A Logical Characterization of Forward and Backward Chaining in the Inverse Method

Kaustuv Chaudhuri, Frank Pfenning, and Greg Price⋆

Department of Computer Science
Carnegie Mellon University
{kaustuv,fp}@cs.cmu.edu and gprice@andrew.cmu.edu

Abstract. The inverse method is a generalization of resolution that can be applied to non-classical logics. We have recently shown how Andreoli's focusing strategy can be adapted for the inverse method in linear logic. In this paper we introduce the notion of focusing bias for atoms and show that it gives rise to forward and backward chaining, generalizing both hyperresolution (forward) and SLD resolution (backward) on the Horn fragment. A key feature of our characterization is the structural, rather than purely operational, explanation for forward and backward chaining. A search procedure like the inverse method is thus able to perform both operations as appropriate, even simultaneously. We also present experimental results and an evaluation of the practical benefits of biased atoms for a number of examples from different problem domains.

1 Introduction

Designing and implementing an efficient theorem prover for a non-classical logic requires deep knowledge about the structure and properties of proofs in this logic. Fortunately, proof theory provides a useful guide, since it has isolated a number of important concepts that are shared between many logics of interest. The most fundamental is Gentzen's cut-elimination property [7] which allows us to consider only subformulas of a goal during proof search. Cut elimination gives rise to the inverse method [6] for theorem proving which applies to many non-classical logics. A more recent development is Andreoli's focusing property [1, 2] which allows us to translate formulas into derived rules of inference and then consider only the resulting big-step derived rules without losing completeness. Even though Andreoli's system was designed for classical linear logic, similar focusing systems for many other logics have been discovered [10, 8].

In prior work we have constructed a focusing system for *intuitionistic* linear logic which is consonant with Andreoli's classical version [5], and shown that restricting the inverse method to work only with big-step rules derived from focusing dramatically improves its efficiency [4]. The key feature of focusing is that each logical connective carries an intrinsic attribute called polarity that determines its behavior under focusing. In the case of linear logic, polarities are uniquely determined for each connective. However, as Andreoli noted, polarities may be chosen freely for atomic formulas as long as

⋆ This work has been supported by the Office of Naval Research (ONR) under grant MURI N00014-04-1-0724 and by the National Science Foundation (NSF) under grant CCR-0306313.

duality is consistently maintained. In this paper we prove that, despite the asymmetric nature of intuitionistic logic, a similar observation can be made here. Furthermore, we show that proof search on Horn formulas with the inverse method behaves either like hyperresolution or SLD resolution, depending on the chosen polarity for atoms. If different atoms are ascribed different polarities we can obtain combinations of these strategies that remain complete. The focused inverse method therefore directly generalizes these two classical proof search strategies. We also demonstrate through an implementation and experimental results that this choice can be important in practical proof search situations and that the standard polarity assumed for atoms in intuitionistic [9] or classical [14] logic programming is often the less efficient one.

Since focusing appears to be an almost universal phenomenon among non-classical logics, we believe these observations have wide applicability in constructing theorem provers. The fact that we obtain well-known standard strategies on the Horn fragment, where classical, intuitionistic, and even linear logic coincide, provides further evidence. We are particularly interested in intuitionistic linear logic and its extension by a monad, since it provides the foundation for the logical framework CLF [3] which we can use to specify stateful and concurrent systems. Theorem proving in CLF thereby provides a means for analyzing properties of such systems.

The remainder of the paper is organized as follows. In Section 2 we present the backward focusing calculus that incorporates focusing bias on atoms. In Section 2.1 we describe the derived rules that are generated with differently biased atoms. We then sketch the focused inverse method in Section 3, noting the key differences between sequents and rules in the forward direction from their analogues in the backward direction. In Section 4 we concentrate on the Horn fragment, where we show that the derived rules generalize hyperresolution (for right-biased atoms) and SLD resolution (for left-biased atoms). Finally, section 5 summarizes our experimental results on an implementation of the inverse method presented in Section 3.

2 Biased focusing

We consider intuitionistic linear logic including the following connectives: linear implication ($\multimap$), multiplicative conjunction ($\otimes$, $\mathbf{1}$), additive conjunction ($\&$, $\top$), additive disjunction ($\oplus$, $\mathbf{0}$), the exponential ($!$), and the first-order quantifiers ($\forall$, $\exists$). Quantification is over a simple term language consisting of variables and uninterpreted function symbols applied to a number of term arguments. Propositions are written using capital letters ($A, B, \ldots$), and atomic propositions with lowercase letters ($p, q, \ldots$). We use a standard dyadic sequent calculus for this logic, having the usual nice properties: identity principle, cut-admissibility, structural weakening and contraction for unrestricted hypotheses. The rules of this calculus are standard and can be found in [4]. In this section we shall describe the focused version of this calculus.

In classical linear logic the synchronous or asynchronous nature of a given connective is identical to its polarity; the negative connectives ($\&$, $\top$, $⅋$, $\bot$, $\forall$) are asynchronous, and the positive connectives ($\otimes$, $\mathbf{1}$, $\oplus$, $\mathbf{0}$, $\exists$) are synchronous. In intuitionistic logic, where the left- and right-hand side of a sequent are asymmetric and no convolutive negation exists, we derive the properties of the connectives via the rules and phases of search: an asynchronous connective is one for which decomposition is complete in the *active phase*; a synchronous connective is one for which decomposition is complete in the *focused phase*.

symbol	**connectives**
P	left-synchronous (&, $\top$, $\multimap$)
Q	right-synchronous ($\otimes$, **1**, !)
L	left-asynchronous ($\otimes$, **1**, !)
R	right-asynchronous (&, $\top$, $\multimap$)

As our backward linear sequent calculus is two-sided, we have left- and right-synchronous and asynchronous connectives. For non-atomic propositions a left-synchronous connective is right-asynchronous, and a left-asynchronous connective right-synchronous; this appears to be universal in well-behaved logics. We define the notations in the adjacent table for *non-atomic* propositions. The contexts in sequents contain linear and unrestricted zones as is usual in dyadic formulations of the sequent calculus. The *unrestricted* zone, written Γ, contains propositions that may be consumed arbitrarily often. The *passive linear* zone, written Δ, contains propositions that must be consumed exactly once. We further restrict this zone to contain only the left-synchronous propositions. We also require a third kind of zone in active rules. This zone, written Ω, contains propositions that must be consumed exactly once, but unlike the passive linear zone, can contain arbitrary propositions. We treat this *active linear* zone as an ordered context and use a centered dot ($\cdot$) instead of commas to join active zones together. As we are in the intuitionistic setting, the right hand side must contain exactly one proposition. If the right proposition C is asynchronous, then we write the right hand side as $C\,;\cdot$. If it is synchronous and not participating in any active rule, then we write it as $\cdot\,; C$. If the shape of the right hand side does not matter, we write it as γ. We have the following kinds of sequents: *right-focal* sequents $\Gamma\,;\Delta \gg A$, *left-focal* sequents $\Gamma\,;\Delta\,;A \ll Q$ (focus on A in both cases), and *active sequents* $\Gamma\,;\Delta\,;\Omega \Longrightarrow \gamma$.

