Logique & Analyse **137-138** (1992), 189-205

A LOGIC-BASED MODELLING OF PROLOG RESOLUTION SEQUENCES

John S. JEAVONS and John N. CROSSLEY

Introduction

In logic programming a clear distinction must be made between the declarative semantics associated with a logic program and the procedural semantics (see for example Lloyd [4]). The declarative semantics associates a set of sentences in formal logic to a given program, whereas the procedural semantics gives a description of how the interpreter searches for a deduction. In languages like PROLOG the deduction procedure employed consists of a sequence of applications of SLD-resolution steps. In this paper we work from the procedural aspect; we first introduce a formal logical system and then we specify an injective map from the set of all finite SLD-derivations into the set of provable sequents of the formal system. From this we are able to provide a logic-based model for the procedural aspect of resolution-based logic programming. Thus we are not addressing the declarative semantics here; we are specifying what may be called the 'internal' logic of the program.

To clarify the differences between the declarative semantics (which we may regard as the external logical content) and the proposed logic-based semantics (for the internal deduction procedure), we first consider a propositional program $\mathcal{P}$ consisting of the five clauses:

1. p :- q, p
2. q :- q
3. p
4. q :- p
5. p :- r.

The declarative content is a set of sentences in classical propositional logic, $S = \{p\&q \Rightarrow p,\ q \Rightarrow q,\ p,\ p \Rightarrow q,\ r \Rightarrow p\}$. The queries ? - p, ? - q, ? - p, q and ? - p, p are all supported since each one of the sentences p, q, $p\&q$ and $p\&p$ is a logical consequence of the set S. Nevertheless, standard PROLOG fails to return an answer for any of these queries if the program clauses are written in the order specified.

Consider now the procedure for an SLD-refutation of $\mathcal{P} \cup \{\leftarrow p, p\}$, as distinct from an SLD-refutation of $\mathcal{P} \cup \{\leftarrow p\}$. The corresponding two resolution sequences are quite distinct entities, although in the declarative semantics the formulae $p\&p$ and p are logically equivalent, so that from $S \models p\&p$ we can infer $S \models p$ and *vice-versa*. The sequence below displays an SLD-refutation of $\mathcal{P} \cup \{\leftarrow p, p\}$, where the selected atom in a query is indicated by underlining, and where the selected program clause is indicated by the number above the arrow

$$? - \underline{p}, p \overset{3}{\rightarrow} ? - \underline{p} \overset{1}{\rightarrow} ? - \underline{q}, p \overset{4}{\rightarrow} ? - \underline{p}, p \overset{3}{\rightarrow} ? - \underline{p} \overset{3}{\rightarrow} \text{true.}$$

This SLD-refutation is not, *per se*, an SLD-refutation of $\mathcal{P} \cup \{\leftarrow p\}$, although if we delete the first query we then obtain such a derivation. From the procedural viewpoint our logic-based semantics for the internal deduction process must distinguish between two such refutations. We propose to associate a provable formula with a finite SLD-derivation in such a manner that corresponding to two distinct derivations we have two formulae which are not logically equivalent. In particular, the formula associated with a query $? - p, p$ will not be logically equivalent to the formula associated with the query $? - p$, so the PROLOG comma will no longer be interpreted as an idempotent connective (like &) in classical logic. We introduce a new connective$\bullet$(see next section) for the logical connective corresponding to the comma in PROLOG clauses. In the proposed logic a sequent $A_1, \ldots, A_n \vdash B$ corresponds to a formula $(A_1 \bullet \ldots \bullet A_n) \supset B$, so that the comma in the left-hand side of the sequent corresponds to our new connective$\bullet$. Now in most logics the sequent $A \vdash A$ is provable, and in our logic the sequent $A, A \vdash A \bullet A$ is provable but we cannot contract on the left-hand side to obtain a proof of $A \vdash A \bullet A$. Similarly, we do not allow weakening, the sequent $A \vdash A$ is provable, but we cannot weaken on the left side to obtain a proof of $A, B \vdash A$, for an arbitrary formula B. The next sequence shows a failed attempt for the query

$$? - p. \; ? - \underline{p} \overset{1}{\rightarrow} ? - \underline{q}, p \overset{4}{\rightarrow} ? - p, \underline{p} \overset{5}{\rightarrow} ? - \underline{p}, r \overset{3}{\rightarrow} ? - \underline{r}$$

The choice of program clause 5. ensures that the resolution sequence can never terminate with true, yet p is clearly a logical consequence (in the external sense) of the set of selected program clauses. This example reinforces our view that in dealing with the internal logic we must indeed consider a formal system with no weakening because weakening allows the introduction of fake dependencies, but in a resolution proof every selected program clause is relevant to the proof.

Finally, the connective • will be non-commutative since a derivation sequence for the query ? - p, q via some computation rule is not, *a priori*, a derivation sequence for the query ? - q, p. The computation rule must also have a semantic interpretation in the proposed internal logic. Lloyd [4] points out that if $\mathcal{P}$ is a definite program and if $\mathcal{P} \cup \{\leftarrow A\}$ has an SLD-refutation via a computation rule $\mathfrak{R}$, then $\mathcal{P} \cup \{\leftarrow A\}$ also has a SLD-refutation via any other computation rule $\mathfrak{R}'$. However, from the viewpoint of the internal logic the two refutations do not correspond to the same deduction of A.

