Proof Search in the Intuitionistic Sequent Calculus

N. Shankar*
Computer Science Laboratory
SRI International
Menlo Park, CA 94025
email: shankar@csl.sri.com

November 8, 1991

Abstract

The use of Herbrand functions (more popularly known as *Skolemization*) plays an important role in classical theorem proving and logic programming. We define a notion of Herbrand functions for the full intuitionistic predicate calculus. The definition is based on the view that the proof-theoretic role of Herbrand functions (to replace universal quantifiers), and of unification (to find instances corresponding to existential quantifiers), is to ensure that the eigenvariable conditions on a sequent proof are respected. The propositional impermutabilities that arise in the intuitionistic but not the classical sequent calculus motivate a generalization of the classical notion of Herbrand functions. Proof search using generalized Herbrand functions also provides a framework for generalizing logic programming to subsets of intuitionistic logic that are larger than Horn clauses. The search procedure described here has been implemented and observed to work effectively in practice. The generalization of Herbrand functions can also be applied to sequent calculi formalizations of logics other than intuitionistic predicate calculus.

1 Introduction

Proof search is an effective way to do automated theorem proving since there is more meaningful information in a proof or a failed proof than there is in a theorem or a failed conjecture. Intuitionistic proofs are interesting because they contain more information than the corresponding classical proofs. Intuitionistic logics can be shown to contain classical logic via Gödel's double-negation interpretation [Göd86b]. However, proof search (and theorem proving) is significantly more difficult for the intuitionistic predicate calculus than for the classical predicate calculus. There are no convenient normal forms such as the prenex, conjunctive, disjunctive forms.

This paper describes an effective technique for carrying out automated proof search in the intuitionistic sequent calculus. Sequent calculi are an excellent foundational medium for studying the mathematics of proof search for several reasons. There is a vast body of proof-theoretic research on sequent calculi and the significant metatheorems turn out to be simpler in the context of sequent calculi. Many other proof search techniques such as the tableau and matrix methods can be viewed as optimizations of sequent calculus proof search and the sequent calculus makes it possible to study these optimizations in their full generality. The technique described here is actually extremely

*Part of this work was performed while the author was at Stanford University and was funded by NSF Grant No. CCR-8718605. The preparation of this paper has been supported by the SRI International Computer Science Laboratory.

general and can be applied to proof search in any sequent calculus with conventional quantifier rules and a cut-elimination theorem. The technique is described just for the intuitionistic calculus in order to keep the exposition concrete.

Proof search techniques for classical sequent calculi are fairly well-known [BB74, Bow82, KW84]. These techniques work by Herbrandizing[1] the goal sequent to eliminate quantifiers and then use propositional reasoning and unification to carry out the proof search on the resulting quantifier-free sequent. Consider the attempt to search for a proof of the following sequent which is unprovable in classical logic:

$$(\forall y\colon (\exists x\colon p(x, y))) \vdash (\exists x\colon (\forall y\colon p(x, y))).$$

In its Herbrandized form, the quantiers of existential strength have been replaced by variables (to be instantiated during proof search) and the quantifiers of universal strength have been replaced by terms, $f(u)$ and $g(v)$ where f and g are newly chosen. The result is:

$$p(f(u), u) \vdash p(v, g(v)).$$

Here, f and g are the Herbrand functions, and u and v are called the Herbrand variables. The Herbrand terms $f(u)$ and $g(v)$ mention only the Herbrand variables corresponding to the surrounding quantifiers of existential strength. The propositional rules of Gentzen's sequent calculus are applied backwards, *i.e.*, from conclusion to premise(s). A branch of the proof is successfully terminated when a formula to the right of the sequent arrow unifies with a formula to the left of the sequent arrow. The resulting unifier is propagated to the next remaining branch and the process is repeated. Proof search on the example above fails because $p(f(u), u)$ and $p(v, g(v))$ do not unify.[2]