Active rules work on active sequents. In each case, a rule either decomposes an asynchronous connective (e.g. $\otimes L$) or transfers a synchronous proposition into one of the passive zones. The order in which propositions are examined is immaterial.

$$\frac{\Gamma\,;\Delta\,;\Omega\cdot A\cdot B\cdot \Omega' \Longrightarrow \gamma}{\Gamma\,;\Delta\,;\Omega\cdot A\otimes B\cdot \Omega' \Longrightarrow \gamma}\ \otimes L \qquad \frac{\Gamma\,;\Delta, P\,;\Omega\cdot \Omega' \Longrightarrow \gamma}{\Gamma\,;\Delta\,;\Omega\cdot P\cdot \Omega' \Longrightarrow \gamma}\ \text{lact} \qquad \frac{\Gamma\,;\Delta\,;\Omega \Longrightarrow \cdot\,;Q}{\Gamma\,;\Delta\,;\Omega \Longrightarrow Q\,;\cdot}\ \text{ract}$$

Because the ordering of propositions in Ω is immaterial, it is then sufficent to designate a particular ordering in which these rules will are applied. We omit the standard details here. Eventually the active sequent is reduced to the form $\Gamma\,;\Delta\,;\cdot \Longrightarrow \cdot\,;Q$, which we call a neutral sequent. We will often write neutral sequents simply as $\Gamma\,;\Delta \Longrightarrow Q$.

A *focusing phase* is launched from a neutral sequent by selecting a proposition from Γ, Δ, or the right hand side:

$$\frac{\Gamma\,;\Delta\,;P \ll Q}{\Gamma\,;\Delta, P \Longrightarrow Q}\ \text{lf} \qquad \frac{\Gamma, A\,;\Delta\,;A \ll Q}{\Gamma, A\,;\Delta \Longrightarrow Q}\ \text{copy} \qquad \frac{\Gamma\,;\Delta' \gg Q \quad Q \text{ non atomic}}{\Gamma\,;\Delta \Longrightarrow Q}\ \text{rf}$$

This focused formula is decomposed under focus until the proposition becomes asynchronous. For example:

$$\frac{\Gamma\,;\Delta \gg A \quad \Gamma\,;\Delta' \gg B}{\Gamma\,;\Delta,\Delta' \gg A\otimes B}\ \otimes R \qquad \frac{\Gamma\,;\Delta\,;A \ll Q}{\Gamma\,;\Delta\,;A\,\&\,B \ll Q}\ \&L_1 \qquad \frac{\Gamma\,;\Delta\,;B \ll Q}{\Gamma\,;\Delta\,;A\,\&\,B \ll Q}\ \&L_2$$

As mentioned before, atomic propositions are somewhat special. Andreoli observed in [1] that it is sufficient to assign arbitrarily a synchronous or asynchronous nature to the atoms as long as duality is preserved; here, the asymmetric nature of the intuitionistic sequents suggests that they should be synchronous. However, we are still left with two possibilities for the initial sequents.

$$\overline{\Gamma\,;\cdot\,;p \ll p} \qquad \text{and} \qquad \overline{\Gamma\,;q \gg q}$$

In previous work [4, 5], we always selected the first of these two possibilities for the initial sequent. In this paper, we allow both kinds of initial sequents depending on the kind of *focusing bias* with regard to specific atoms. A *right-biased* atom has the Horn-like interpretation; here initial sequents have a *left* focus, and the right hand side is treated like the neutral "goal" in logic programming. A *left-biased* atom has the state-like interpretation; here initial sequents have a right focus, and the constitution of the linear context corresponds more directly to the evolution of the state.

The full set of rules is omitted; they can be reconstructed from [5, 4]. We will briefly mention below the completeness theorem which proceeds via cut-elimination for the focusing calculus. This kind of theorem is not a contribution of this paper; we provided a similar proof for the right-focused system in [5]. The basic idea is to interpret every non-focusing sequent as an active sequent in the focusing calculus, then to show that the backward rules are admissible in the focusing calculus using cut. Because propositions have dual synchronicities based on which side of the sequent arrow they appear in, a left-focal sequent matches only an active sequent in a cut; similarly for right-synchronous propositions. Cuts destroy focus as they generally require commutations spanning phase boundaries; this is not significant for our purposes as we interpret non-focusing sequents as active sequents.

Theorem 1 (cut). *If*

1. $\Gamma\,;\Delta \gg A$ *and:*
   (a) $\Gamma\,;\Delta'\,;\Omega\cdot A \Longrightarrow \gamma$ *then* $\Gamma\,;\Delta,\Delta'\,;\Omega \Longrightarrow \gamma$.
   (b) $\Gamma\,;\Delta',A\,;\Omega \Longrightarrow \gamma$ *then* $\Gamma\,;\Delta,\Delta'\,;\Omega \Longrightarrow \gamma$.
2. $\Gamma\,;\cdot \gg A$ *and* $\Gamma,A\,;\Delta\,;\Omega \Longrightarrow \gamma$ *then* $\Gamma\,;\Delta\,;\Omega \Longrightarrow \gamma$.
3. $\Gamma\,;\Delta\,;\Omega \Longrightarrow A\,;\cdot$ *or* $\Gamma\,;\Delta\,;\Omega \Longrightarrow \cdot\,;A$ *and:*
   (a) $\Gamma\,;\Delta'\,;A \ll Q$ *then* $\Gamma\,;\Delta,\Delta'\,;\Omega \Longrightarrow Q$.
   (b) $\Gamma\,;\Delta'\,;\Omega'\cdot A \Longrightarrow \gamma$ *then* $\Gamma\,;\Delta,\Delta'\,;\Omega\cdot\Omega' \Longrightarrow \gamma$.
   (c) $\Gamma\,;\Delta',A\,;\Omega' \Longrightarrow \gamma$ *then* $\Gamma\,;\Delta,\Delta'\,;\Omega\cdot\Omega' \Longrightarrow \gamma$.
4. $\Gamma\,;\cdot\,;\cdot \Longrightarrow A\,;\cdot$ *or* $\Gamma\,;\cdot\,;\cdot \Longrightarrow \cdot\,;A$ *and* $\Gamma,A\,;\Delta\,;\Omega \Longrightarrow \gamma$*, then* $\Gamma\,;\Delta\,;\Omega \Longrightarrow \gamma$.

