, was shown to generalise existing BG techniques by deriving multiple clauses in response to a single example, and constructing hypotheses outside the semantics of BG. The HAIL proof procedure is based on a new semantics, called Kernel Set Subsumption (KSS), which was shown to be a sound generalisation of BG. But so far KSS is defined only for Horn clauses. This paper extends the semantics of KSS from Horn clause logic to general clausal logic, where it is shown to remain a sound extension of BG. A generalisation of the C-derivation, called a K*-derivation, is introduced and shown to provide a sound and complete characterisation of KSS. Finally, the K*-derivation is used to provide a systematic comparison of existing proof procedures based on IE.

1 Introduction

Abduction and induction are of great interest to those areas of *Artificial Intelligence* (AI) concerned with the tasks of explanation and generalisation, and efforts to analyse and mechanise these forms of reasoning are gaining in importance. In particular, the disciplines of *Abductive Logic Programming (ALP)* [4] and *Inductive Logic Programming (ILP)* [8] have developed semantics and proof procedures of theoretical and practical value. Fundamentally, both ALP and ILP are concerned with the task, called *Inverse Entailment (IE)*, of constructing a hypothesis that logically entails a given example relative to a given background theory. In practice, the main difference between ALP and ILP is that whereas abductive hypotheses are normally restricted to sets of ground atoms, inductive hypotheses can be general clausal theories.

To date, the inference method of *Bottom Generalisation (BG)* [6, 15] is one of the most general approaches for IE to have resulted in the development of high-performance tools of wide practical application. Central to this success has been the use of Muggleton's notion of *Bottom Set (BS)* [6] to bound a search space that would otherwise be intractable. However, methods based *directly* on BG

B. Demoen and V. Lifschitz (Eds.): ICLP 2004, LNCS 3132, pp. 165–179, 2004.

are subject to several key limitations. By definition they can only hypothesise a single clause in response to a given example, and Yamamoto [14] has shown they are further limited to deriving a class of hypotheses characterised by Plotkin's notion of *C-derivation* [10]. In practice, known proof procedures for BG are limited to Horn clause logic; as evidenced, for example, by the state-of-the-art ILP system Progol [6].

Recently, Ray *et al.* [12] proposed a hybrid ALP-ILP proof procedure, called *HAIL*, that extends the Progol approach by hypothesising multiple clauses in response to a single example, and by constructing hypotheses outside the semantics of BG. Also in [12], a semantics for HAIL called *Kernel Set Subsumption (KSS)* was presented and shown to subsume that of BG. So far, this new semantics is defined only for Horn clause logic and, as yet, no corresponding characterisation of KSS has been found to generalise the relationship between BG and C-derivations. It was conjectured in [12], however, that a natural extension of the C-derivation, called a *K-derivation*, could be used to obtain such a characterisation of KSS.

In this paper, the semantics of KSS is extended from Horn clauses to general clauses, where it is shown to remain a sound generalisation of BG. A new derivation is defined, called a *K*-derivation*, that both refines the K-derivation and generalises the C-derivation. The K*-derivation is shown to give a sound and complete characterisation of KSS, thereby resolving the conjecture above. The paper is structured as follows. Section 2 reviews the relevant background material. Section 3 lifts the semantics of KSS to general clausal logic. Section 4 introduces the K*-derivation and shows how it characterises the generalised KSS. Section 5 uses the K*-derivation as a means of comparing related approaches. The paper concludes with a summary and directions for future work.

2 Background

This section reviews the necessary background material. After a summary of notation and terminology, the notions of ALP and ILP are briefly described in order to motivate the underlying task of IE. Relevant definitions and results are recalled concerning the semantics of BG and KSS.

Notation and Terminology. A *literal* L is an atom A or the (classical) negation of an atom $\neg A$. A *clause* C is a set of literals $\{L_1, ..., L_n\}$ that for convenience will be represented as a disjunction $L_1 \vee ... \vee L_n$. Any atom that appears negated in C is called a *negative* or *body* atom, and any atom that appears unnegated in C is called a *positive* or *head* atom. A *Horn clause* is a clause with at most one head atom. The *empty clause* is denoted $\Box$. An *expression* is a term, a literal, or a clause. A *theory* T is a set of clauses $\{C_1, ..., C_m\}$ that for convenience will be represented as a conjunction $\{C_1 \wedge ... \wedge C_m\}$. This paper assumes a given first-order language $\mathfrak{L}$ that includes Skolem constants. An expression or theory is said to be *Skolem-free* whenever it contains no Skolem constant. The symbols $\top$ and $\bot$ will denote the logical constants for truth and falsity. The symbol $\models$ will denote classical first-order logical entailment. The equivalence $X \wedge Y \models Z$ iff $X \models \neg Y \vee Z$

will be called the *Entailment Theorem*. Whenever a clause is used in a logical formula, it is read as the universal closure of the disjunction of its literals. Whenever a theory is used in a logical formula, it is read as the conjunction of its clauses. In general, the symbols L, M will denote literals; λ, μ will denote ground literals; P, N will denote atoms; α, δ will denote ground atoms; S, T will denote theories; and C, D, E will denote clauses. Symbols B, H will denote Skolem-free theories representing background knowledge and hypotheses, respectively. Symbols e, h will denote Skolem-free clauses representing examples and hypotheses, respectively. A substitution σ is called a *Skolemising substitution for* a clause C whenever σ binds each variable in C to a *fresh* Skolem constant. A clause C is called a *factor* of a clause D whenever $C = D\phi$ and ϕ is a *most general unifier (mgu)* of one or more literals in D. A clause C is said to *θ-subsume* a clause D, written $C \succeq D$, whenever $C\theta \subseteq D$ for some substitution θ. A theory S is said to *θ-subsume* a theory T, written $S \sqsupseteq T$, whenever each clause in T is θ-subsumed by at least one clause in S. If L is a literal, then the *complement* of L, written $\overline{L}$, denotes the literal obtained by negating L if it is positive, and unnegating L if it is negative. If $C = L_1 \vee ... \vee L_n$ is a clause and σ is a Skolemising substitution for C, then the *complement of C (using σ)*, written $\overline{C}$, is defined as the theory $\overline{C} = \{\overline{L_1}\sigma \wedge ... \wedge \overline{L_n}\sigma\}$. The standard definition of *resolvent* is assumed, as defined for example in [2]. A *resolution derivation* of clause C from theory T is a finite non-empty sequence of clauses $\mathcal{R} = (R_1, \ldots, R_n{=}C)$ such that each clause $R_i \in (R_1, \ldots, R_n)$ is either a *fresh* variant of some clause $D \in T$, or a resolvent of two preceding clauses $P, Q \in (R_1, \ldots, R_{i-1})$. In the first case, R_i is called an *input clause*, and D is called the *generator* of R_i. In the second case, R_i is called a *resolvent*, and P and Q are called the *parents* of R_i. A *tree derivation* of C from T is a resolution derivation of C from T in which each clause except the last is the parent of exactly one child. A derivation of $\Box$ from T will also be called a *refutation from T*. The *composition* of two tree derivations $\mathcal{R}_1$ and $\mathcal{R}_2$, written $\mathcal{R}_1 + \mathcal{R}_2$, is the tree derivation obtained by concatenating the sequence $\mathcal{R}_2$ on to the sequence $\mathcal{R}_1$, taking care to rename any variables that may clash. The *Subsumption Theorem* states if a theory T logically entails a clause C then either C is a tautology or else there exists a tree derivation from T of a clause D that θ-subsumes C, as shown for example in [9].