In classical logic the expression $A_1, \ldots, A_n \vdash B$ may either be a sequent (using $\vdash$ as a separator of the right-hand side from the left-hand side), or the expression may stand for the assertion that from the axioms $A_1, \ldots, A_n$ we can deduce B (in some formal system). The two notions are related since in most traditional logics we have that the sequent $A_1, \ldots, A_n \vdash B$ is provable if, and only if, we can prove the sequent $\vdash B$ from the axiom sequents $\vdash A_1, \ldots, \vdash A_n$. In our proposed logic the two notions no longer coincide and to avoid confusion we will write sequents of our formal system as $A_1, \ldots, A_n = \vdash B$.

To each finite SLD-resolution sequence we will associate a sequent $\Gamma = \vdash A$, where Γ is a *sequence* of formulae. Each member of Γ corresponds either to an application of the computation rule, or to a selected program clause, or to a re-arrangement of the atoms in a goal clause. If we have an SLD-refutation of $\mathcal{P} \cup \{\leftarrow A, B\}$ the associated sequent is of the form $\Gamma = \vdash A \bullet B$, where we note that the Prolog comma is not interpreted as &. If the resolution sequence for the goal $\leftarrow A, B$ fails finitely with final goal $\leftarrow \underline{C}, D, E$, say, then the associated sequent is of the form $\Gamma = \vdash (C \bullet D \bullet E) \supset (A \bullet B)$ which can be interpreted as 'after the sequence of steps represented by Γ, in order to deduce $A \bullet B$ we need a deduction of $C \bullet D \bullet E$'.

In order to represent an infinite SLD derivation we would require an infinitary logic. In this paper we specify only a finitary logic, together with its semantics. We indicate at the end of this paper how such an infinitary version of the logic may be developed, but the details are not included as the precise formulation of the infinitary version is much more cumbersome than for the finitary case.

The rest of the paper is as follows. Section 1.1 specifies the syntax for the logical system. In section 1.2 we give a Kripke-type semantics in terms of monoids with a partial order relation. A soundness and completeness result is established in section 1.3. In section 2 we give the translation of an SLD-resolution step into the logic and extend this to the specification of any finite SLD-derivation in the logic.

1.1 *The formal system* L_I

Our logic for representing the internal logic of a resolution based inference system will be denoted by L_I. The basic alphabet contains the following symbols:

(i) a countable set of variables $x, y, \ldots$;
(ii) function symbols $f, g, \ldots$;
(iii) predicate symbols $P, Q, \ldots$;
(iv) propositional constants $\bot$ and $\mathbf{1}$;
(v) binary connectives $\bullet$ and $\supset$;
(vi) universal quantifier $\forall$;
(vii) punctuation symbols (,).

The terms of L_I and the well-formed formulae of L_I are specified in the usual inductive manner.

The axioms and inference rules are given below in sequent form $\Delta = \vdash A$, where A is a L_I formula and upper case Greek letters stand for finite sequences of formulae.

Axioms

$$A = \vdash A \qquad \text{(for any formula } A\text{)}$$

$$\Delta, \bot, \Sigma = \vdash A \qquad \text{(for any sequences } \Delta, \Sigma \text{ and any formula } A\text{)}$$

$$= \vdash \mathbf{1}$$

Inference Rules

(1 - weakening) $$\frac{\Delta, \Sigma = \vdash A}{\Delta, \mathbf{1}, \Sigma = \vdash A}$$

(cut) $$\frac{\Gamma = \vdash A \quad \Delta, A, \Sigma = \vdash B}{\Delta, \Gamma, \Sigma = \vdash \mathrm{B}}$$

($\supset$-*R*) $$\frac{\Gamma, A = \vdash B}{\Gamma = \vdash A \supset B}$$

($\supset$-L) $$\frac{\Gamma = \vdash A \quad \Delta, B, \Sigma = \vdash C}{\Delta, A \supset B, \Gamma, \Sigma = \vdash C}$$

(•-R) $\dfrac{\Gamma = \vdash A \quad \Delta = \vdash B}{\Gamma, \Delta = \vdash A \bullet B}$ (•-L) $\dfrac{\Gamma, A, B, \Sigma = \vdash C}{\Gamma, A \bullet B, \Sigma = \vdash C}$

($\forall$-R) $\dfrac{\Gamma = \vdash A(y)}{\Gamma = \vdash \forall x\, A(x)}$ ($\forall$-L) $\dfrac{\Gamma, A(t), \Delta = \vdash C}{\Gamma, \forall x\, A(x), \Delta = \vdash C}$

The rule ($\forall$-R) is subject to the eigenvariable condition that y does not occur free in any formula of Γ; in the rule ($\forall$-L) t may be any term.

We say that a formula A is *provable* (or is a *theorem*) if, and only if, the sequent $= \vdash A$ is provable. It is easy to show that the sequent $A_1, \ldots, A_n = \vdash B$ is provable if, and only if, $A_1 \supset (A_2 \supset \ldots (A_n \supset B) \ldots)$ is provable, and this formula is provable if, and only if, $(A_1 \bullet (A_2 \bullet (\ldots \bullet A_n) \ldots)) \supset B$ is provable. We do not have to be particular with the bracketing when writing $(A_1 \bullet \ldots \bullet A_n)$ as $\bullet$ is associative, but we do of course when we write $A_1 \supset (A_2 \supset \ldots)$.

The formula $((A \supset B) \bullet A) \supset B$ is a theorem (since $A \supset B$, A $= \vdash B$ is easily seen to be provable), but the formula $(A \bullet (A \supset B)) \supset B$ is not a theorem as $\bullet$ is not commutative.

The system L_I enjoys the cut-elimination property; this can be established in a similar manner to that given by Gentzen [1] for classical logic, but the proof in our case is actually easier due to the absence of structural rules in L_I.