The proof is by lexicographic induction over the structure of the two input derivations. It has one important difference from similar structural cut-admissibility proofs: when permuting a cut into an active derivation, we sometimes need to reorder the input derivation in order to allow permuting the cut to the point where it becomes a principal cut. Thus, we have to generalize the induction hypothesis to be applicable not only to structurally smaller derivations, but also permutations of the smaller derivations that differ in the order of the active rules. For lack of space, we omit the details of this proof.

Theorem 2 (completeness). *If* $\Gamma\,;\Delta \Longrightarrow C$ *in non-focused intuitionistic linear logic, then* $\Gamma\,;\cdot\,;\Delta \Longrightarrow C\,;\cdot$.

The proof uses cut to show that the non-focusing rules are admissible in the focusing system.

2.1 Derived inference rules

The primary benefit of focusing is the ability to generate derived "big step" inference rules: the intermediate results of a focusing or active phase are not important. Andreoli called these rules "bipoles" because they combine two phases with principal formulas of opposite polarities. Each derived rule starts (at the bottom) with a neutral sequent from which a synchronous proposition is selected for focus. This is followed by a sequence of focusing steps until the proposition under focus becomes asynchronous. Then, the active rules are applied, and we eventually obtain a collection of neutral sequents as the leaves of this fragment of the focused derivation. These neutral sequents are then treated as the premisses of the derived rule that produces the neutral sequent with which we started.

For lack of space, we omit a formal presentation of the derived rule calculus; instead, we will motivate it with an example. Consider the negative proposition $q \otimes n \multimap d \otimes d \otimes d$[1] in the unrestricted context Γ. We start with focus on this proposition, and construct the following derivation tree.

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma ; \Delta_1 \Longrightarrow q}{\Gamma ; \Delta_1 ; \cdot \Longrightarrow q ; \cdot}}{\Gamma ; \Delta_1 \gg q}\,\text{rb} \quad \dfrac{\dfrac{\Gamma ; \Delta_2 \Longrightarrow n}{\Gamma ; \Delta_2 ; \cdot \Longrightarrow n ; \cdot}}{\Gamma ; \Delta_2 \gg n}\,\text{rb}}{\Gamma ; \Delta_1, \Delta_2 \gg q \otimes n}\,{\otimes}R \quad \dfrac{\dfrac{\Gamma ; \Delta_3, d, d, d \Longrightarrow Q}{\Gamma ; \Delta_3 ; d \otimes d \otimes d \Longrightarrow \cdot ; Q}\,{\otimes}L; {\otimes}L; \text{lact} \times 3}{\Gamma ; \Delta_3 ; d \otimes d \otimes d \ll Q}\,\text{lb}}{\dfrac{\Gamma ; \Delta_1, \Delta_2, \Delta_3 ; q \otimes n \multimap d \otimes d \otimes d \ll Q}{\Gamma ; \Delta_1, \Delta_2, \Delta_3 \Longrightarrow Q}\,\text{copy}}\,{\multimap}L$$

Here we assume that all atoms are right-biased, so none of the branches of the derivation can be closed off with an "init" rule. Thus, we obtain the derived rule:

$$\dfrac{\Gamma ; \Delta_1 \Longrightarrow q \quad \Gamma ; \Delta_2 \Longrightarrow n \quad \Gamma ; \Delta_3, d, d, d \Longrightarrow Q}{\Gamma ; \Delta_1, \Delta_2, \Delta_3 \Longrightarrow Q} \qquad (D_1)$$

The situation is considerably different if we assume that all atoms are left-biased. In this case, we get the following derivation:

$$\dfrac{\dfrac{\dfrac{}{\Gamma ; q \gg q}\,\text{linit} \quad \dfrac{}{\Gamma ; n \gg n}\,\text{linit}}{\Gamma ; q, n \gg q \otimes n}\,{\otimes}R \quad \dfrac{\dfrac{\Gamma ; \Delta, d, d, d \Longrightarrow \cdot ; Q}{\Gamma ; \Delta ; d \otimes d \otimes d \Longrightarrow \cdot ; Q}\,{\otimes}L; {\otimes}L; \text{lact} \times 3}{\Gamma ; \Delta ; d \otimes d \otimes d \ll Q}\,\text{lb}}{\dfrac{\Gamma ; q, n, \Delta ; q \otimes n \multimap d \otimes d \otimes d \ll Q}{\Gamma ; q, n, \Delta \Longrightarrow Q}\,\text{copy}}\,{\multimap}L$$

In this left-biased case, we can terminate the left branch of the derivation with a pair of "init" rules. This rule forces the linear context in this branch of the proof to contain just the atoms q and n. The derived rule we obtain is, therefore,

$$\dfrac{\Gamma ; \Delta, d, d, d \Longrightarrow Q}{\Gamma ; \Delta, q, n \Longrightarrow Q} \qquad (D_2)$$

[1] Standing roughly for "quarter and nickel goes to three dimes".

There are two key differences to observe between the derived rules (D_1) and (D_2). The first is that simply altering the bias of the atoms has a huge impact on the kinds of rules that are generated; even if we completely ignore the semantic aspect, the rule (D_2) is vastly preferable to (D_1) because it is much easier to use single premiss rules.

The second — and more important — observation is that the rule that was generated for the left-biased atoms has a stronger and more obvious similarity to the proposition $q \otimes n \multimap d \otimes d \otimes d$ that was under focus. If we view the linear zone as the "state" of a system, then the rule (D_2) corresponds to transforming a portion of the state by replacing q and n by three ds (reading the rule from bottom to top). If, as is common for linear logic, the unrestricted context Γ contains state transition rules for some encoding of a stateful system, then the derived rules generated by left-biasing allows us to directly observe the evolution of the state of the system by looking at the composition of the linear zone.