Abductive and Inductive Logic Programming (ALP & ILP) [4, 8] formalise in a logic programming context the notions of *explanation* and *generalisation*. With respect to a given theory, ALP constructs explanations for given observations, while ILP computes generalisations of given examples. Many ALP and ILP techniques are incremental in that they focus on one observation or example at a time and try to construct a partial hypothesis, H, that entails this one example, e, relative to the background theory, B. This fundamental problem, which is known as the task of *Inverse Entailment (IE)*, is formally defined as follows. Given a theory B and a clause e, find a theory H such that $B \cup H \models e$. For reasons of efficiency, some form of search bias is normally imposed on the process used to find H, and one such method is discussed next.

Bottom Generalisation (BG) [6, 15] is an important approach for IE that is based on the construction and generalisation of a particular clause called a *Bottom Set*. Formally, as shown in Definition 1 below, the Bottom Set of B and e, denoted $Bot(B, e)$, contains the set of ground literals μ whose complements are entailed by B and the complement of e. As shown in Definition 2, the hypotheses derivable by BG are those clauses h that θ-subsume $Bot(B, e)$. It is worth emphasising that B, h and e are all assumed to be Skolem-free.

Definition 1 (Bottom Set). *Let B be a theory, let e be a clause, let σ be a Skolemising substitution for e, and let $\overline{e}$ be the complement of e using σ. Then the* Bottom Set of B and e (using σ)*, denoted $Bot(B, e)$, is the clause $Bot(B, e) = \{\mu \mid B \cup \overline{e} \models \overline{\mu}\}$ where the μ are ground literals.*

Definition 2 (BG). *Let B be a theory, and let e and h be clauses. Then h is said to be* derivable by BG from B and e *iff $h \succeq Bot(B, e)$.*

The key point is that instead of exploring the entire IE hypothesis space, which is intractable, BG only considers a sub-space that is both smaller and better structured than the original. Formally, this sub-space is the θ-subsumption lattice bounded by the Bottom Set and the empty set. But, as described below, the advantage of a more tractable search space comes at the price of incompleteness. This incompleteness can be characterised by Plotkin's notion of C-derivation [10], which is formalised in Definition 3.

Definition 3 (C-derivation). *Let T be a theory, and C and D be clauses. Then a* C-derivation of D from T with respect to C *is a tree derivation of D from $T \cup \{C\}$ such that C is the generator of at most one input clause. A clause D is said to be* C-derivable *from T with respect to C, denoted $(T, C) \vdash_c D$, iff there exists a C-derivation of D from T with respect to C.*

Informally, a C-derivation is a tree derivation in which some given clause C may be used at most once. The important result, as shown in [15], is that a hypothesis h is derivable by BG from B and e if and only if there is a C-refutation from $B \cup \overline{e}$ with respect to h. Therefore C-derivations characterise the restrictions on the hypotheses derivable by BG. In order to (partially) overcome these restrictions, the semantics of KSS was introduced, as described next.

Kernel Set Subsumption (KSS) [12] can be seen as extending BG to derive multiple clause hypotheses drawn from a larger hypothesis space. Like BG, KSS considers only a bounded lattice based sub-space of the full IE hypothesis space. But, whereas BG uses a single clause Bottom Set to bound its search space, KSS uses instead a *set* of clauses called a Kernel Set. The relevant notions are now recalled for the Horn clause case in Definitions 4, 5 and 6 below.

As shown in Definition 4, before a Kernel Set is formed, the inputs B and e are first normalised by Skolemising e and transferring the body atoms as facts to B. Formally, the normalised example ϵ is the clause containing the Skolemised head atom of e, while the normalised background knowledge $\mathcal{B}$ is the original theory B augmented with the Skolemised body atoms of e. In all of these definitions negative clauses are formally treated as if they had the head atom '$\bot$'.

Definition 4 (Horn Normalisation). *Let B be a Horn theory, let $e = P \vee \neg N_1 \vee ... \vee \neg N_m$ be a Horn clause, and let σ be a Skolemising substitution for e. Then the* normalisation of B and e (using σ), *consists of the theory $\mathcal{B} = B \cup \{N_1\sigma \wedge ... \wedge N_m\sigma\}$, and the clause $\epsilon = P\sigma$.*

Definition 5 (Horn Kernel Set). *Let $\mathcal{B}$ and ϵ be the result of normalising a Horn theory B and a Horn clause e, and let $\mathcal{K} = \{k_1 \wedge \ldots \wedge k_n\}$ be a set of ground Horn clauses $k_i = \alpha_i \vee \neg\delta_i^1 \vee \ldots \vee \neg\delta_i^{m_i}$. Then $\mathcal{K}$ is called a* Kernel Set of B and e *iff $\mathcal{B} \cup \{\alpha_1 \wedge \ldots \wedge \alpha_n\} \models \epsilon$, and $\mathcal{B} \models \delta_i^j$ for all $1 \le i \le n$ and $1 \le j \le m_i$.*

Definition 6 (Horn KSS). *Let B and H be Horn theories, and let e be a Horn clause. Then H is said to be* derivable by KSS from B and e *iff $H \sqsupseteq \mathcal{K}$ for some Kernel Set $\mathcal{K}$ of B and e.*

As shown in Definition 5 above, a Horn Kernel Set of a Horn theory B and Horn clause e is a Horn theory $\mathcal{K}$ whose head atoms $\alpha_1, \ldots, \alpha_n$ collectively entail the normalised example ϵ with respect to $\mathcal{B}$, and whose body atoms δ_i^j are individually entailed by the normalised background $\mathcal{B}$. Here, $n \ge 1$ denotes the (non-zero) number of clauses in $\mathcal{K}$, and $m_i \ge 0$ denotes the (possibly-zero) number of body atoms in the i^{th} clause k_i. As shown in Definition 6, a theory H is derivable by KSS whenever it θ-subsumes a Kernel Set $\mathcal{K}$ of B and e.