The system presented here is in fact a subsystem of a logic presented by Komori [2]. We do not include the connectives &, $\vee$ (or the quantifier $\exists$) and this results in a simpler semantics for L_I than for the logic given by Komori which includes these connectives. Our logic L_I is sound with respect to the structures presented in Komori [2]. Moreover, since Komori establishes a cut-elimination result for his logic, it follows by the sub-formula property that if a sequent is not provable in L_I then it is not provable in Komori's logic. From this it follows that L_I is also complete with respect to Komori's semantics. In sections 1.2 and 1.3 below we establish a soundness and completeness result for L_I with respect to a simpler class of structures than the structures employed by Komori (see the remarks at the end of section 1.3).

1.2 *Semantics*

Our semantics for L_I is a Kripke [3] style semantics where we have "states in the progress of knowledge". We first define a partially ordered monoid structure by $\mathbf{M} = \langle M, \cdot, 1, \leq, \infty \rangle$ where M is a set and

(i) $\langle M, \cdot, 1 \rangle$ is a monoid with binary operation $\cdot$ and identity 1;

(ii) $\langle M, \leq, \infty \rangle$ is a partially ordered set such that $m_1 \leq m_2$ implies that, for all $m \in M$, $m_1 \cdot m \leq m_2 \cdot m$ and $m \cdot m_1 \leq m \cdot m_2$. The element ∞ is the greatest element of M and for all $m \in M$ we have $m \cdot \infty = \infty \cdot m = \infty$.

We now introduce a universe U which is a non-empty set. A *pre-interpretation* for the language of L_I is a pair $\mathbf{U} = \langle U, F_f : f$ is a function symbol of $L_I \rangle$ such that if f is an n-ary function symbol then $F_f : \mathrm{U}^n \rightarrow U$. The triple $\langle \mathbf{M}, M, \mathbf{U} \rangle$ is called a *frame*. Denote by $L_I[U]$ the expanded alphabet of L_I formed by adjoining a set of new constant symbols, $\{\underline{u} : u \in U\}$. We can now define a *valuation* on the frame $\langle \mathbf{M}, M, \mathbf{U} \rangle$ as a relation $\models$ between elements of M and closed atomic formulae of $L_I[U]$ which is to satisfy the conditions:

(i) $m \models \bot$ if, and only if, $m = \infty$ and $\infty \models P(t_1, \ldots, t_n)$ for each closed atomic formula of L_I[U],
(ii) $m \models \mathbf{1}$ if, and only if, $1 \leq m$,
(iii) For any closed formula $P(t_1, \ldots, t_n)$ of $L_I[U]$, if $m \models P(t_1, \ldots, t_n)$ and $m \leq m'$, then $m' \models P(t_1, \ldots, t_n)$.

This valuation is extended to all closed formulae of $L_I[U]$ by defining

(iv) $m \models A \supset B$ if, and only if, for all $\langle m_1, m_2 \rangle \in M^2$ such that $m_1 \models A$ and $m \cdot m_1 \leq m_2$, we have $m_2 \models B$;
(v) $m \models A \bullet B$ if, and only if, there exists $\langle m_1, m_2 \rangle \in M^2$ with $m_1 \models A$ and $m_2 \models B$ and $m_1 \cdot m_2 \leq m$;
(vi) $m \models \forall x\, A(x)$ if, and only if, for each $u \in U$, $m \models A(\underline{u})$.

We can also extend the relation $\models$ to include closed sequents of $L_I[U]$ by defining

(vii) $m \models C_1, \ldots, C_n = \vdash B$ if, and only if, $m \models (C_1 \bullet \ldots \bullet C_n) \supset B$.

A straightforward induction shows that for any closed $L_I[U]$ formula, A, we have that if $m \models A$ holds and $m \leq m'$ then $m' \models A$ holds.

The quadruple $\langle \mathbf{M}, M, \mathbf{U}, \models \rangle$ is called a *model* with domain U. An $L_I[U]$ sequent $C_1, \ldots, C_n = \vdash B$ is *valid in this model* if, and only if, $1 \models C_1^*, \ldots, C_n^* = \vdash B^*$ holds for each closed sequent $C_1^*, \ldots, C_n^* = \vdash B^*$, where $C_1^*, \ldots, C_n^* = \vdash B^*$ is any closed $L_I[U]$ sequent which can be obtained from $C_1, \ldots, C_n = \vdash B$ by uniformly replacing any free variables present by constants of $L_I[U]$. Finally, a sequent $C_1, \ldots, C_n = \vdash B$ is called *valid* if, and only if, the sequent is valid in every model. It is easily verified that the axiom sequents of the form $A = \vdash A$ and $= \vdash \mathbf{1}$ are valid. The axiom sequent $\Delta, \bot, \Sigma = \vdash A$ is also valid; this follows by observing that if $m \models D_1 \bullet \ldots \bullet D_n \bullet \bot \bullet E_1 \bullet \ldots \bullet E_m$, then $m = \infty$, and also that $\infty \models A$ holds for every formula A, which is established by induction on the complexity of A.

1.3 *Soundness and Completeness*

Theorem 1. If the sequent $\Gamma = \vdash A$ *is provable in* L_I *then* $\Gamma = \vdash A$ *is valid.*

Proof. From the remarks at the end of the previous section, it suffices to establish that all inference rules preserve validity. We illustrate the argument by considering the two rules ($\supset$-L) and ($\bullet$-L).