3 The focused inverse method

In this section we will briefly sketch the inverse method using the focusing calculus of the previous section. The construction of the inverse method for linear logic is described in more detail in [4]. To distinguish forward from backward sequents, we shall use a single arrow ($\longrightarrow$), but keep the names of the rules the same. In the forward direction, the primary context management issue concerns rules where the conclusion cannot be simply assembled from the premisses. The backward $\top R$ rule has an arbitrary linear context Δ, and the unrestricted context Γ is also unknown in several rules such as init and $\top R$. For the unrestricted zone, this problem is solved in the usual (non-linear) inverse method by collecting only the needed unrestricted assumptions and remembering that they can be weakened if needed [6]. We adapt the solution to the linear zone, which may either be precisely determined (as in the case for initial sequents) or subject to weakening (as in the case for $\top R$). We therefore differentiate sequents whose linear context can be weakened and those whose can not.

Definition 3 (forward sequents). *A* forward sequent *is of the form $\Gamma\,;[\Delta]_w \longrightarrow \gamma$, with w a Boolean (0 or 1) called the* weak-flag, *and γ being either empty ($\cdot$) or a singleton. The sequent $\Gamma\,;[\Delta]_w \longrightarrow \gamma$ corresponds to the backward sequent $\Gamma'\,;\Delta' \Longrightarrow C$ if $\Gamma \subseteq \Gamma'$, $\gamma \subseteq C$; and $\Delta = \Delta'$ if $w = 0$ and $\Delta \subseteq \Delta'$ if $w = 1$. Sequents with $w = 1$ are called* weakly linear *or simply* weak, *and those with $w = 0$ are* strongly linear *or* strong.

Initial sequents are always strong, since their linear context cannot be weakened. On the other hand, $\top R$ always produces a weak sequent. For binary rules, the unrestricted zones are simply juxtaposed. We can achieve the effect of taking their union by applying the explicit contraction rule (which is absent, but admissible in the backward calculus). For the linear zone we have to distinguish cases based on whether the sequent is weak or not. We write the rules using two operators on the linear context – multiplicative composition ($\times$) and additive composition ($+$).

$$\frac{\Gamma\,;[\Delta]_w \longrightarrow A \quad \Gamma'\,;[\Delta']_{w'} \longrightarrow B}{\Gamma,\Gamma'\,;[\Delta]_w \times [\Delta']_{w'} \longrightarrow A \otimes B}\ \otimes R \qquad \frac{\Gamma\,;[\Delta]_w \longrightarrow^{w} A \quad \Gamma'\,;[\Delta']_{w'} \longrightarrow B}{\Gamma,\Gamma'\,;[\Delta]_w + [\Delta']_{w'} \longrightarrow A \,\&\, B}\ \&R$$

These compositions are defined as follows: For multiplicative rules, it is enough for one premiss to be weak for the conclusion to be weak; the weak flags are therefore joined with a disjunction ($\vee$). Dually, for additive rules, both premisses must be weak for the conclusion to be weak; in this case the weak flags are joined with a conjunction ($\wedge$).

Definition 4 (context composition). *The partial operators $\times$ and $+$ on forward linear contexts are defined as follows: $[\Delta]_w \times [\Delta']_{w'} =_{def} [\Delta, \Delta']_{w \vee w'}$, and*

$$[\Delta]_w + [\Delta']_{w'} =_{def} \begin{cases} [\Delta]_0 & \textit{if } w = 0 \textit{ and either } w' = 0 \textit{ and } \Delta = \Delta', \textit{ or } w' = 1 \textit{ and } \Delta' \subseteq \Delta \\ [\Delta']_0 & \textit{if } w' = 0,\, w = 1 \textit{ and } \Delta \subseteq \Delta' \\ [\Delta \sqcup \Delta']_1 & \textit{if } w = w' = 1 \end{cases}$$

Here $\Delta \sqcup \Delta'$ is the multiset union of Δ and Δ'.

In the lifted version of this calculus with free variables, there is no longer a single context represented by $\Delta \sqcup \Delta'$ because two propositions might be equalized by substitution. The approach taken in [4] was to define an additional "context simplification" procedure that iteratively calculates a set of candidates that includes every possible context represented by $\Delta \sqcup \Delta'$ by means of contraction. Many of these candidates are then immediately rejected by subsumption arguments. We refer to [4] for the full set of rules, the completeness theorem, and the lifted version of this forward calculus.

3.1 Focused forward search

The sketched calculus in the previous section mentioned only single-step rules. We are interested in doing forward search with derived inference rules generated by means of focusing. We therefore have to slightly generalize the context composition operators into a language of context expressions. In the simplest case, we merely have to add a given proposition to the linear context, irrespective of the weak flag. This happens, for instance, in the "lf" rule where the focused proposition is transferred to the linear context. We write this adjunction as usual using a comma. In the more general case, however, we have to combine two context expressions additively or multiplicatively depending on the kind of rule they were involved in; for these uses, we appropriate the same syntax we used for the single step compositions in the previous section.

$$\text{(context expressions)} \qquad \mathcal{D} \quad ::= \quad [\Delta]_w \mid \mathcal{D}, A \mid \mathcal{D}_1 + \mathcal{D}_2 \mid \mathcal{D}_1 \times \mathcal{D}_2$$

Context expressions can be *simplified* into forward contexts in a bottom-up procedure. We write $\mathcal{D} \hookrightarrow [\Delta]_w$ to denote that $\mathcal{D}$ simplifies into $[\Delta]_w$; it has the following rules.

$$\frac{}{[\Delta]_w \hookrightarrow [\Delta]_w} \qquad \frac{\mathcal{D} \hookrightarrow [\Delta]_w}{\mathcal{D}, A \hookrightarrow [\Delta, A]_w} \qquad \frac{\mathcal{D}_1 \hookrightarrow [\Delta_1]_{w_1} \quad \mathcal{D}_2 \hookrightarrow [\Delta_2]_{w_2}}{\mathcal{D}_1 + \mathcal{D}_2 \hookrightarrow [\Delta_1]_{w_1} + [\Delta_2]_{w_2}} \qquad \frac{\mathcal{D}_1 \hookrightarrow [\Delta_1]_{w_1} \quad \mathcal{D}_2 \hookrightarrow [\Delta_2]_{w_2}}{\mathcal{D}_1 \times \mathcal{D}_2 \hookrightarrow [\Delta_1]_{w_1} \times [\Delta_2]_{w_2}}$$