So far KSS is defined only for the Horn clause subset of clausal logic. In this context it has been shown in [12] that KSS is sound with respect to IE and complete with respect to BG. However, as yet, no exact characterisation of the class of hypotheses derivable by KSS has been established. Such a task clearly requires a more general notion than the C-derivation. For this purpose, the concept of K-derivation, formalised in Definition 7 below, was introduced in [12] and conjectured to provide such a characterisation.

Definition 7 (K-derivation). *Let T and K be theories, and let D be a clause. Then a* K-derivation of D from T with respect to K *is a tree derivation of D from $T \cup K$ such that each clause $k \in K$ (but not in T) is the generator of at most one input clause, which is called a* k-input clause. *Clause D is said to be* K-derivable *from T with respect to K, denoted $(T, K) \vdash_k D$, iff there exists a K-derivation of D from T with respect to K.*

The notion of K-derivation generalises that of C-derivation in the following way. Whereas the C-derivation refers to a clause C that may be used at most once, in a K-derivation there are a *set of clauses K each of which* may be used at most once. A C-derivation is therefore a special case of a K-derivation, in which this set $K = \{C\}$ is a singleton.

In the next two sections, the semantics of KSS is extended to general clausal logic and a refinement of the K-derivation, called a K*-derivation, is introduced in order to provide a precise characterisation of KSS in the general case. The soundness and completeness results mentioned above are also lifted to the general case and the conjecture is proved.

3 Kernel Set Semantics for General Clausal Logic

In this section the semantics of KSS is generalised from Horn clauses to arbitrary clauses. It is shown in this general case that KSS remains sound with respect to IE and continues to subsume the semantics of BG. First the notion of normalisation is generalised in Definition 8, and two key properties are shown in Lemma 1. Then the generalised notion of Kernel Set is formalised in Definition 9.

Definition 8 (Normalisation). *Let B be a theory, let $e = P_1 \vee ... \vee P_n \vee \neg N_1 \vee ... \vee \neg N_m$ be a clause, and let σ be a Skolemising substitution for e. Then the* normalisation of B and e (using σ), *consists of the theory $\mathcal{B} = B \cup \{N_1\sigma \wedge ... \wedge N_m\sigma\}$, and the clause $\epsilon = P_1\sigma \vee ... \vee P_n\sigma$.*

Lemma 1. *Let $\mathcal{B}$ and ϵ be the result of normalising a theory B and a clause $e = P_1 \vee ... \vee P_n \vee \neg N_1 \vee ... \vee \neg N_m$ using σ. Let $\overline{e}$ denote the complement of e, also using σ. Then (1) $\mathcal{B} \cup T \models \epsilon$ iff $B \cup \overline{e} \cup T \models \Box$ for all theories T, and (2) $\mathcal{B} \cup H \models \epsilon$ iff $B \cup H \models e$ for all (Skolem-free) theories H.*

Proof. Taking each case in turn:

1. $\mathcal{B} \cup T \models \epsilon$ iff $B \cup \{N_1\sigma \wedge ... \wedge N_m\sigma\} \cup T \models P_1\sigma \vee ... \vee P_n\sigma$ (by Definition 8), iff $B \cup \{N_1\sigma \wedge ... \wedge N_m\sigma \wedge \neg P_1\sigma \wedge ... \wedge \neg P_n\sigma\} \cup T \models \Box$ (by the Entailment Theorem), iff $B \cup \overline{e} \cup T \models \Box$ (by properties of complementation).
2. $\mathcal{B} \cup H \models \epsilon$ iff $B \cup \{N_1\sigma \wedge ... \wedge N_m\sigma\} \cup H \models P_1\sigma \vee ... \vee P_n\sigma$ (by Definition 8), iff $B \cup H \models P_1\sigma \vee ... \vee P_n\sigma \vee \neg N_1\sigma \vee ... \vee \neg N_m\sigma$ (by the Entailment Theorem), iff $B \cup H \models e\sigma$ (by properties of substitution), iff $B \cup H \models e$ (the forward direction uses the fact σ binds each variable in e to a constant not in B, H or e, the reverse direction uses the fact $e\sigma$ is an instance of e).

Definition 9 (Kernel Set). *Let $\mathcal{B}$ and ϵ be the result of normalising a theory B and a clause e. A* Kernel Set *of B and e is a ground theory $\mathcal{K}$ that can be written in the form:*

$$\mathcal{K} = \left\{ \begin{array}{c} \lambda_1^0 \vee \lambda_1^1 \vee \cdots\cdots\cdots \vee \lambda_1^{m_1} \\ \vdots \\ \lambda_i^0 \vee \lambda_i^1 \vee \cdots \lambda_i^j \cdots \vee \lambda_i^{m_i} \\ \vdots \\ \lambda_n^0 \vee \lambda_n^1 \vee \cdots\cdots\cdots \vee \lambda_n^{m_n} \end{array} \right\}$$

where $\mathcal{B} \cup \{\lambda_1^0 \wedge ... \wedge \lambda_n^0\} \models \epsilon$, and $\mathcal{B} \cup \{\lambda_i^j\} \models \epsilon$ for all $1 \leq i \leq n$ and $1 \leq j \leq m_i$. In this case, the literals $\lambda_1^0, \ldots, \lambda_n^0$ are called the key literals *of $\mathcal{K}$.*

In moving from the Horn case to the general case, the role previously played by the head atoms α_i is now played by the so-called *key literals* λ_i^0. Although any literal in a clause can be chosen as the key literal, for notational convenience it will be assumed that the key literal of a kernel clause k_i will always be denoted λ_i^0. As shown in Definition 9 above, for $\mathcal{K}$ to be a Kernel Set of B and e the key literals $\lambda_1^0, \ldots, \lambda_n^0$ must *collectively* entail ϵ with respect to $\mathcal{B}$, and the non-key literals λ_i^j must *individually* entail ϵ with respect to $\mathcal{B}$.

It is straightforward to show any Horn theory $\mathcal{K}$ that is a Kernel Set by Definition 5 is also a Kernel Set by Definition 9. From Definition 5 it holds $\mathcal{B} \cup \{\alpha_1 \wedge \ldots \wedge \alpha_n\} \models \epsilon$ and $\mathcal{B} \models \delta_i^j$. From the latter it follows $\mathcal{B} \cup \{\neg\delta_i^j\} \models \bot \models \epsilon$. Upon identifying each key literal λ_i^0 with the head atom α_i and each non-key literal λ_i^j with the negated body atom $\neg\delta_i^j$ it follows that $\mathcal{B} \cup \{\lambda_1^0 \wedge ... \wedge \lambda_n^0\} \models \epsilon$ and $\mathcal{B} \cup \{\lambda_i^j\} \models \epsilon$. Hence $\mathcal{K}$ is also a Kernel Set by Definition 9.