The above system of proof search has several advantages. Proof search is transparent, since it closely follows the proof rules of the sequent calculus. User interaction becomes more manageable. The choice of the term to be existentially generalized is postponed to the axiom case where the term can be constructed by means of unification. This means that there is less backtracking and less wasted effort in constructing a proof. In fact in the classical case, there is backtracking only on the selection of unifiers and not on the order of the proof steps. It is worth emphasizing this point: *naive sequent calculus proof search without the use of Herbrand functions would be entirely impractical due to the amount of backtracking involved.*

The present paper demonstrates that the above classical sequent based method can be adapted for theorem proving in the intuitionistic predicate calculus. The modifications require a careful proof-theoretic analysis of the role of Herbrand functions and variables, and of unification in proof search [Her67]. The crucial aspect of our proof-theoretic approach is the observation that Herbrand functions along with unification, serve to enforce the eigenvariable conditions on a sequent calculus derivation. The approach of Herbrandizing the goal formula prior to proof search does not work for nonclassical calculi such as the intuitionistic one. Herbrand functions have to be introduced during the proof search. One obvious approach to introducing a Herbrand function f corresponding to

[1] Skolemization is the process of eliminating quantifiers from a sentence so that free variables replace (essentially) universally quantified variables, and Skolem terms of the form $f(u_1, \ldots, u_n)$ replace (essentially) existentially quantified variables that are within the scope of the universal quantifiers binding the u_i. A sentence is satisfiable iff the universal closure of its Skolemized form is satisfiable. The Herbrandized form of a sentence is the Skolemization of its negation. It is also the case that Herbrand and Gentzen deserve the credit for the proof-theoretic insights, whereas Skolem paid scant attention to proof theory and missed claiming important results such as the completeness of first-order logic.

[2] The actual explanation of proof search is a little more complicated since it needs to account for the copying of formulas corresponding to the use of the contraction rule in a proof. Proof search on the formula, $(\exists x\colon (\forall y\colon A(x) \supset A(y)))$, is illustrative of the need for copying, renaming Herbrand variables, etc.. The examples chosen in this paper skirt these issues. We hope to discuss the extremely rich subject of classical proof search in a future paper.

universal quantifier steps is to make the Herbrand term $f(u_1, \ldots, u_n)$ depend on all the Herbrand variables $u_1, \ldots, u_n$, introduced in the existential quantifier steps in the previous steps of the proof search. This approach[3] does take the guesswork out of discovering the term to be existentially generalized but has no other advantage over nondeterministically guessing the entire proof. What is missing is the ability to also ignore the order of the quantifier steps during the proof search. As an example, consider the use of this technique for proof search on the (classically and intuitionistically) provable sequent

$$(\exists x\colon (\forall y\colon p(x, y))) \vdash (\forall y\colon (\exists x\colon p(x, y))).$$

If proof search on this sequent proceeds by first applying the left quantifier rules and then the right quantifier rules, we would arrive at the sequent $p(a, y) \vdash p(x, f(y))$ where unification fails due to the occurs-check as a result of the spurious dependency between f and y. This form of Herbrand function introduction would make proof search hopelessly inefficient since it could require backtracking over all the possible orders in which the quantifier steps could occur. Our observation is that a Herbrand term does not have to depend on all the currently active Herbrand variables but only on a minimal set of these variables. The set of these Herbrand variables can be computed using the impermutabilities between the sequent calculus rules. The use of Herbrandization to record just the impermutabilities in the intuitionistic sequent calculus reduces (but clearly does not eliminate) the amount of backtracking since the relative order of the permutable inferences in the proof search are rendered irrelevant. We show how the unifier returned by a successful proof search can be used to derive the order of introduction of quantifier rules in order to to construct a correct sequent calculus proof by permuting the propositional proof returned by the search procedure. We can also show that the the dependencies between Herbrand functions and variables cannot (in a sense) be further minimized since this would make the search procedure unsound. Our results also indicate which classes of formulas can and cannot be Herbrandized initially as can be done with classical logic. For example in classical logic, since all the non-quantifier rules can be permuted, Herbrandization is a sound step in proof search. We can also demonstrate the failure of the Herbrand theorem for heriditary Harrop sentences [MNPS91].