The forward version of backward derived rules can be written with these context expressions in a natural way by allowing unsimplified context expressions in the place of linear contexts in forward sequents. As an example, the negative unrestricted proposition $q \otimes n \multimap d \otimes d \otimes d$ has the following derived rule with right-biased atoms

$$\frac{\Gamma_1\,;[\Delta_1]_{w_1} \longrightarrow q \quad \Gamma_2\,;[\Delta_2]_{w_2} \longrightarrow n \quad \Gamma_3\,;[\Delta_3]_{w_3}, d, d, d \longrightarrow Q}{\Gamma_1, \Gamma_2, \Gamma_3\,;[\Delta_1]_{w_1} \times [\Delta_2]_{w_2} \times [\Delta_3]_{w_3} \longrightarrow Q}$$

After constructing the neutral sequent with a context expression we then simplify it. Note that context simplification is a partial operation, so we may not obtain any conclusions, for example, if the premisses to an additive rule are strong sequents but the linear contexts do not match.

3.2 Focusing in the inverse method

The details of the focused inverse method have been sketched in detail in [5]; here we briefly summarize the major differences that arise as a result of focusing bias. The key calculation as laid out in [5] is of the *frontier literals* of the goal sequent, i.e., those subformulas that are available in neutral sequents to be focused on. For all but the atoms the calculation is the same as before, and for the atoms we make the following modifications.

1. A positive atom is in the frontier if it lies in the boundary of a phase transition from active to focus, and it is left-biased.
2. A negative atom is in the frontier if it lies in the boundary of a phase transition from active to focus, and it is right-biased.

We then specialize the inference rules to these frontier literals by computing the derived rules that correspond to giving focus to these literals.

Although the addition of bias gives us different rules for focusing, the use of the rules in the search engine is no different from before. The details of the implementation of the main loop can be found in [4]. The main innovation in our formulation of the inverse method in comparison with other descriptions in the literature is the use of a lazy variant of the OTTER loop that both simplifies the design of the rules and performs well in practice.

3.3 Globalization

The final unrestricted zone Γ_g is shared in all branches in a proof of $\Gamma_g \,;\, \Delta_g \Longrightarrow \gamma_g$. One thus thinks of Γ_g as part of the ambient state of the prover, instead of representing it explicitly as part of the current goal. Hence, there is never any need to explicitly record Γ_g or portions of it in the sequents themselves. This gives us the following global and local versions of the copy rule in the forward direction.

$$\frac{\Gamma \,;\, [\Delta]_w \,;\, A \ll \gamma \quad A \in \Gamma_g}{\Gamma \,;\, [\Delta]_w \longrightarrow \gamma}\ \text{delete} \qquad \frac{\Gamma \,;\, [\Delta]_w \,;\, A \ll \gamma \quad A \notin \Gamma_g}{\Gamma, A \,;\, [\Delta]_w \longrightarrow \gamma}\ \text{copy}$$

Globalization thus corresponds to a choice of whether to add the constructed principal formula of a derived rule into the unrestricted zone or not, depending on whether or not it is part of the unrestricted zone in the goal sequent.

4 The Horn fragment

In complex specifications that employ linearity, there are often significant sub-specifications that lie in the Horn fragment. Unfortunately, the straightforward inverse method is quite inefficient on Horn formulas, something already noticed by Tammet [16]. His prover switches between hyperresolution for Horn and near-Horn formulas and the inverse method for other propositions.

With focusing, this *ad hoc* strategy selection becomes entirely unnecessary. The focused inverse method for intuitionistic linear logic, when applied to a classical, non-linear Horn formula, will exactly behave as classical hyperresolution or SLD resolution depending on the focusing bias of the atomic propositions. This remarkable property gives further credence to the power of focusing as a technique for forward reasoning. In the next two sections we will describe this correspondence in slightly more detail.

A Horn clause has the form $\neg p_1, \ldots, \neg p_n, p$ where the p_i and p are atomic predicates over their free variables. This can easily be generalized to include conjunction and truth, but we restrict our attention to this simple clausal form, as theories with conjunction and truth can be simplified into this form. A Horn theory Ψ is just a set of Horn clauses, and a Horn query is of the form $\Psi \vdash g$ where g is a ground atomic "goal" formula[2]. In the following section we use a simple translation $(-)^o$ of these Horn clauses into linear logic where $\neg p_1, \ldots, \neg p_n, p$ containing the free variables $\vec{x}$ is translated into $\forall \vec{x}.\ p_1 \multimap \cdots \multimap p_n \multimap p$, and the query $\Psi \vdash g$ is translated as $(\Psi)^o\,;\,[\cdot]_0 \longrightarrow g$. This is a special case of a general, focusing-preserving translation from intuitionistic to intuitionistic linear logic [5].

4.1 Hyperresolution

The hyperresolution strategy for the Horn query $\Psi \vdash g$ is just forward reasoning with the following rule (for $n > 1$):

$$\frac{p_1' \quad \cdots \quad p_n'}{\theta p} \quad \begin{cases} \text{where } \neg p_1, \ldots, \neg p_n, p \in \Psi;\ \rho_1, \ldots, \rho_n \text{ are renaming substs; and} \\ \theta = \mathrm{mgu}(\langle \rho_1 p_1', \ldots, \rho_n p_n' \rangle, \langle p_1, \ldots, p_n \rangle) \end{cases}$$

The hyperresolution procedure begins with the collection of unit clauses in Ψ and $\neg g$ as the initial set of facts. The proof succeeds if the empty fact (contradiction) is generated. Because every clause in the theory has a positive literal, the only way an empty fact can be generated is if it proves the fact g itself (note that g is ground).