As formalised in Definition 10 below, the notion of KSS is the same in the general case as in Horn case. As before, a hypotheses H is derivable by KSS from B and e whenever it θ-subsumes a Kernel Set of B and e. The only difference is that general clauses are now used in place of Horn clauses, and the general Kernel Set replaces the Horn Kernel Set. As shown in Theorems 1 and 2, the key results from the Horn clause case apply also in the general case.

Definition 10 (KSS). *Let B and H be theories, and e be a clause. Then H is* derivable by KSS from B and e *iff $H \sqsupseteq \mathcal{K}$ for some Kernel Set $\mathcal{K}$ of B and e.*

Theorem 1 (Soundness of KSS wrt IE). *Let B and H be theories, let e be a clause, and let $\mathcal{K} = \{k_1 \wedge \ldots \wedge k_n\}$ be a Kernel Set of B and e. Then $H \sqsupseteq \mathcal{K}$ implies $B \cup H \models e$.*

Proof. By Definition 9 it holds that $\mathcal{B} \cup \{\lambda_1^0 \wedge ... \wedge \lambda_n^0\} \models \epsilon$. Therefore $B \cup \overline{e} \cup \{\lambda_1^0 \wedge ... \wedge \lambda_n^0\} \models \Box$ by Lemma 1. Hence $B \cup \overline{e} \models \neg\lambda_1^0 \vee ... \vee \neg\lambda_n^0$ by the Entailment Theorem. If $\mathcal{M}$ is any model of $B \cup \overline{e}$ then for some $1 \leq i \leq n$ it follows $\mathcal{M}$ falsifies the key literal λ_i^0. But, by an analogous argument, it also follows from Definition 9 and Lemma 1 that $B \cup \overline{e} \models \neg\lambda_1^j$ for all $1 \leq i \leq n$ and $1 \leq j \leq m_i$. Hence $\mathcal{M}$ also falsifies all of the non-key literals $\lambda_i^1, \ldots, \lambda_i^{m_i}$. Therefore $\mathcal{M}$ falsifies the Kernel clause $k_i = \lambda_i^0 \vee \lambda_i^1 \vee ... \vee \lambda_i^{m_i}$ and also, therefore, the Kernel theory $\mathcal{K} = \{k_1 \wedge \ldots \wedge k_n\}$. Consequently, $B \cup \overline{e} \cup \mathcal{K} \models \Box$. Since $H \sqsupseteq \mathcal{K}$, it follows $H \models \mathcal{K}$ and so $B \cup \overline{e} \cup H \models \Box$. Therefore $\mathcal{B} \cup H \models \epsilon$ by Lemma 1 part (1) and hence $B \cup H \models e$ by Lemma 1 part (2).

Theorem 2 (KSS Extends BG). *Let B be a theory, let $e = P_1 \vee ... \vee P_n \vee \neg N_1 \vee ... \vee \neg N_m$ be a clause, let $Bot(B, e)$ be the Bottom Set of B and e using σ, and let $h = L_0 \vee ... \vee L_p$ be a clause. Then a clause h is derivable by BG from B and e only if the theory $H = \{h\}$ is derivable by KSS from B and e.*

Proof. Suppose h is derivable by BG from B and e. Then $h \succeq Bot(B, e)$ by Definition 2, and so $h\theta \subseteq Bot(B, e)$ for some θ. By Definition 1 it holds $B \cup \overline{e} \models \overline{L_i}\theta$ for all $0 \leq i \leq p$. Since $L_i\theta$ is a ground atom, $\overline{L_i}\theta = \neg L_i\theta$ and so $B \cup \overline{e} \cup \{L_i\theta\} \models \Box$ by the Entailment Theorem. Consequently, $\mathcal{B} \cup \{L_i\theta\} \models \epsilon$ by Lemma 1. By Definition 9 the theory $\mathcal{K} = \{L_0\theta \vee L_1\theta \vee ... \vee L_p\theta\}$ is a single clause Kernel Set of B and e. By construction $\{h\} \sqsupseteq \mathcal{K}$ and hence it follows by Definition 10 that $H = \{h\}$ is derivable by KSS from B and e.

To show KSS is stronger than BG, simply let $B = \{p \vee \neg q(a) \vee \neg q(b)\}$ and $e = p$. In this case the three hypotheses $\{q(x)\}$ and $\{q(a) \wedge q(b)\}$ and $\{p\}$ are all derivable by KSS, but only the last one is derivable by BG from B and e. In order to illustrate the ideas presented above and to show how KSS can also derive non-Horn theories, this section now concludes with Example 1 below.

Example 1. Let the background theory B represent the knowledge that anyone in a *bar* may be served a *drink* unless he is a *child*, and that anyone in a *cafe* may be served a *drink*. Let the example e denote the fact that all *adults* may be served a *drink*.

$$B = \left\{ \begin{array}{l} drink \vee child \vee \neg bar \\ drink \vee \neg cafe \end{array} \right\} \qquad\qquad e = drink \vee \neg adult$$

Then the hypothesis H shown below, which states that an *adult* will go either to the *cafe* or to the *bar*, and that no one is both an *adult* and a *child*, is correct with respect to IE since it can be verified that $B \cup H \models e$.

$$H = \left\{ \begin{array}{l} bar \vee cafe \vee \neg adult \\ \neg child \vee \neg adult \end{array} \right\}$$

Using the abbreviations $a = adult, b = bar, c = child, d = drink, f = cafe$ it can be verified that the result of normalising B and e consists of the theory $\mathcal{B}$ and the clause ϵ shown below, and that the following sequents are true.

$$\mathcal{B} = \left\{ \begin{array}{l} d \vee c \vee \neg b \\ d \vee \neg f \\ a \end{array} \right\} \qquad \epsilon = d \qquad \begin{array}{l} \mathcal{B} \cup \{b \wedge \neg c\} \models \epsilon \\ \mathcal{B} \cup \{f\} \models \epsilon \\ \mathcal{B} \cup \{\neg a\} \models \epsilon \end{array}$$

Therefore, by Definitions 9 and 10, theory H is a Kernel Set of B and e, with key literals b and $\neg c$, and H is derivable by KSS from B and e. But note $Bot(B, e) = drink \vee cafe \vee \neg adult$ and so neither clause in H is derivable by BG.