There are a number of results in the literature directed at mechanical theorem proving for intuitionistic logic. We list only a few of these below. Mints [Min88] describes a resolution-style proof method for intuitionistic logic and a general technique for deriving resolution-style systems from sequent calculi. Beeson [Bee88], for example has implemented a Prolog proof search procedure called GENTZEN for an intuitionistic sequent calculus. Felty and Miller [FM88] have written a λ-*Prolog* procedure for proof search in the intuitionistic sequent calculus that maintains the eigenvariable conditions through the use of higher-order unification and hereditary Harrop program clauses [MNPS91, NM90]. The procedure we describe has a significant efficiency advantage over the approaches of Beeson, and Felty and Miller, since it postpones commitments on the order of introduction of certain quantifiers until the terminal nodes of the search tree are reached. Wallen [Wal90] uses Gödel's translation [Göd86a] of intuitionistic logic into the modal logic S4 to define a matrix proof procedure for intuitionistic logic that employs string unification to ensure that the introduction rules for S4 modalities are respected. While Wallen's method is an ingenious and effective one, the translation into S4, whether implicit or explicit, complicates the intuitive reading of the given intuitionistic formula and its proof. Pym and Wallen [PW90] present an approach to proof search in an intuitionistic type theory that is similar to our approach but they do not exploit Herbrand functions to encode impermutabilities during the proof search.

[3] The technique of introducing Herbrand functions in this manner is fairly well-known in automated reasoning. Fitting [Fit90] describes this technique as applied to tableau systems.

2 Proof Theoretic Background

We first fix some of the syntax. The alphabet (along with the metavariable conventions) consists of

- the logical symbols: $\neg, \supset, \vee, \wedge, \forall, \exists$;
- the punctuation symbols: '(', ',', ':', ')';
- the variable symbols represented by the metavariables x, y, z;
- the function symbols (with arity) represented by f, g, h;
- the constant symbols represented by c, d, e;
- the predicate symbols (with non-negative arity) represented by p, q, r;
- the parameters represented by a, b.

A *term* is either a constant, a variable or of the form $f(t_1, \ldots, t_n)$, where f is an n-ary function symbol and $t_1, \ldots, t_n$ are terms. The metavariables s and t range over terms. An *atomic formula* is an n-ary predicate symbol followed by n terms. A *formula* is either an atomic formula or of the form $\neg A, A \supset B, A \vee B, A \wedge B, (\forall x{:}A), (\exists x{:}\, A)$, where A and B are smaller formulas and x is a variable. A *sequent* is of the form $\Gamma \vdash \Delta$, where Γ and Δ are multisets of formulas. If Γ is of the form $A_1, \ldots, A_m$ and Δ is of the form $B_1, \ldots, B_n$, then the sequent $\Gamma \vdash \Delta$ can be interpreted as

$$A_1 \wedge \ldots \wedge A_m \supset B_1 \vee \ldots \vee B_n$$

where the empty conjunction is taken to be logically true, and the empty disjunction is logically false.

The notions of positive and negative occurrences of subformulas in a formula and in a sequent should be clear. A positive occurrence of $\exists$ or a negative occurrence of $\forall$ is labelled an *existential* quantifier. Correspondingly, a positive $\forall$ occurrence or a negative $\exists$ occurrence is an *universal* quantifier. So existential and universal quantification refer to the strengths of the quantification and not to the quantifier symbol that is used.

It should be noted that Γ and Δ are viewed as (possibly empty) multisets of formulas rather than lists or sets. Both Γ, A and A, Γ represent the multiset union of $\{A\}$ and Γ, and Γ_1, Γ_2 represents the multiset union of Γ_1 and Γ_2. An additional piece of syntax is required for these rules: terms are extended to include parameters a, b, used in the quantifier rules below.

It is useful to present the sequent calculi for both classical logic, called LK, and for intuitionistic logic, called LJ [Sza69]. The presentation of LK in Figure 2 is essentially the same as Kleene's G3 [Kle52a].

Figure 2 displays the rules for LJ and these are similar to those of LK with the important restriction that the right hand side of any sequent in a proof can contain at most one formula.

A proof using these rules of LK or LJ is a tree rooted from below by the conclusion sequent, where the leaf sequents are all axioms, and each non-leaf sequent is derived from its premise sequents by a rule application. The systems LK and LJ have some important properties [Kle52a, Sza69].