Let $\langle \mathbf{M}, M, \mathbf{U}, \models \rangle$ be an arbitrary model and let F be any closed formula of $L_I[U]$. We will denote by m_F an arbitrary element of M such that $m_F \models F$ holds in the model. Consider an application of the rule ($\supset$-L)

$$\frac{C_1, \ldots, C_n = \vdash A \quad D_1, \ldots, D_m, B, E_1, \ldots, E_r = \vdash C}{D_1, \ldots, D_m, A \supset B, C_1, \ldots, C_n, E_1, \ldots, E_r = \vdash C}$$

Since we are assuming that the premisses are valid we have (using the notation introduced in the last paragraph of section 1.2)

$$m_{C_1^*} \cdot \ldots \cdot m_{C_n^*} \models A^* \tag{1}$$

and

$$m_{D_1^*} \cdot \ldots \cdot m_{D_m^*} \cdot m_{B^*} \cdot m_{E_1^*} \cdot \ldots \cdot m_{E_r^*} \models C^* \tag{2}$$

Now from the definition of the relation $m \models A^* \supset B^*$ we have that $m_{A^* \supset B^*} \cdot m_{A^*} \models B^*$, so using (1), with $m_{A^*} = m_{C_1^*} \ldots m_{C_n^*}$, we have

$$m_{A^* \supset B^*} \cdot m_{C_1^*} \cdot \ldots \cdot m_{C_n^*} \models B^* \tag{3}$$

Using (3) in (2) with $m_{B^*} = m_{A^* \supset B^*} \cdot m_{C_1^*} \, . \, \dots \, . \, m_{C_n^*}$ then yields

$$m_{D_1^*} \, . \, \dots \, . \, m_{D_m^*} \cdot m_{A^* \supset B^*} \cdot m_{C_1^*} \, . \, \dots \, . \, m_{C_n^*} \cdot m_{E_1^*} \, . \, \dots \, . \, m_{E_r^*} \models C^*,$$

and this establishes that the conclusion sequent of our rule ($\supset$-L) above is valid.

Consider now an application of the rule ($\bullet$-L)

$$\frac{C_1, \dots, C_r, A, B, D_1, \dots, D_n = \vdash E}{C_1, \dots, C_r, A \bullet B, D_1, \dots, D_n = \vdash E}$$

Since the premiss is assumed to be valid we have

$$m_{C_1^*} \, . \, \dots \, . \, m_{C_r^*} \cdot m_{A^*} \cdot m_{B^*} \cdot m_{D_1^*} \, . \, \dots \, . \, m_{D_n^*} \models E^*. \tag{4}$$

From the definition of the relation $m \models A^* \bullet B^*$, we have that for any $m_{A^* \bullet B^*}$ there exist an m'_{A^*} and an m'_{B^*} such that $m'_{A^*} \cdot m'_{B^*} \leq m_{A^* \bullet B^*}$. Since (4) holds for any m_{A^*} and m_{B^*}, we can replace m_{A^*} and m_{B^*} by m'_{A^*} and m'_{B^*} respectively, and then the order preservation property of the monoid operation gives us that

$$m_{C_1^*} \, . \, \dots \, . \, m_{C_r^*} \cdot m'_{A^*} \cdot m'_{B^*} \cdot m_{D_1^*} \, . \, \dots \, . \, m_{D_n^*} \leq m_{C_1^*} \, . \, \dots \, . \, m_{C_r^*} \cdot m_{A^* \bullet B^*} \cdot m_{D_1^*} \, . \, \dots \, . \, m_{D_n^*},$$

and hence we can conclude that we have

$$m_{C_1^*} \, . \, \dots \, . \, m_{C_r^*} \cdot m_{A^* \bullet B^*} \cdot m_{D_1^*} \, . \, \dots \, . \, m_{D_n^*} \models E^*,$$

which establishes the validity of the conclusion sequent for the rule ($\bullet$-L).

The remaining rules are dealt with in a similar manner, and this establishes our soundness property.

Theorem 2. If $\Gamma = \vdash A$ *is not provable in* L_I *then* $\Gamma = \vdash A$ *is not valid.*

Proof. We construct a model with the property that if $\Gamma = \vdash A$ is not provable in L_I, then $\Gamma = \vdash A$ is not valid in our model. We first specify our pre-interpretation for the language. Without loss of generality we may assume our language contains at least one constant symbol, and at least one function symbol (of arity $\geq$ 1). We take as our pre-interpretation the infinite Herbrand universe generated by the constant symbols and function symbols of L_I. Thus $L_I[U]$ contains (new) constants naming each element of our Herbrand universe.

We now specify our partially ordered monoid structure. Let W denote the set of all closed formulae of $L_I[U]$, and for each $A \in W$ define $[A] = \{B \in W$: $A = \vdash B$ is provable in $L_I[U]\}$.

Let $M = \{[A]: A \in W\} \cup \{\varnothing\}$, where $\varnothing$ is the empty set of $L_I[U]$ formulae. Partially order M by set inclusion so that $[A] \leq [B]$ if, and only if, $[A] \subseteq [B]$. Then the greatest element (∞) of M is $[\bot]$, and there is also a least element $\varnothing$. The monoid operation on M is defined by $[A]\cdot[B] = [A \bullet B]$ for any A, $B \in W$, and $\varnothing \cdot [A] = [A] \cdot \varnothing = \varnothing$. This operation is well defined since if $[A_1] = [A_2]$ and $[B_1] = [B_2]$ then $[A_1]\cdot[B_1] = [A_2]\cdot[B_2]$. The neutral element (1) for the monoid operation is $[\mathbf{1}]$.

This completes our specification of a frame $\langle \mathbf{M}, M, \mathbf{U} \rangle$.

We now specify a model based on this frame as follows. If A is any closed $L_I[U]$ formula then define $[A] \models P$ if, and only if, $P \in [A]$, for each closed atomic $L_I[U]$ formula P. Furthermore, we specify *not* $\varnothing \models P$ for all closed atomic formulae P.