4 Characterisation in Terms of K*-Derivations

This section provides a sound and complete characterisation of Kernel Sets in terms of a new derivation, called a *K*-derivation.* The K*-derivation is at the same time a generalisation of the C-derivation and a refinement of the K-derivation. Where a K-derivation requires that each clause in a set K is used at most once, a K*-derivation imposes one additional restriction on the way such clauses are used. The basis of this restriction is a new notion, called *T-reduction*, formalised in Definition 11 below and illustrated in Fig 1.

Definition 11 (T-reduction). *Let T be a theory, and let C and D be clauses. Then a* T-reduction of C to D *is a C-derivation $\mathcal{R}$ of D from T with respect to C such that $\mathcal{R} = \mathcal{R}^1 + ... + \mathcal{R}^m + (C{=}C^0, ..., C^m{=}D)$ where for all $1 \le i \le m$ each $\mathcal{R}^i$ is a tree derivation of a clause E^i from T, and C^i is a resolvent of C^{i-1} with a unit factor of E^i. The clause C is said to be reduced to D by T in $\mathcal{R}$.*

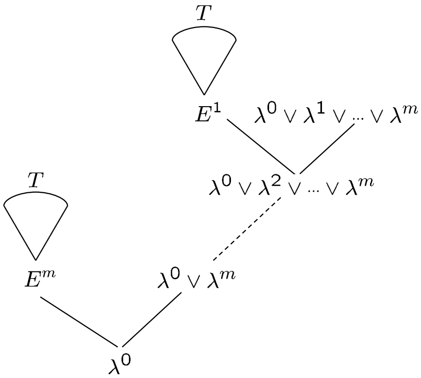
T
E^1
$\lambda^0 \vee \lambda^1 \vee ... \vee \lambda^m$
$\lambda^0 \vee \lambda^2 \vee ... \vee \lambda^m$
T
E^m
$\lambda^0 \vee \lambda^m$
λ^0

Fig. 1. T-reduction

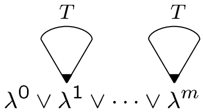

Fig. 2. Abbreviation of Fig 1

Informally, T-reduction is the process of progressively resolving a clause C with clauses derived from a theory T. Fig 1 illustrates the T-reduction of a ground clause $C = C^0 = \lambda^0 \vee \lambda^1 \vee ... \vee \lambda^m$ to the ground literal $\lambda^0 = C^m = D$. Each descendent $C^i = \lambda^0 \vee \lambda^{i+1} \vee ... \vee \lambda^m$ of C differs from predecessor C^{i-1} in the removal of a literal λ^i, which is 'resolved away' by a clause E^i derived from T. Each wedge denotes the tree derivation $\mathcal{R}^i$ of the clause E^i shown at the base of the wedge, from the theory T shown at the top.

To simplify the representation of the T-reduction of a ground clause to a single literal, it is convenient to introduce the graphical abbreviation shown in Fig 2. Just as in Fig 1, the wedges denote the tree derivations $\mathcal{R}^i$ of the clauses E^i. But now, instead of the clause E^i, the complementary literal λ^i appears at the base each wedge. The black tip of each wedge emphasises that it is not the literal λ^i which is derived, but the clause E^i that resolves away λ^i. Intuitively, this graphic shows the literals of C being resolved away until only λ^0 remains.

The notion of T-reduction is now used to define the concept of K*-derivation. As formalised in Definition 12, a K*-derivation consists of a *principal derivation*, which is a tree derivation of some clause C_0 from T, and zero or more *reduction trees*, in each of which a k-input clause k_i is reduced to a clause D_i by T. Clause C_0 is then reduced to D in the 'tail' $(C_1, ..., C_n)$ of the derivation, by resolving with each of the D_i in turn. For an example, the reader is referred to Fig 5.

Definition 12 (K*-derivation). *Let T and K be theories, and let D be a clause. Then a* K*-derivation of D from T with respect to K *is a K-derivation $\mathcal{R}$ of D from T with respect to K such that $\mathcal{R} = \mathcal{R}_0 + \mathcal{R}_1 + ... + \mathcal{R}_n + (C_1, ..., C_n{=}D)$ where $\mathcal{R}_0$ is a tree derivation, called the* principal derivation*, of a clause C_0 from T, and for all $1 \leq i \leq n$ each $\mathcal{R}_i$ is a T-reduction, called a* reduction tree*, of a k-input clause $k_i \in K$ to a clause D_i, and each C_i is the resolvent of C_{i-1} with a unit factor of D_i. The clause D is said to be* K*-derivable *from T with respect to K, denoted $(T, K) \vdash_{k*} D$, whenever such a derivation exists.*

The rest of this section shows how the K*-derivation provides a sound and complete characterisation of KSS. First, Lemma 2 demonstrates an important relationship between abduction and K*-derivations. Informally, this result states that all of the ground facts used in a refutation, need be used just once, at the very end. Such a refutation $\mathcal{R}$ is shown in Fig 3, where the principal derivation $\mathcal{R}_0$ is denoted by the wedge, and $\lambda'_1, \ldots, \lambda'_p$ are the ground facts used.

Lemma 2. *Let T be a theory, and let $\Delta = \{\lambda_1 \wedge ... \wedge \lambda_n\}$ be a ground unit theory. Then $T \cup \Delta \models \Box$ implies $(T, \Delta) \vdash_{k*} \Box$ with a K*-derivation of the form $\mathcal{R} = \mathcal{R}_0 + (\lambda'_1, ..., \lambda'_p) + (C_1, ..., C_p{=}\Box)$ for some $\{\lambda'_1 \wedge ... \wedge \lambda'_p\} \subseteq \Delta$.*