Proposition 2.1 (Cut Elimination) *The Cut rule is redundant in both LK and LJ:*

1. *If* $\Gamma \vdash_{LK} \Delta$, *then* $\Gamma \vdash_{\{LK-Cut\}} \Delta$.
2. *If* $\Gamma \vdash_{LJ} \Delta$, *then* $\Gamma \vdash_{\{LJ-Cut\}} \Delta$.

Left	Right
$\dfrac{}{\Gamma, A \vdash A, \Delta}(Ax)$	
$\dfrac{\Gamma, \neg A \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta}(\neg \vdash)$	$\dfrac{\Gamma, A \vdash \neg A, \Delta}{\Gamma \vdash \neg A, \Delta}(\vdash \neg)$
$\dfrac{\Gamma, (A \supset B), B \vdash \Delta \qquad \Gamma \vdash A, \Delta}{\Gamma, (A \supset B) \vdash \Delta}(\supset \vdash)$	$\dfrac{\Gamma, A \vdash B, (A \supset B), \Delta}{\Gamma \vdash (A \supset B), \Delta}(\vdash \supset)$
$\dfrac{\Gamma, (A \vee B), A \vdash \Delta \qquad \Gamma, (A \vee B), B \vdash \Delta}{\Gamma, (A \vee B) \vdash \Delta}(\vee \vdash)$	$\dfrac{\Gamma \vdash A, B, (A \vee B), \Delta}{\Gamma \vdash (A \vee B), \Delta}(\vdash \vee)$
$\dfrac{\Gamma, (A \wedge B), A, B \vdash \Delta}{\Gamma, (A \wedge B) \vdash \Delta}(\wedge \vdash)$	$\dfrac{\Gamma \vdash A, (A \wedge B), \Delta \qquad \Gamma \vdash B, (A \wedge B), \Delta}{\Gamma \vdash (A \wedge B), \Delta}(\vdash \wedge)$
$\dfrac{\Gamma \vdash C, \Delta \qquad \Gamma, C \vdash \Delta}{\Gamma \vdash \Delta}Cut(C)$	
$\dfrac{\Gamma, (\forall x\colon A), A\{t/x\} \vdash \Delta}{\Gamma, (\forall x\colon A) \vdash \Delta}(\forall \vdash)$	$\dfrac{\Gamma \vdash A\{a/x\}, (\forall x\colon A), \Delta}{\Gamma \vdash (\forall x\colon A), \Delta}(\vdash \forall)^*$
$\dfrac{\Gamma, (\exists x\colon A), A\{a/x\} \vdash \Delta}{\Gamma, (\exists x\colon A) \vdash \Delta}(\exists \vdash)^*$	$\dfrac{\Gamma \vdash A\{t/x\}, (\exists x\colon A), \Delta}{\Gamma \vdash (\exists x\colon A), \Delta}(\vdash \exists)$

*: a not free in Γ, Δ (*eigenvariable* condition).

Figure 1: Rules for *LK*

Note that $|\Delta| \leq 1$.