Consider the goal sequent in the translation $(\Psi)^o\,;\,[\cdot]_0 \longrightarrow g$ where the atoms are all right-biased. The frontier is every clause $\forall \vec{x}.\ p_1 \multimap \cdots \multimap p_n \multimap p \in (\Psi)^o$. Focusing on one such clause gives the following abstract derivation in the forward direction (using lifted sequents):

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma_1\,;\,[\Delta_1]_{w_1} \longrightarrow p_1}{\Gamma_1\,;\,[\Delta_1]_{w_1}\,;\,\cdot \longrightarrow p_1\,;\,\cdot}}{\Gamma_1\,;\,[\Delta_1]_{w_1} \gg p_1} \quad \cdots \quad \dfrac{\dfrac{\Gamma_n\,;\,[\Delta_n]_{w_n} \longrightarrow p_n}{\Gamma_n\,;\,[\Delta_n]_{w_n}\,;\,\cdot \longrightarrow p_n\,;\,\cdot}}{\Gamma_n\,;\,[\Delta_n]_{w_n} \gg p_n} \quad \dfrac{}{\Gamma\,;\,[\cdot]_0\,;\,p \ll p}\ \text{rinit}}{\Gamma_1, \ldots, \Gamma_n\,;\,\Delta\,;\,p_1 \multimap \cdots \multimap p_n \multimap p \ll p}\ {\multimap}L}{\Gamma_1, \ldots, \Gamma_n\,;\,[\Delta_1]_{w_1}, \ldots, [\Delta_n]_{w_n}\,;\,\forall \vec{x}.\ p_1 \multimap \cdots \multimap p_n \multimap p \ll p}\ \forall L}{\Gamma_1, \ldots, \Gamma_n\,;\,[\Delta_1]_{w_1} \times \cdots \times [\Delta_n]_{w_n} \longrightarrow p}\ \text{delete}}$$

In other words, the derived rule is

$$\frac{\Gamma_1\,;\,\Delta_1 \longrightarrow p_1 \quad \cdots \quad \Gamma_n\,;\,[\Delta_n]_{w_n} \longrightarrow p_n}{\Gamma_1, \ldots, \Gamma_n\,;\,[\Delta_1]_{w_1} \times \cdots \times [\Delta_n]_{w_n} \longrightarrow p}$$

[2] Queries with more general goals can be compiled to this form by adding an extra clause to the theory from the desired goal to a fresh ground goal literal.

In the case where $n = 0$, i.e., the clause in the Horn theory was a unit clause p, we obtain an initial sequent of the form $\cdot\,;[\cdot]_0 \longrightarrow p$. As this clause has an empty left hand side, and none of the derived rules add elements to the left, we can make an immediate observation (lem.5) that gives us an isomorphism of rules (thm.6).

Lemma 5. *Every sequent generated in the proof of the goal* $(\Psi)^o\,;[\cdot]_0 \longrightarrow g$ *has an empty left hand side.* □

Theorem 6 (isomorphism of rules). *Every hyperresolution rule for the query* $\Psi \vdash g$ *is isomorphic to an instance of a derived rule for the overall goal sequent* $(\Psi)^0\,;[\cdot]_0 \longrightarrow g$ *with empty left-hand sides.* □

These facts let us establish an isomorphism between hyperresolution and right-biased focused derivations.

Theorem 7. *Every hyperresolution derivation for the Horn query* $\Psi \vdash g$ *has an isomorphic focused derivation for the goal sequent* $(\Psi)^o\,;[\cdot]_0 \longrightarrow g$ *with right-biased atoms.*

Proof (Sketch). For every fact p' generated by the hyperresolution strategy, we have a corresponding fact $\cdot\,;[\cdot]_0 \longrightarrow p'$ in the focused derivation (up to a renaming of the free variables). When matching these sequents for consideration as input for a derived rule corresponding to the Horn clause $\neg p_1, \ldots, \neg p_n, p$, we calculate the simultaneous mgu of all the p_i and p'_i for a Horn clause, which is precisely the operation also performed in the hyperresolution rule. The required isomorphism then follows from thm. 6. □

4.2 SLD Resolution

SLD Resolution [11] is a variant of linear resolution that is complete for Horn theories. For the Horn query $\Psi \vdash g$, we start with just the initial clause g, and then perform forward search using the following rule (using Ξ to stand for a clauses).

$$\frac{\Xi, q}{\theta(\Xi, p_1, p_2, \ldots, p_n)} \quad \begin{cases} \text{where } \neg p_1, \ldots, \neg p_n, p \in \Psi; \rho \text{ is a renaming subst; and} \\ \theta = \text{mgu}(\rho p, q) \end{cases}$$

The composition of a clause is thus a contraction-free collection of atoms. When $n = 0$, i.e., for unit clauses in the Horn theory, this rule corresponds to simply deleting the member of the input clause that was unifiable with the unit clause. The search procedure succeeds when it is able to derive the empty clause.

To show how SLD resolution is modeled by our focusing system, we reuse the translation from before, but this time all atoms are given a left bias. The derivation that corresponds to focusing on the translation of the Horn clause $\neg p_1, \ldots, \neg p_n, p$ is:

$$\dfrac{\dfrac{\dfrac{}{\cdot\,; p_1 \gg p_1}\text{ linit} \quad \cdots \quad \dfrac{}{\cdot\,; p_n \gg p_n}\text{ linit} \quad \dfrac{\dfrac{\Gamma\,;[\Delta]_w, p \longrightarrow Q}{\Gamma\,;[\Delta]_w\,; p \longrightarrow \cdot\,; Q}}{\Gamma\,;[\Delta]_w\,; p \ll \cdot\,; Q}}{\Gamma\,;[\Delta]_w, p_1, \ldots, p_n\,; p_1 \multimap \cdots p_n \multimap p \ll \cdot\,; Q}\multimap L}{\Gamma\,;[\Delta]_w, p_1, \ldots, p_n \Longrightarrow Q}\text{ delete}$$

In other words, the derived rule is:

$$\frac{\Gamma\,;[\Delta, p]_w \longrightarrow Q}{\Gamma\,;[\Delta, p_1, \ldots, p_n]_w \longrightarrow Q}$$

The frontier of the goal $(\Psi)^0\,;[\cdot]_0 \longrightarrow g$ in the left-biased setting contains every member of $(\Psi)^0$, so we obtain one such derived rule for each clause in the Horn theory. The frontier contains, in addition, the positive atom g; assuming there is a negative instance of g somewhere in the theory, we will thus generate a single initial sequent, $\cdot\,;[g]_0 \longrightarrow g$. We immediately observe that:

Lemma 8. *Every sequent generated in the focused derivation of $(\Psi)^0\,;[\cdot]_0 \longrightarrow g$ is of the form $\cdot\,;[\Delta]_0 \longrightarrow g$.* □

Theorem 9 (isomorphism of rules). *Every SLD resolution rule for the Horn query $\Psi \vdash g$ is isomorphic to an instance of a derived inference rule for the overall goal sequent $(\Psi)^0\,;[\cdot]_0 \longrightarrow g$ with empty unrestricted zones and g on the right.* □

As should be clear, the interpretation of a clause Ξ is the linear zone of the forward sequent, which also does not admit contraction.