Proof. Suppose that $T \cup \Delta \models \Box$. Then by the Entailment Theorem $T \models D$ where $D = \overline{\lambda}_1 \vee ... \vee \overline{\lambda}_n$. If D is a tautology, then Δ contains two complementary unit clauses, which means there is a trivial tree derivation of $\Box$ from these two unit clauses, and so $(T, \Delta) \vdash_{k*} \Box$ by Definition 12. If D is not a tautology, then by the Subsumption Theorem there is a tree derivation $\mathcal{R}_0$, from the theory T of a clause C_0 such that $C_0\theta \subseteq D$ for some substitution θ. Now, define the set $\Delta' = \{\lambda'_1, ..., \lambda'_p\}$ such that $\Delta' = \overline{C_0}\theta \cap \Delta$. (i.e. Δ' is the subset of Δ whose literals are complementary to those in C_0 under θ). Next, let $(C_1, \ldots, C_p{=}\Box)$ be the (unique) sequence of clauses $C_i = M_i^0 \vee M_i^1 \vee ... \vee M_i^{q_i}$ such that each clause $C_{i+1} \in (C_1, \ldots, C_p)$ is the resolvent of C_i and λ'_{i+1} on the factor $C_i\phi_i$ of C_i where ϕ_i is the mgu of the set $S_i = \{M_i^j \in C_i \mid M_i^j\theta = \lambda'_{i+1}\}$. (i.e. S_i is the subset of C_i whose literals are complementary to λ'_{i+1} under θ). Then, by Definition 12, it follows $\mathcal{R} = \mathcal{R}_0 + (\lambda'_1, ..., \lambda'_p) + (C_1, ..., C_p{=}\Box)$ is a K*-refutation from T with respect to Δ, with principal derivation $\mathcal{R}_0$ and trivial reduction trees where each $\mathcal{R}_i = (\lambda'_i)$ simply contains the unit clause $\lambda'_i \in \Delta$ (see Fig 3).

Lemma 2 is now used in Theorem 3 to show that a theory $\mathcal{K}$ is a Kernel Set of B and e if and only if there exists a K*-refutation from $B \cup \overline{e}$ with respect to $\mathcal{K}$. Informally, as illustrated in Fig 4, a K*-derivation can always be constructed in which (zero or more) Kernel clauses are reduced by the theory $B \cup \overline{e}$ to their key literals. The key literals are then used one by one in the tail of the derivation to resolve away the clause C_0 derived in the principal derivation.

Theorem 3 (K*-derivations characterise Kernel Sets). *Let $\mathcal{B}$ and ϵ be the result of normalising a theory B and clause e using σ. Let $\overline{e}$ be the complement of e, also using σ. Let $\mathcal{K} = \{k_1 \wedge ... \wedge k_n\}$ be a theory of ground clauses written $k_i = \lambda_i^0 \vee \lambda_i^1 \vee ... \vee \lambda_i^{m_i}$, and let $\Delta = \{\lambda_1^0 \wedge ... \wedge \lambda_n^0\}$. Then the theory $\mathcal{K}$ is a Kernel Set of B and e iff $(B \cup \overline{e}, \mathcal{K}) \vdash_{k*} \Box$*

Proof. Taking the "if" and "only if" cases individually:

1. Suppose $(B \cup \overline{e}, \mathcal{K}) \vdash_{k*} \Box$. Then by Definition 12 for some $0 \leq p \leq n$ there is a K*-derivation $\mathcal{R} = \mathcal{R}_0 + \mathcal{R}_1 + ... + \mathcal{R}_p + (C_1, ..., C_p{=}\Box)$ with principal derivation $\mathcal{R}_0$ and reduction trees $\mathcal{R}_1, \ldots, \mathcal{R}_p$. By Definition 11 it follows for all $1 \leq i \leq p$ that $\mathcal{R}_i = \mathcal{R}_i^1 + ... + \mathcal{R}_i^{m_i} + (k_i{=}C_i^0, ..., C_i^{m_i}{=}\lambda'_i)$ is a T-reduction of a ground k-input clause $k_i \in \mathcal{K}$ to single literal $\lambda'_i \in k_i$ by the theory $B \cup \overline{e}$. Without loss of generality, assume $\mathcal{K}$ has been written so $\mathcal{K} = \{k_1 \wedge ... \wedge k_p \wedge ... \wedge k_n\}$ and $k_i = \lambda_i^0 \vee \lambda_i^1 \vee ... \vee \lambda_i^{m_i}$ with $\lambda'_i = \lambda_i^0$ for all

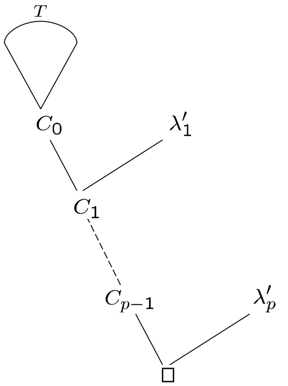

Fig. 3. Abductive Derivation

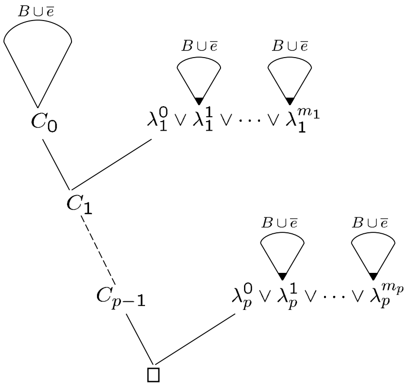

Fig. 4. K*-derivation

$1 \leq i \leq p$. Therefore $\mathcal{R}' = \mathcal{R}_0 + (\lambda_1^0, ..., \lambda_p^0) + (C_1, ..., C_p{=}\Box)$ is a tree refutation from $B \cup \overline{e} \cup \Delta$. Hence $B \cup \overline{e} \cup \Delta \models \Box$ by the soundness of resolution, and so $\mathcal{B} \cup \Delta \models \epsilon$ by Lemma 1. Now, by Definition 11 each $\mathcal{R}_i^j$ with $1 \leq i \leq p$ and $1 \leq j \leq m_i$ is a tree derivation of a clause D_i^j from $B \cup \overline{e}$ such that D_i^j resolves away λ_i^j. Therefore $\mathcal{R}_i^j + (\lambda_i^j) + (\Box)$ is a tree refutation from $B \cup \overline{e} \cup \{\lambda_i^j\}$. Hence $B \cup \overline{e} \cup \{\lambda_i^j\} \models \Box$ by soundness of resolution, and so $\mathcal{B} \cup \{\lambda_i^j\} \models \epsilon$ by Lemma 1. Since $\mathcal{B} \cup \Delta \models \epsilon$ and $\mathcal{B} \cup \{\lambda_i^j\} \models \epsilon$ it follows by Definition 9 that $\mathcal{K}$ is a Kernel Set of B and e with key literals Δ.