Lemma 1. In the model specified above we have that if C and A are any closed $L_I[U]$ formulae then $[A] \models C$ if, and only if, $A = \vdash C$ is provable in $L_I[U]$.

Proof. We argue by induction on the structure of C. For atomic formulae the result is immediate from the specification of the model and the $A = \vdash A$ axioms. Consider the case where C is $A \supset B$. The induction hypothesis gives us that $[D] \models A$ if, and only if, $D = \vdash A$ is provable and also $[E] \models B$ if, and only if, $E = \vdash B$ is provable. Now suppose $[F] \models A \supset B$, then we have to show $F = \vdash A \supset B$ is provable. We have $[F]\cdot[D] \models B$ holds whenever $[D] \models A$, so in particular $[F]\cdot[A] \models B$ holds. Thus $[F \bullet A] \models B$ and the induction hypothesis allows us to conclude that $F \bullet A = \vdash B$ is provable. But since $F, A = \vdash F \bullet A$ is clearly provable we now conclude that $F, A = \vdash B$ is provable and hence also $F = \vdash A \supset B$.

For the converse, suppose $F = \vdash A \supset B$ is provable. If $[D] \models A$ then the induction hypothesis implies that $D = \vdash A$ is provable. By definition of the monoid operation $[F]\cdot[D] = [F \bullet D]$, and $F \bullet D = \vdash B$ is provable using proofs of $F = \vdash A \supset B$ and $D = \vdash A$, so that now using the induction hypothesis we again conclude that $[F]\cdot[D] \models B$. This establishes that $[F] \models A \supset B$, as required.

Consider the case where C is $A \bullet B$, and suppose that $[F] \models A \bullet B$. Then we have that for some $[D]$, $[E] \in W$, $[D] \cdot [E] \leq [F]$ where $[D] \models A$ and $[E] \models B$. The induction hypothesis then implies that $D = \vdash A$ and $E = \vdash B$ are provable, hence also $D \bullet E = \vdash A \bullet B$. Since $[D] \cdot [E] = [D \bullet E] \leq [F]$ implies that $F = \vdash D \bullet E$ is provable, we have that $F = \vdash A \bullet B$ is provable. For the converse, suppose $F = \vdash A \bullet B$ is provable, then from the induction hypothesis $[A] \models A$ and $[B] \models B$ so that $[A] \cdot [B] \leq [F]$ and hence $[F] \models A \bullet B$ holds.

For the case where C is the closed $L_I[U]$ formula $\forall x\, B(x)$, suppose first that $[F] \models \forall x\, B(x)$ holds. Then we have that $[F] \models B(\underline{u})$ holds for every constant $\underline{u}$ of $L_I[U]$ and the induction hypothesis implies that $F = \vdash B(\underline{u})$ is provable for all constants $\underline{u}$ in $L_I[U]$. Now if we select a proof of $F = \vdash B(\underline{u})$, where $\underline{u}$ is a constant that does not occur in F, and replace each occurrence of $\underline{u}$ in this proof by a variable y that also does not occur in the given proof, then we have a proof of $F = \vdash B(y)$. An application of the rule ($\forall$-R) then yields a proof of $F = \vdash \forall x\, B(x)$.

For the other direction, suppose that $F = \vdash \forall x\, B(x)$ is provable. Then $F = \vdash B(\underline{u})$ is provable for each constant $\underline{u}$ of $L_I[U]$ and hence we have from the induction hypothesis that $[F] \models B(\underline{u})$ holds for all such $\underline{u}$, and thus $[F] \models \forall x\, B(x)$ holds. ■

Lemma 2. Let $D_1, \ldots, D_n = \vdash A$ be an L_I sequent that is not provable. Then this sequent is not valid in the model specified above.

Proof. We show that $D_1, \ldots, D_n = \vdash A$ is not valid in the model specified prior to the preceding lemma. To do that we must establish that the relation $[\mathbf{1}] \models D_1, \ldots, D_n = \vdash A$ does not hold. Let $\{x_1, \ldots, x_n\}$ contain all the free variables present in $D_1, \ldots, D_n = \vdash A$, and without loss of generality we can assume that no variable occurs both free and bound and also that no x_i occurs as an eigenvariable in the given proof. Suppose for a contradiction that $[\mathbf{1}] \models D_1, \ldots, D_n = \vdash A$ holds. Then we have that $[\mathbf{1}] \models D_1^*, \ldots, D_n^* = \vdash A^*$ holds for all closed sequents $D_1^*, \ldots, D_n^* = \vdash A^*$ formed from the sequent $D_1, \ldots, D_n = \vdash A$ by replacing the free variables by constants of $L_I[U]$. The preceding lemma then implies that the sequent $\mathbf{1} = \vdash (D_1^* \bullet \ldots \bullet D_n^*) \supset A^*$ is provable. Now consider the case where each free variable x_i in $D_1, \ldots, D_n = \vdash A$ is replaced by a constant $\underline{u}_i$ such that $\underline{u}_1, \ldots, \underline{u}_n$ are all distinct. Given a proof of $\mathbf{1} = \vdash (D_1^* \bullet \ldots \bullet D_n^*) \supset A^*$ we can replace each instance of $\underline{u}_i$ by x_i, for $i = 1, \ldots, n$, and we obtain a proof of $\mathbf{1} = \vdash (D_1 \bullet \ldots D_n) \supset A$. However, $\mathbf{1} = \vdash (D_1 \bullet \ldots \bullet D_n) \supset A$ is provable if, and only if, $D_1, ..., D_n = \vdash A$ is provable, and this gives our contradiction as $D_1, ..., D_n = \vdash A$ is not provable by hypothesis.