Left	Right
$\dfrac{}{\Gamma, A \vdash A}(Ax)$	
$\dfrac{\Gamma, \neg A \vdash A}{\Gamma, \neg A \vdash \Delta}(\neg \vdash)$	$\dfrac{\Gamma, A \vdash \neg A}{\Gamma \vdash \neg A}(\vdash \neg)$
$\dfrac{\Gamma, (A \supset B), B \vdash \Delta \qquad \Gamma \vdash A}{\Gamma, (A \supset B) \vdash \Delta}(\supset \vdash)$	$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash (A \supset B)}(\vdash \supset)$
$\dfrac{\Gamma, (A \vee B), A \vdash \Delta \qquad \Gamma, (A \vee B), B \vdash \Delta}{\Gamma, (A \vee B) \vdash \Delta}(\vee \vdash)$	$\dfrac{\Gamma \vdash A}{\Gamma \vdash (A \vee B)}(\vdash \vee_1) \qquad \dfrac{\Gamma \vdash B}{\Gamma \vdash (A \vee B)}(\vdash \vee_2)$
$\dfrac{\Gamma, (A \wedge B), A, B \vdash \Delta}{\Gamma, (A \wedge B) \vdash \Delta}(\wedge \vdash)$	$\dfrac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash (A \wedge B)}(\vdash \wedge)$
$\dfrac{\Gamma \vdash C \qquad \Gamma, C \vdash \Delta}{\Gamma \vdash \Delta}Cut(C)$	
$\dfrac{\Gamma, (\forall x\colon A), A\{t/x\} \vdash \Delta}{\Gamma, (\forall x\colon A) \vdash \Delta}(\forall \vdash)$	$\dfrac{\Gamma \vdash A\{a/x\}}{\Gamma \vdash (\forall x\colon A)}(\vdash \forall)^*$
$\dfrac{\Gamma, (\exists x\colon A), A\{a/x\} \vdash \Delta}{\Gamma, (\exists x\colon A) \vdash \Delta}(\exists \vdash)^*$	$\dfrac{\Gamma \vdash A\{t/x\}}{\Gamma \vdash (\exists x\colon A)}(\vdash \exists)$

*: a not free in Γ, Δ (*eigenvariable condition* condition).

Figure 2: Rules for *LJ*

The cut rule is the only rule in which there is a formula appearing in the premise sequent(s) that is not a subformula of its conclusion sequent. Substituting the term t for the free occurrences of x in A yields $A\{t/x\}$, where the bound variables in A are renamed to avoid any clashes. Note that $A\{t/x\}$ for any t is taken to be a subformula of both $(\forall x\colon A)$ and $(\exists x\colon A)$. Cut-free proofs therefore have the subformula property, i.e., every formula occurrence in the proof is a subformula of all the sequents appearing below it, and in particular, of the conclusion sequent of the proof. Each of the non-cut rules introduces exactly one formula into the conclusion sequent, the *principal formula* of the rule.

Proposition 2.2 (Subformula Property) *The formulas in a sequent in a cut-free LK or LJ proof are all subformulas of the conclusion sequent of the proof.*

The subformula property can be used to argue the nonprovability of $\vdash A \vee \neg A$ in LJ since the final rule of inference in a cut-free proof must be $(\vdash \vee)$, but neither of the premises of the rule application are provable in LJ.

Proposition 2.3 (Permutability) *The order in which the rules occur in a cut-free LK proof can be permuted without changing the conclusion of the proof, provided:*

1. *The eigenvariable conditions are not violated.*
2. *The subformula property is not violated.*

In particular, the order of the rules in a propositional proof is permutable so long as the subformula property is maintained. For instance, the rules in the proof

$$\dfrac{\dfrac{A, B \vdash B \quad A, B \vdash A}{A, B \vdash B \wedge A}(\vdash \wedge)}{A \wedge B \vdash B \wedge A}(\wedge \vdash)$$

can be permuted as

$$\dfrac{\dfrac{A, B \vdash B}{A \wedge B \vdash B}(\wedge \vdash) \quad \dfrac{A, B \vdash A}{A \wedge B \vdash A}(\wedge \vdash)}{A \wedge B \vdash B \wedge A}(\vdash \wedge)$$

There are certain additional impermutable pairs of inferences in LJ as will be described in Section 3 below. The properties of the sequent calculi play a significant role in justifying the proof search procedures.

3 Proof Search in LJ

In this section, we present and analyze a proof search predicate, *Search*, that searches for LJ proofs. We omit the details of classical proof search though we do make informal references to it.

To examine why intuitionistic proof search is different from classical proof search, we consider the following sentence that is classically provable but is not intuitionistically provable:

$$(\forall x\colon (p(x) \vee q)) \supset (q \vee (\forall x\colon p(x))).$$

A classical proof search would proceed by Herbrandizing this formula to form the sequent

$$\vdash (p(u) \vee q) \supset (q \vee p(c))$$

then applying the ($\vdash\supset$) and ($\vdash\vee$) rule to get

$$(p(u) \vee q) \vdash q, p(c)$$

which by ($\vee\vdash$) reduces to the two subgoals

$$\begin{array}{rcl} p(u) & \vdash & q, p(c) \\ q & \vdash & q, p(c). \end{array}$$

The second of these subgoals is a propositional axiom. The first subgoal is established by instantiating the Herbrand variable u with the Herbrand constant c.