Theorem 10. *Every SLD resolution derivation for the Horn query $\Psi \vdash g$ has an isomorphic focused derivation for the goal sequent $(\Psi)^o\,;[\cdot]_0 \longrightarrow g$ with left-biased atoms.*

Proof (Sketch). Very similar argument as thm. 7, except we note that in the matching conditions in the derived rules we rename the input sequents, whereas in the SLD resolution case we rename the Horn clause itself. However, this renaming is merely an artifact of the procedure and does not itself alter the derivation. □

Although the derivations are isomorphic, the focused derivations may not be as efficient as the SLD resolution in practice because of the need to rename (i.e., copy) the premisses as part of the matching conditions of a derived rule– premisses might contain many more components than the Horn clause itself.

5 Experiments

5.1 Propositional linear logic

The first class of experiments we performed were on propositional linear logic. We implemented several minor variants of an inverse method prover for propositional linear logic. The propositional fragment is the only fragment where we can compare with external provers, as we are not aware of any first order linear logic provers besides our own. The external prover we compared against is Tammet's Gandalf "nonclassical" distribution (version 0.2), compiled using a packaged version of the Hobbit Scheme compiler. This classical linear logic prover comes in two flavors: resolution (**Gr**) and tableau (**Gt**). Neither version incorporates focusing or globalization, and we did not attempt to bound the search for either prover. Other provers such as LinTAP [13] and llprover [17] fail to prove all but the simplest problems, so we did not do any serious comparisons against them. Our experiments were all run on a 3.4GHz Pentium 4 machine with 1MB L1 cache and 1GB main memory; our provers were compiled using

MLTon version 20060213 using the default optimization flags; all times indicated are wall-clock times in seconds and includes the GC time; × denotes unprovability within a time limit of 1 hour. In the following tables, iters refers to number of iterations of the lazy OTTER loop, gen the number of generated sequents, and subs the number of subsumed sequents.

Stateful system encodings In these examples, we encoded the state transition rules for stateful systems such as a change machine, a Blocks World problem with a fixed number of blocks, a few sample Petri nets. For the Blocks World example, we also compared a version that uses the CLF monad [3] and one without.

	right-biased				**left-biased**				**Gt**	**Gr**
name	iters	gen	subs	time	iters	gen	subs	time	time	time
`blocks`	20	43	18	0.001	12	84	61	0.001	×	×
`blocks-clf`	27	65	26	0.002	5	24	7	<**0.001**	N/A	N/A
`change`	16	22	7	0.001	11	20	6	0.001	0.63	0.31
`petri-1`	23	38	23	**0.001**	284	1099	921	0.062	×	7.08
`petri-2`	57	133	105	**0.003**	393	1654	1433	0.068	×	7.13

Graph exploration algorithms In these examples we encode the algorithm for exploring graph for calculating Euler or Hamiltonian tours. The problems have an equal balance of proofs (i.e., a tour exists) and refutations (i.e., no tour exists).

	right-biased				**left-biased**			
name	iters	gen	subs	time	iters	gen	subs	time
`euler-1`	6291	11853	5565	9.010	6291	11853	5565	**8.570**
`euler-2`	15640	34329	18689	152.12	15640	34329	18689	**145.9**
`euler-3`	64360	159194	94834	3043.35	64360	159194	94834	**2938.55**
`hamilton`	708	911	185	0.11	165	178	0	<**0.001**

The Euler tour computation uses a symmetric algorithm, so both backward and forward chaining generate the same facts, though, interestingly, a left-biased search performs slightly better than the right-biased system. For the Hamiltonian tour examples, the left-biased search is vastly superior.

Affine logic encoding Linearity is often too stringent a requirement for situations where we simply need *affine* logic, i.e., where every hypothesis is consumed *at most* once. Affine logic can be embedded into linear logic by translating every affine arrow $A \to B$ as either $A \multimap B \otimes \top$ or $A \,\&\, \mathbf{1} \multimap B$. Of course, one might select complex encodings; for example choosing $A \,\&\, !(\mathbf{0} \multimap X) \multimap B$ (for some arbitrary fresh proposition X) instead of $A \,\&\, \mathbf{1} \multimap B$. Even though the two translations are equivalent, the prover performs poorly on the former. The Gandalf provers **Gt** and **Gr** fail on these examples.

	right-biased				**left-biased**			
encoding	iters	gen	subs	time	iters	gen	subs	time
$A \multimap B \otimes \top$	38	108	73	0.003	34	107	73	**0.002**
$A \,\&\, \mathbf{1} \multimap B$	252	1103	828	0.098	62	229	126	**0.019**
$A \,\&\, !(\mathbf{0} \multimap X) \multimap B$	264	7099	6793	2.028	235	841	578	**0.042**

Quantified Boolean formulas In these examples we used two variants of the algorithm from [12] for encoding QBFs in linear logic. The first variant uses exponentials to encode reusable "copy" rules; this variant performs very well in practice, so the table below collates the results of all the example QBFs in one entry. The second variant maps to the multiplicative-additive fragment of linear logic without exponentials. This variant produces problems that are considerably harder, so we have divided the problems in three sets in increasing order of complexity.

	right-biased				**left-biased**			
encodings	iters	gen	subs	time	iters	gen	subs	time
`qbf-exp`	1508	1722	140	**0.13**	7948	17610	9590	2.69
`qbf-nonexp-1`	1457	5590	4067	**0.54**	1581	4352	2612	0.58
`qbf-nonexp-2`	15267	517551	502174	368.92	9469	49777	37716	**29.55**
`qbf-nonexp-3`	28556	990196	961494	2807.64	21233	89542	115917	**308.24**

For these examples, when the number of iterations is low (i.e., the problems are simple), the right-biased search appears to perform better than the left-biased system. However, as the problems get harder, the left-biased system becomes dominant.

5.2 First-order linear logic

We have also implemented a first-order prover for linear logic. Experiments with an early version of the first-order were documented in [4]. Since then we have made several improvements to the prover, including a complete reimplmentation of the focused rule generation engine, and also increased our collection of sample problems.