The result in Lemma 2 completes the proof of Theorem 2.

As we remarked above, L_I is in fact a fragment of a formal system of Komori [2]. Komori establishes a completeness result using monoid structures similar to ours, but the universe U of individuals is no longer fixed, so that for each element $m \in M$ we have some corresponding universe $U(m)$. More precisely, Komori considers structures $\langle \mathbf{M}, K, U \rangle$ where $\mathbf{M}$ is a partially ordered monoid structure and, further, $\langle M, \leq \rangle$ is a meet semi-lattice with $a \cdot (b \cap c) = a \cdot b \cap a \cdot c$ and $(b \cap c) \cdot a = b \cdot a \cap b \cdot c$ for all $a, b, c \in M$. The least upper bound operation is used to interpret the connective $\vee$, and the quantifier $\exists$. The set K is a subset of M (in our logic we take K to be M), and U is a function from K to the power set of some non-empty set. Komori shows that if a sequent is not provable in his system, then there is a model based on a frame $\langle \mathbf{M}, M, U \rangle$, in which the sequent is not valid. Furthermore, this model can be chosen as a finite domain model, so that $U(m)$ is a finite set for each $m \in M$.

The corresponding result in our semantics for L_I employs an infinite (fixed domain) model, so our completeness result does not follow from Komori's semantics by deleting the meet semi-lattice property used to interpret $\vee$ and $\exists$.

Ono [5] has given a semantics for predicate logics without the contraction rule using monoids with fixed domain. However, Ono allows the weakening rule and so our semantics for L_I cannot be deduced from the semantics given by Ono.

2. *Representing SLD-Derivations in L_I*

In this section we begin by considering a single SLD-resolution step. The resolution step can be considered to consist of three sequential sub-steps (see below), and we specify an injective map from the set of SLD-resolution steps into the set of provable L_I sequents. This map is then used to specify an injection from the set of all finite SLD-derivations into the set of provable L_I sequents.

We adopt the terminology of Lloyd [4]. If $\mathcal{P}$ is a definite program and $\mathbf{G}$ a definite goal $\leftarrow A_1, ..., A_m, ..., A_k$ then the goal $\mathbf{G}'$ derived from $\mathbf{G}$ using a (variant of) a program clause $A_m \leftarrow B_1, ..., B_q$ is given by $\leftarrow (A_1, ..., A_{m-1}, B_1, ..., B_q, A_{m+1}, ..., A_k)\theta$, where θ is an m.g.u. of the atom A_m selected from **G** and the head A of the specified program clause. The complete derivation step from **G** to **G′** is specified by

(i) an application of the computation rule to select an atom, A_m, from **G**,
(ii) selection of a program clause whose head can be unified with A_m, and
(iii) construction of the derived goal, **G′**.

This sequence of stages will be represented in L_I by a provable sequent of the form

$$G_0,\ COMP_1,\ H_1,\ R_1\ =\ \vdash\ G_1$$

where G_0 is a sentence representing the initial goal $\mathbf{G}_0$, $COMP_1$ is a sentence representing the application of the computation rule, H_1 is a sentence representing the instance of the input clause, and R_1 is a sentence representing the construction of the resultant goal, this goal being represented by the sentence G_1.

Let $\mathbf{G}_0$ be $\leftarrow P_1, P_2, \ldots, P_n$ and suppose the computation rule selects the atom P_j. Let the selected input clause be $Q \leftarrow Q_1, Q_2, \ldots, Q_m$, with associated unifier θ, so $P_j\theta$ is syntactically identical to $Q\theta$. The resultant goal is $\leftarrow (P_1, \ldots, P_{j-1}, Q_1, \ldots, Q_m, P_{j+1}, \ldots, P_n)\theta$. We take as G_0 the universally closed sentence

$\forall((\mathrm{P}_1\bullet P_2\bullet\ldots\bullet P_n) \supset \bot)$, and similarly G_1 is the sentence
$\forall((P_1\bullet\ldots\bullet P_{j-1}\bullet Q_1\bullet\ldots\bullet Q_m\bullet P_{j+1}\bullet\ldots\bullet P_n)\theta \supset \bot)$.

$COMP_1$ is the sentence

$$\forall(P_j\bullet(P_1\bullet\ldots\bullet P_{j-1}\bullet P_{j+1}\bullet\ldots\bullet P_n) \supset (P_1\bullet\ldots\bullet P_j\bullet\ldots\bullet P_n)).$$

We take for H_1 the sentence

$$\forall((Q_1\bullet\ldots\bullet Q_m) \supset Q),$$

and for R_1 the sentence

$$\forall((P_1\bullet\ldots,\bullet P_{j-1}\bullet Q_1\bullet\ldots,\bullet Q_m\bullet P_{j+1}\bullet\ldots\bullet P_n)\theta \supset (Q_1\bullet\ldots\bullet Q_m\bullet P_1\bullet\ldots\bullet P_{j-1}\bullet P_{j+1}\bullet\ldots\bullet P_n)\theta).$$

Note that if the connective $\bullet$ is replaced by the classical conjunction, then both the modified $COMP_1$ and R_1 are provable in classical logic. However, neither $COMP_1$ nor R_1 are theorems in L_I.

We now show the sequent

$$G_0,\ COMP_1,\ H_1,\ R_1 \ = \vdash\ G_1$$

is provable in L_I.