2. Suppose $\mathcal{K}$ is a Kernel Set of B and e with key literals Δ. By Definition 9 it follows $\mathcal{B} \cup \Delta \models \epsilon$. Consequently, $B \cup \overline{e} \cup \Delta \models \Box$ by Lemma 1. By Lemma 2 there is a K*-refutation $\mathcal{R} = \mathcal{R}_0 + (\lambda'_1, ..., \lambda'_p) + (C_1, ..., C_p{=}\Box)$ from $B \cup \overline{e}$ for some $\{\lambda'_1 \wedge ... \wedge \lambda'_p\} \subseteq \Delta$ (see Fig 3). Without loss of generality, assume $\mathcal{K}$ has been written so $\mathcal{K} = \{k_1 \wedge ... \wedge k_p \wedge ... \wedge k_n\}$ and $k_i = \lambda_i^0 \vee \lambda_i^1 \vee ... \vee \lambda_i^{m_i}$ with $\lambda'_i = \lambda_i^0$ for all $1 \leq i \leq p$. By Definition 9 for each $1 \leq j \leq m_i$ it follows $\mathcal{B} \cup \{\lambda_i^j\} \models \epsilon$. Consequently $B \cup \overline{e} \cup \{\lambda_i^j\} \models \Box$ by Lemma 1. By Lemma 2 there is a K*-refutation $\mathcal{R}_i^j + (\lambda_i^j) + (\Box)$ in which λ_i^j may or may not be used. If λ_i^j is not used, then by Definition 12 it trivially follows $(B \cup \overline{e}, \mathcal{K}) \vdash_{k*} \Box$ and the theorem is proved. If λ_i^j is used, then by Definition 12 it follows $\mathcal{R}_i^j$ is a tree derivation from $B \cup \overline{e}$ of a clause E_i^j that resolves away λ_i^j. Now, for all $1 \leq i \leq p$ and $0 \leq j \leq m_i$ define the clause $C_i^j = \lambda_i^0 \vee ... \vee \lambda_i^j$ and the derivation $\mathcal{R}_i = \mathcal{R}_i^1 + ... + \mathcal{R}_i^{m_i} + (k_i{=}C_i^{m_i}, ..., C_i^1, C_i^0{=}\lambda_i^j)$. Then by Definition 11 it follows $\mathcal{R}_i$ is a tree derivation in which k_i is T-reduced to λ_i^j by $B \cup \overline{e}$. Hence by Definition 11 the tree derivation $\mathcal{R}' = \mathcal{R}_0 + \mathcal{R}_1 + ... + \mathcal{R}_p + (C_1, ..., C_p{=}\Box)$ is a K*-refutation from $B \cup \overline{e}$ with respect to $\mathcal{K}$ (see Fig 4). Thus $(B \cup \overline{e}, \mathcal{K}) \vdash_{k*} \Box$ by Definition 12.

To characterise the *hypotheses* derivable by KSS in terms of K*-derivations, one complication must be addressed. Given that $H \sqsupseteq \mathcal{K}$, it is possible for one clause $h \in H$ to θ-subsume more than one clause in $\mathcal{K}$, so that more than one instance of h is needed to derive $\Box$ from $B \cup \overline{e}$ and H. For example, let

$B = \{p \vee \neg q(a) \vee \neg q(b)\}$ and $e = p$. Then hypothesis $H = p(X)$ subsumes the Kernel Set $\mathcal{K} = \{q(a) \wedge q(b)\}$, and *two* instances of H are required.

One way of handling this complication is by treating the hypothesis H as a *multi-set*, and treating the relation $H \sqsupseteq \mathcal{K}$ as an *injection* that maps each clause k in set $\mathcal{K}$ to a clause h in multi-set H such that $h \succeq k$. Then it can be shown that a theory H is derivable by KSS from a theory B and a clause e, if and only if there is a K*-refutation from $B \cup \overline{e}$ with respect to H. For completeness the proof of this result is sketched in Corollary 1 below.

Corollary 1. *Let B be a set of clauses, let e be a clause, and let H be a multi-set of clauses. Then H is derivable by KSS from B and e iff* $(B \cup \overline{e}, H) \vdash_{k*} \Box$

Proof. (Sketch)

1. Suppose H is derivable by KSS from B and e. Then there is a theory $\mathcal{K}$ such that $H \sqsupseteq \mathcal{K}$ and $\mathcal{K}$ is a Kernel Set of B and e. By Theorem 3 there is a K*-refutation from $B \cup \overline{e}$ with respect to $\mathcal{K}$. Now replace each reduction tree of a k-input clause k by the reduction tree of a fresh variant of the clause h to which k is mapped by $\sqsupseteq$. After appropriate syntactic changes, the result is a K*-refutation from $B \cup \overline{e}$ with respect to H.
2. Suppose such a K*-derivation exists. Replace each reduction tree of a k-input clause h by the reduction tree of a ground instance of k of h consistent with the substitutions in the derivation. After appropriate syntactic changes, the result is a K*-refutation from $B \cup \overline{e}$ with respect to the clauses k. By Theorem 3 this set of clauses is a Kernel Set of B and e, and by construction it is θ-subsumed by H. Therefore H is derivable by KSS from B and e.

In order to illustrate the concepts introduced above, this section concludes by presenting a full K*-derivation for Example 1. As shown in Fig 5, the reduction trees of the two underlined hypothesis clauses to the literals b and $\neg c$ are indicated by the dashed triangles. The principal derivation of the clause $c \vee \neg b$ containing the complements of these literals is marked by the dashed rectangle. The tail of the derivation is shown by the dashed ellipse. Finally, Fig 6 shows how this derivation is abbreviated using the compact notation introduced in Fig 2.

5 Related Work

The semantics of KSS is aimed ultimately at extending the principles of BG in order to provide a more general context for developing practical proof procedures for IE. However, it can also be used as a means of systematically comparing existing methods for BG and KSS. In particular, as shown in the previous section, independently of how a hypothesis H is actually computed from B and e, there will always be an associated K*-refutation from $B \cup \overline{e}$ with respect to H. Consequently, existing methods can be classified in terms of the restrictions needed on the principal and reduction derivations in order to characterise the class of derivable hypotheses. Several methods are now compared in this way.

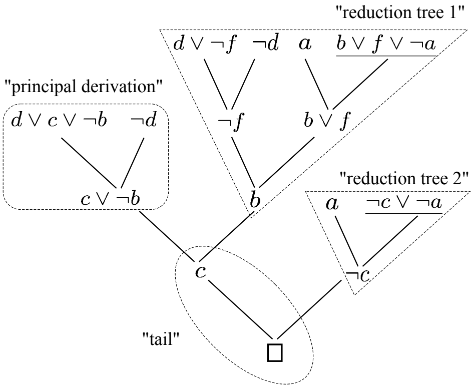

Fig. 5. K*-derivation of Example 1

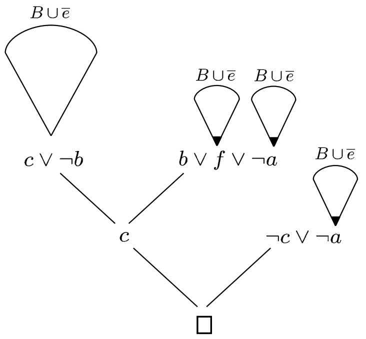

Fig. 6. Abbreviation of Fig 5

Progol4 [6] is one of the best known and widely applied systems in ILP. This procedure uses a methodology called *Mode Directed Inverse Entailment (MDIE)* that efficiently implements BG the use of user-specified language bias. A subset of the Bottom Set is constructed by an SLD procedure and is then generalised by a lattice based search routine. Like all BG approaches, Progol4 induces only a single clause for each example, and like most existing procedures it is restricted to Horn clause logic. The hypotheses derivable by Progol4 are associated with K*-derivations of a simple form. The principal derivation always consists of a single unit clause containing the (unique) negative literal from $\overline{e}$; and this literal is always resolved away by a single reduction tree in which the (only) hypothesis clause is reduced to an instance of its head atom.