First-order stateful systems The first experiments were with first-order encodings of various stateful systems. We selected a first-order Blocks World encoding (both with and without the CLF monad), Dijkstra's Urn Game, and an AI planning problem for a certain board game. The left-biased system performs consistently better than the right-biased system for these problems.

	right-biased				**left-biased**			
problem	iters	gen	subs	time	iters	gen	subs	time
`blocks`	45	424	317	0.12	26	387	337	**0.04**
`blocks-clf`	64	697	412	0.264	15	81	69	**0.006**
`urn`	29	72	27	0.24	13	58	55	**0.11**
`board`	349	7021	3138	3.26	166	5296	1752	**0.88**

Purely intuitionistic problems Unfortunately, we are unable to compare our implementation with any other linear provers; to the best of our knowledge, our prover is the only first-order linear prover in existence. We therefore ran our prover on some problems drawn from the SICS benchmark [15]. These intuitionistic problems were translated into linear logic in two different ways– the first uses Girard's original encoding of classical logic in classical linear logic where every subformula is affixed with the exponential, and the second is a focus-preserving encoding as described in [5]. We also compared our prover with *Sandstorm*, a focusing inverse method theorem prover for intuitionistic logic implemented by students at CMU. The focus-preserving translation is always better than the Girard-translation; however, the complexity of linear logic, particularly the significant complexity of linear contraction, makes it uncompetitive with the intuitionistic prover.

	right-biased				**left-biased**				**SS**
problem	iters	gen	subs	time	iters	gen	subs	time	time
`SICS1-gir`	360	1948	1394	1.312	368	2897	2181	0.6	0.04
`SICS1-foc`	56	365	313	0.056	64	496	415	0.04	
`SICS2-gir`	3035	16391	11732	11.04	3460	27192	20389	5.856	0.06
`SICS2-foc`	489	3133	2688	0.472	616	4672	3902	0.376	
`SICS3-gir`	20958	1131823	810085	762.312	12924	1015552	761517	218.712	1.12
`SICS3-foc`	3377	21659	18646	33.096	2300	17464	14969	23.296	
`SICS4-gir`	×	×	×	×	×	×	×	×	3.89
`SICS4-foc`	8896	57056	49047	87.184	6144	46818	39993	62.24	

Horn examples from TPTP For our last example, we selected 20 non-trivial Horn problems from the TPTP version 3.1.1. The selection of problems was not systematic, but we did not constrain our selection to any particular section of the TPTP. We used the translation described in sec. 4. For lack of space we omit the list of selected problems, which can be found from the first author's web-page.[3]

right-biased				**left-biased**			
iters	gen	subs	time	iters	gen	subs	time
4911	314640	287004	**462.859**	6289	704482	526207	638.818

For Horn problems, the right-biased system, which models hyperresolution, performs better than the left-biased system, which models SLD resolution. This observation is not unprecedented— the Gandalf system switches to a Hyperresolution strategy for Horn theories [16]. The likely reason is that in the left-biased system, unlike in SLD resolution system, the derived rule renames the input sequent rather than the rule itself.

6 Conclusion

We have presented an improvement of the focusing inverse method that exploits the flexibility in assigning polarity to atoms which we call bias. This strictly generalizes both hyperresolution and SLD resolution on (classical) Horn clauses to all of intuitionistic linear logic. This strategy shows significant improvement on a number of example problems. Among the future work will be to explore strategies for determining appropriate bias for atoms from the problem statement to optimize overall search behavior.

References

1. Jean-Marc Andreoli. Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation*, 2(3):297–347, 1992.

[3] `http://www.cs.cmu.edu/~kaustuv/papers/ijcar06`

2. Jean-Marc Andreoli. Focussing and proof construction. *Annals of Pure and Applied Logic*, 107:131–163, 2001.
3. Iliano Cervesato, Frank Pfenning, David Walker, and Kevin Watkins. A concurrent logical framework I & II. Technical Report CMU-CS-02-101 and 102, Department of Computer Science, Carnegie Mellon University, 2002. Revised May 2003.
4. Kaustuv Chaudhuri and Frank Pfenning. A focusing inverse method theorem prover for first-order linear logic. In *Proceedings of CADE-20*, pages 69–83, Tallinn, Estonia, July 2005. Springer-Verlag LNAI-3632.
5. Kaustuv Chaudhuri and Frank Pfenning. Focusing the inverse method for linear logic. In Luke Ong, editor, *Proceedings of CSL 2005*, pages 200–215, Oxford, UK, August 2005. Springer-Verlag LNCS-3634.
6. Anatoli Degtyarev and Andrei Voronkov. The inverse method. In James Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, pages 179–272. MIT Press, September 2001.
7. Gerhard Gentzen. Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1935. English translation in M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131, North-Holland, 1969.
8. Jean-Yves Girard. Locus solum: from the rules of logic to the logic of rules. *Mathematical Structures in Computer Science*, 11:301–506, 2001.
9. Joshua S. Hodas and Dale Miller. Logic programming in a fragment of intuitionistic linear logic. *Information and Computation*, 110(2):327–365, 1994.
10. Jacob M. Howe. *Proof Search Issues in Some Non-Classical Logics*. PhD thesis, University of St. Andrews, September 1998.
11. R. Kowalski and D. Kuehner. Linear resolution with selection function. *Artificial Intelligence*, 2:227–260, 1971.
12. Patrick Lincoln, J. C. Mitchell, Andre Scedrov, and Natarajan Shankar. Decision problems for propositional linear logic. *Annals of Pure and Applied Logic*, 56:239–311, 1992.
13. Heiko Mantel and Jens Otten. LinTAP: A tableau prover for linear logic. In A. Murray, editor, *International Conference TABLEAUX'99*, pages 217–231, New York, June 1999. Springer-Verlag LNAI 1617.
14. D. Miller. A multiple-conclusion meta-logic. In S. Abramsky, editor, *Ninth Annual Symposium on Logic in Computer Science*, pages 272–281, Paris, France, July 1994. IEEE Computer Society Press.
15. Dan Sahlin, Torkel Franzén, and Seif Haridi. An intuitionistic predicate logic theorem prover. *Journal of Logic and Computation*, 2(5):619–656, 1992.
16. Tanel Tammet. Resolution, inverse method and the sequent calculus. In *Proceedings of KGC'97*, pages 65–83. Springer-Verlag LNCS 1289, 1997.
17. Naoyuki Tamura. Llprover. At: `http://bach.istc.kobe-u.ac.jp/llprover`.