The two quantifier-free sequents

$$(P_1\bullet\ldots\bullet P_n)\theta \supset \bot,\ (P_j\bullet P_1\bullet\ldots\bullet P_{j-1}\bullet P_{j+1}\bullet\ldots\bullet P_n)\theta \supset (P_1\bullet\ldots\bullet P_j\bullet\ldots\bullet P_n)\theta,\ (P_j\bullet P_1\bullet\ldots\bullet P_{j-1}\bullet P_{j+1}\bullet\ldots\bullet P_n)\theta \ = \vdash \bot$$

and

$$(Q_1\bullet\ldots\bullet Q_m)\theta \supset P_j\,\theta,\ (Q_1\bullet\ldots\bullet Q_m\bullet P_1\bullet\ldots\bullet P_{j-1}\bullet P_{j+1}\bullet\ldots\bullet P_n)\theta, = \vdash\ (P_j\bullet P_1\bullet\ldots\bullet P_{j-1}\bullet P_{j+1}\bullet\ldots\bullet P_n)\theta$$

are easily seen to be provable in L_I. Application of the cut rule then yields the provable sequent

$$(P_1\bullet\ldots\bullet P_n)\theta \supset \bot,\ (P_j\bullet P_1\bullet\ldots\bullet P_{j-1}\bullet P_{j+1}\bullet\ldots\bullet P_n)\theta \supset (P_1\bullet\ldots\bullet P_j\bullet\ldots\bullet P_n)\theta,\ (Q_1\bullet\ldots\bullet Q_m)\theta \supset P_j\,\theta,\ (Q_1\bullet\ldots\bullet Q_m\bullet P_1\bullet\ldots\bullet P_{j-1}\bullet P_{j+1}\bullet\ldots\bullet P_n)\theta \ = \vdash\ \bot.$$

If we now apply the rule ($\forall$-L) to the first three formulae in the antecedent of this last sequent, we have a proof of the sequent

$$G_0,\ COMP_1,\ H_1,\ (Q_1\bullet\ldots\bullet Q_m\bullet P_1\bullet\ldots\bullet P_{j-1}\bullet P_{j+1}\bullet\ldots\bullet P_n)\theta \ = \vdash\ \bot.$$

An application of ($\supset$-L) then yields a proof of

$$G_0,\ COMP_1,\ H_1,\ (P_1\bullet\ldots\bullet P_{j-1}\bullet Q_1\bullet\ldots\bullet Q_m\bullet P_{j+1}\bullet\ldots\bullet P_n)\theta \supset (Q_1\bullet\ldots\bullet Q_m\bullet P_1\bullet\ldots\bullet P_{j-1}\bullet P_{j+1}\bullet\ldots\bullet P_n)\theta,\ (P_1\bullet\ldots\bullet P_{j-1}\bullet Q_1\bullet\ldots\bullet Q_m\bullet P_{j+1}\bullet\ldots\bullet P_n)\theta \ = \vdash\ \bot.$$

Now an application of ($\supset$-R), followed by an application of ($\forall$-L), gives

$$G_0,\ COMP_1,\ H_1,\ R_1\ =\ \vdash (P_1 \bullet \ldots \bullet P_{j-1} \bullet Q_1 \bullet \ldots \bullet Q_m \bullet P_{j+1} \bullet \ldots \bullet P_n)\theta \supset \bot .$$

Finally, applying ($\forall$-R), noting all formulae in the antecedent are now closed, we have

$$G_0,\ COMP_1,\ H_1,\ R_1,\ =\ \vdash\ G_1$$

as required.

Successful SLD-derivations in L_I

Given a successful derivation with initial goal $\mathbf{G}_0$, and final empty goal $\mathbf{G}_n$, we have in L_I the associated provable sequents, for each derivation step,

$$\begin{array}{ll} G_0,\ COMP_1,\ H_1,\ R_1 & = \vdash\ \mathrm{G}_1 \\ \mathrm{G}_1,\ COMP_2,\ H_2,\ R_2 & = \vdash\ G_2 \\ \vdots & \vdots \\ G_{n-1},\ COMP_n,\ H_n & = \vdash\ \bot \end{array} \qquad (1)$$

Repeated application of the rule ($\bullet$-R) then gives the provable sequent,

$$G_0,\ COMP_1,\ H_1,\ R_1,\ G_1,\ COMP_2, \ldots,\ G_{n-1},\ COMP_n,\ H_n\ = \\ \vdash G_1 \bullet G_2 \bullet \ldots \bullet \bot .$$

Now since we have as an initial sequent

$$G_1,\ G_2,\ \ldots,\ G_{n-1},\ \bot\ =\ \vdash\ \bot$$

we have the provable sequent

$$G_0,\ COMP_1,\ H_1,\ R_1,\ G_1,\ \ldots,\ H_n\ =\ \vdash\ \bot . \qquad (2)$$

We can regard (2) as representing the entire derivation in L_I, but an alternative to (2) is obtained by applying the cut rule (n-1) times using (1) (to eliminate $G_1, \ldots, G_{n-1}$ as antecedent formulae)

$$G_0,\ COMP_1,\ H_1,\ R_1,\ COMP_2,\ H_2,\ R_2,\ COMP_3, \ldots,\ COMP_n,\ H_n\ =\ \vdash \bot . (3)$$

If the derivation has computed answer substitution θ, then corresponding to the soundness of SLD-resolution, as given in Lloyd [4], we have the following provable sequent of L_I

$$COMP_1,\ H_1,\ R_1,\ COMP_2,\ \ldots,\ R_{n\text{-}1},\ COMP_n,\ H_n\ =\ \vdash\ \forall(P_1\bullet\ldots\bullet P_n)\theta, \quad (4)$$

where $\mathbf{G}_0$ is $\leftarrow P_1,\ P_2,\ \ldots,\ P_n$.