Progol5 [7] is the latest member of the Progol family. This proof procedure realises a technique called *Theory Completion by Inverse Entailment (TCIE)* that augments MDIE with a reasoning mechanism based on *contrapositive locking* [13]. Although the principal derivations associated with Progol5 hypotheses also result in a negative unit clause, unlike Progol4 they may involve the non-trivial derivation of a negative literal distinct from that in $\overline{e}$. However, due to an incompleteness of the contrapositive reasoning mechanism identified in [12], no *merging* of literals may occur within the principal derivation. In the corresponding reduction tree, the single hypothesis clause is reduced to an instance of its head atom.

HAIL [11] is a recently proposed proof procedure that extends TCIE with the ability to derive multiple clause hypotheses within the semantics of KSS. This procedure is based on an approach called *Hybrid Abductive-Inductive Learning* that integrates explicit abduction, deduction and induction, within a cycle of learning that generalises the mode-directed approach of Progol5. Key literals of the Kernel Set are computed using an ALP proof procedure [5], while non-key literals are computed using the same SLD procedure used by Progol. Like Progol5, HAIL is currently restricted to Horn clause logic, but unlike Progol, the hypotheses derivable by HAIL can give rise to K*-derivations in which there is no restriction on merging, and where the principal derivation may result in a negative clause with more than one literal. Each of these literals is resolved away by a corresponding reduction tree, in which one of the hypothesis clauses is reduced to an instance of its head atom.

The proof procedures discussed above use the notions of Bottom Set or Kernel Set to deliberately restrict their respective search spaces. This is in contrast to some other recent approaches that attempt to search the complete IE hypothesis space. A technique based on *Residue Hypotheses* is proposed in [16] for Hypothesis Finding in general clausal logic. In principle, this approach subsumes all approaches for IE - including BG and KSS - because no restrictions are placed on B, H or e. But, in practice, it is not clear how Residue Hypotheses may be efficiently computed, or how language bias may be usefully incorporated into the reasoning process. An alternative method, based on *Consequence Finding* in full clausal logic, is proposed in [3] that supports a form of language bias called a *production field* and admits pruning strategies such as *clause ordering*. However, this procedure is still computationally expensive and has yet to achieve the same degree of practical success as less complete systems such as Progol.

6 Conclusion

In this paper the semantics of KSS has been extended from Horn clauses to general clausal logic. It was shown in the general case that KSS remains sound with respect to the task of IE and that it continues to subsume the semantics of BG. In addition, an extension of Plotkin's C-derivation, called a K*-derivation, was introduced and shown to provide a sound and complete characterisation of the hypotheses derivable by KSS in the general case. These results can be seen as extending the essential principles of BG in order to enable the derivation multiple clause hypotheses in general clausal logic and thereby enlarging the class of soluble problems.

The aim of this work is to provide a general context in which to develop practical proof procedures for IE. It is believed such procedures can be developed for KSS through the integration ALP and ILP methods and the efficient use of language bias. A hybrid ALP-ILP proof procedure has been proposed in [12] for computing multiple clause hypotheses, but currently this procedure is restricted to Horn clauses and has not been implemented. To address these issues, efficient abductive and inductive procedures are required for general clausal logic. One promising approach would be to adapt the work already begun by [1] and [3] in the context of semantic tableaux and to apply them in the context of KSS.

References

1. A. Aliseda. *Seeking Explanations: Abduction in Logic, Philosophy of Science and Artifical Intelligence.* PhD thesis, Institute for Logic, Language and Computation (ILLC), University of Amsterdam, 1997.

2. C. Chang and R.C. Lee. *Symbolic Logic and Mechanical Theorem Proving*. Academic Press, 1973.
3. 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.
4. A.C. Kakas, R.A. Kowalski, and F. Toni. Abductive Logic Programming. *Journal of Logic and Computation*, 2(6):719–770, 1992.
5. 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.
6. S.H. Muggleton. Inverse Entailment and Progol. *New Generation Computing, Special issue on Inductive Logic Programming*, 13(3-4):245–286, 1995.
7. S.H. Muggleton and C.H. Bryant. Theory Completion Using Inverse Entailment. *Lecture Notes in Computer Science*, 1866:130–146, 2000.
8. S.H. Muggleton and L. De Raedt. Inductive Logic Programming: Theory and Methods. *Journal of Logic Programming*, 19,20:629–679, 1994.
9. S.H. Nienhuys-Cheng and R. de Wolf. *Foundations of Inductive Logic Programming*, volume 1228 of *Lecture Notes in Artificial Intelligence*. Springer, first edition, 1997.
10. G.D. Plotkin. *Automatic Methods of Inductive Inference*. PhD thesis, Edinburgh University, 1971.
11. O. Ray. HAIL: Hybrid Abductive-Inductive Learning. Technical Report 2003/6, Department of Computing, Imperial College London, 2003.
12. O. Ray, K. Broda, and A. Russo. Hybrid Abductive Inductive Learning: a Generalisation of Progol. In T. Horváth and A. Yamamoto, editors, *13th International Conference on Inductive Logic Programming*, volume 2835 of *Lecture Notes in AI*, pages 311–328. Springer Verlag, 2003.
13. M.E. Stickel. A Prolog technology theorem prover: Implementation by an extended Prolog compiler. In J. H. Siekmann, editor, *Journal of Automated Reasoning*, volume 4(4), pages 353–380, 1988.
14. 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.
15. A. Yamamoto. An Inference Method for the Complete Inverse of Relative Subsumption. *New Generation Computing*, 17(1):99–117, 1999.
16. A. Yamamoto and B. Fronhöfer. Hypothesis Finding via Residue Hypotheses with the Resolution Principle. In *Proceedings of the 11th International Conference on Algorithmic Learning Theory*, volume 1968 of *Lecture Notes in Computer Science*, pages 156–165. Springer-Verlag, 2000.