In searching for a sequent calculus proof as shown above, each application of a proof rule to transform a sequent matching the conclusion of the rule to the corresponding premises of the rule is termed a *reduction*.

We can attempt a similar proof search with the Herbrandized sentence using only the rules of LJ. Starting with

$$\vdash (p(u) \vee q) \supset (q \vee p(c))$$

we apply ($\vee\vdash$) to get the subgoals

$$\begin{array}{rcl} p(u) & \vdash & q \vee p(c) \\ q & \vdash & q \vee p(c). \end{array}$$

The first subgoal can be completed by applying ($\vdash\vee_1$) instantiating x with c. Applying ($\vdash\vee_2$) to the second subgoal reduces it to an axiom, and completes the proof search successfully.

Since we started with an intuitionistically unprovable sentence, there must be a flaw in the above approach to intuitionistic proof search that caused it to terminate successfully. The flaw is highlighted when we attempt to reconstruct the proof found by the search procedure as shown below:

$$\dfrac{\dfrac{\dfrac{\dfrac{p(a) \vdash p(a)}{p(a) \vdash (\forall x\colon p(x))}(\vdash\forall)}{p(a) \vdash q \vee (\forall x\colon p(x))}(\vdash\vee) \qquad \dfrac{q \vdash q}{q \vdash q \vee (\forall x\colon p(x))}(\vdash\vee)}{(\forall x\colon (p(x) \vee q)) \vdash (q \vee (\forall x\colon p(x)))}(\vee\vdash)}{\vdash (\forall x\colon (p(x) \vee q)) \supset (q \vee (\forall x\colon p(x)))}(\vdash\supset)$$

The above "proof" obviously violates the eigenvariable condition in the ($\vdash\forall$) step. The crucial observation however is that there is no way to repair this violation of the eigenvariable condition by changing the order of inferences in the proof. In the classical sequent calculus LK, it would be possible to change the order of the rules as required since ($\vee\vdash$) rule permutes above the ($\vdash\vee$) rule. The reordered LK proof does satisfy the eigenvariable condition. A similar reordering of inferences is not permitted in LJ since the only way to prove $p(c) \vee q \vdash q \vee p(c)$ is with the ($\vee\vdash$) rule applied below the ($\vdash\vee$) rule.

We therefore observe that the impermutability of certain pairs of inferences in LJ makes it incorrect to directly use Herbrandization for proof search. Our solution to this problem is to dynamically introduce Herbrand functions to replace universal quantifiers during the proof search so that the variables $u_1, \ldots, u_n$ in the Herbrand term $f(u_1, \ldots, u_n)$ can encode the propositional impermutabilities that arise in LJ proofs. In the above example, the corrected form of proof search proceeds as follows. We start with the sequent

$$\vdash (\forall x\colon (p(x) \vee q)) \supset (q \vee (\forall x\colon p(x)))$$

applying ($\vdash\supset$) to get

$$(\forall x\colon (p(x) \vee q)) \vdash (q \vee (\forall x\colon p(x)))$$

then introducing the Herbrand variable u corresponding to the ($\forall\vdash$) rule to get

$$p(u) \vee q \vdash (q \vee (\forall x\colon p(x))).$$

The ($\vee\vdash$) rule must then be applied to yield the subgoals

$$\begin{array}{rcl} p(u) & \vdash & (q \vee (\forall x\colon p(x))) \\ q & \vdash & (q \vee (\forall x\colon p(x))). \end{array}$$

The second of these subgoals is easily proved by applying the ($\vdash\vee_1$) rule. For the first subgoal, we clearly will apply the reduction corresponding to the ($\vdash\vee_2$) rule, but in doing so we note that the ($\vee\vdash$) rule might not permute above a ($\vdash\vee$) rule and the Herbrand variable u governs the ($\vee\vdash$) step. Therefore the Herbrand term corresponding to the ($\vdash\forall$) rule is introduced as

$$p(u) \vdash p(f(u)).$$

Occurs-checking during unification ensures that the Herbrand variable u cannot be instantiated with a term containing f. In the above case, the proof search rightly fails since occurs-checking prevents u from being unified with $f(u)$. There are of course other directions in which the above proof search could have proceeded but they all fail rather more easily. We will show that if a Herbrand variable u is instantiated with a term t containing a Herbrand function f, the proof structure returned by a successful proof search can always be permuted so that the existential quantifier corresponding to u can be introduced above the universal quantifier corresponding to f in order to satisfy the eigenvariable condition.