This result is readily established using an argument analogous to that used in establishing the provability of the sequent G_0, $COMP_1$, H_1, $R_1 = \vdash G_1$, given in the beginning of this section. We take (4) as our L_I sequent representing a refutation branch for $\mathcal{P} \cup \{\leftarrow P_1, \ldots, P_n\}$.

Finitely failed SLD-derivations in L_I

Finitely failed derivations end in the application of the computation rule to select an atom which cannot be unified with the head of any input clause. Corresponding to (1) above we have the provable sequents

$$\begin{array}{ll} G_0,\ COMP_1,\ H_1,\ R_1 & = \vdash\ G_1 \\ \vdots & \vdots \\ G_{n\text{-}2},\ COMP_{n\text{-}1},\ H_{n\text{-}1},\ R_{n\text{-}1} & = \vdash\ G_{n\text{-}1} \\ G_{n\text{-}1},\ COMP_n, & = \vdash\ G_n^* \end{array} \quad (5)$$

where, if $\mathbf{G}_{n\text{-}1}$ is $\leftarrow Q_1, \ldots, Q_r$, the final goal, and $COMP_n$ is

$$\forall((Q_j\bullet Q_1\bullet\ldots\bullet Q_{j\text{-}1}\bullet Q_{j+1}\bullet\ldots\bullet Q_r) \supset (Q_1\bullet\ldots\bullet Q_r))$$

corresponding to the selection of the atom Q_j, then G_n^* is defined to be the formula $\forall((Q_j\bullet Q_1\bullet\ldots\bullet Q_{j\text{-}1}\bullet Q_{j+1}\bullet\ldots\bullet Q_r \supset \bot)$

To obtain a sequent analogous to (2) we see that from (5) we have the provable sequent

$$G_0,\ COMP_1,\ H_1,\ R_1,\ G_1,\ \ldots\ G_{n\text{-}1},\ COMP_n\ =\ \vdash\ G_0\bullet\ldots\bullet G_n^*.$$

Since the sequent

$$G_0\bullet\ldots\bullet G_n^*\bullet Q_j\bullet Q_1\bullet\ldots\bullet Q_{j\text{-}1}\bullet Q_{j+1}\bullet\ldots\bullet Q_r\ =\ \vdash\ \bot$$

is provable we then have the provable sequent

$$G_0,\ COMP_1,\ H_1,\ R_1,\ G_1,\ \ldots,\ G_{n-1},\ COMP_n,\ Q_j \bullet Q_1 \bullet \ldots \bullet Q_{j-1} \bullet Q_{j+1} \bullet \ldots \bullet Q_r\ =\ \vdash\ \bot$$

which then yields

$$G_0,\ COMP_1,\ H_1,\ R_1,\ G_1,\ \ldots,\ G_{n-1},\ COMP_n\ =\ \vdash\ G_n^*$$

as the analogue of (2).

Similarly an application of the cut rule (n-1) times in (5) yields the provable sequent

$$G_0,\ COMP_1,\ H_1,\ R_1,\ COMP_2,\ H_2,\ \ldots,\ COMP_n\ =\ \vdash\ G_n^*$$

which is the analogue of (3).

If the failed derivation has associated unifier sequence $\theta_1, \ldots, \theta_{n-1}$ then letting θ be the composition of this unifier sequence, we also have the provable sequent

$$COMP_1,\ H_1,\ R_1,\ COMP_2,\ \ldots,\ COMP_n\ =\ \vdash\ \forall(Q_j \bullet Q_1 \bullet \ldots \bullet Q_{j-1} \bullet Q_{j+1} \bullet \ldots \bullet Q_r) \supset \forall(P_1 \bullet \ldots \bullet P_n)\theta. \tag{6}$$

corresponding to (4).

We take (6) as our L_I sequent for a finitely failed derivation of $\mathcal{P} \cup \{\leftarrow P_1, \ldots, P_n\}$, terminating with the goal $\leftarrow Q_1, \ldots, Q_r$ in which the atom Q_j is selected and fails to unify with the head of any program clause.

3. *Concluding Remarks*

We have introduced a formal system L_I for representing finite SLD-derivations so that a finite derivation may be represented by a provable sequent $A_1, \ldots, A_n = \vdash B$, where the sequence of formulae $A_1, \ldots, A_n$ represent the sequential applications of the computation rule, program clause selection, and subsequent re-arrangement of atoms in the derived goals. A sound and complete semantics for L_I is obtained in terms of Kripke-style semantics using a monoid equipped with a partial order relation.

To extend the analysis to include infinite SLD-derivations we have to extend L_I to an infinitary logic in which sequents of the form $A_1, \ldots, = \vdash B$ are allowed with an infinite sequence of formulae in the antecedent. Such an extension is indeed possible by adjoining to L_I infinitary connectives to represent an infinitary expression like $A_1 \bullet A_2 \bullet \ldots$. A soundness property for this infinitary logic is obtained by considering monoids further equipped with infinitary operations, but we do not have a corresponding completeness property.

Monash University

Bibliography

[1] Gentzen, G., *Collected Works*, M.E. Szabo (ed.), North-Holland, Amsterdam, 1969.

[2] Komori, Y., Predicate Logics Without the Structure Rules, *Studia Logica 45:* 393-404, 1985.

[3] Kripke, S., Semantical Analysis of Intuitionistic Logic I, in *Formal Systems and Recursive Functions*, North-Holland, Amsterdam, 1965.

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

[5] Ono, H., Semantical Analysis of Predicate Logics Without the Contraction Rule, *Studia Logica 44:* 187-196, 1985.