The restrictions on the succedent of LJ sequents leads to a number of impermutabilities that are not present in LK. Kleene [Kle52b] lists the basic impermutabilities along with examples of theorems whose LJ proofs can only be constructed with R1 above R2 (i.e., R1/R2). These are displayed in Figure 3. In each instance of R1/R2, the rule R2 cannot be permuted above the rule R1 in any proof of the corresponding example. It can be checked that for any pair of rules R1 and R2 not listed below, an occurrence of R2 can always be moved above an occurrence of R1 in an LJ proof.

We call the impermutabilities 4–11 the LJ impermutabilities since they are peculiar to LJ. We first observe certain uniformities in the LJ impermutabilities above:

- In any propositional impermutability R1/R2 above, the only Left rule in the R2 position is ($\vee\vdash$).
- The only Right rules are ($\vdash\supset$) and ($\vdash\neg$).
- In both the cases (when R2 is either ($\vdash\supset$) or ($\vdash\neg$)), R1 is either ($\supset\vdash$) or ($\neg\vdash$).

These uniformities are exploited in the proof search procedure below. The proof search procedure is presented as a predicate that takes five arguments:

1. A (partially Herbrandized) sequent consisting of a set of zero or more antecedent forms and at most one succedent form. Each *form* is a pair $[X\colon A]$ consisting of a formula A and the set X of governing Herbrand variables. The initial sequent to the proof search should not contain any free variables, and the set of governing variables in each form is empty.

	R1/R2	Example
1.	$(\forall\vdash)/(\vdash\forall)$	$(\forall x\colon A(x)) \vdash (\forall x\colon (A(x) \vee B))$
2.	$(\forall\vdash)/(\exists\vdash)$	$(\forall x\colon A(x)), (\exists x\colon \neg A(x)) \vdash$
3.	$(\vdash\exists)/(\exists\vdash)$	$(\exists x\colon (A(x) \wedge B)) \vdash (\exists x\colon A(x))$
4.	$(\supset\vdash)/(\vdash\supset)$	$(A \supset \neg A) \vdash (A \supset B)$
5.	$(\neg\vdash)/(\vdash\supset)$	$\neg A \vdash (A \supset B)$
6.	$(\supset\vdash)/(\vdash\neg)$	$(A \supset \neg A) \vdash \neg A$
7.	$(\neg\vdash)/(\vdash\neg)$	$\neg A \vdash \neg(A \wedge A)$
8.	$(\supset\vdash)/(\vee\vdash)$	$A \vee B, A \supset B \vdash B$
9.	$(\vdash\vee)/(\vee\vdash)$	$(A \vee B) \vdash (B \vee A)$
10.	$(\neg\vdash)/(\vee\vdash)$	$(A \vee B), \neg A \vdash B$
11.	$(\vdash\exists)/(\vee\vdash)$	$A(a) \vee A(b) \vdash (\exists x\colon A(x))$

Figure 3: Impermutabilities in LJ

2. A set of Herbrand variables, *lvars*, that govern any forms to which the $(\vee\vdash)$ rule has been applied in the course of the proof search so far.

3. A set of Herbrand variables, *rvars*, that govern any forms to which the $(\vdash\supset)$ or $(\vdash\neg)$ has been applied in the course of proof search so far.

4. The input unifier, which is initially the empty unifier.

5. The output unifier that is the outcome of the proof search.

The predicate *Search*(*sequent*; *lvars*; *rvars*; *input*; *output*) can then be described by a collection of Horn clauses. The search sequent is represented as $\Sigma \vdash \Theta$, where Σ is a (possibly empty) multiset of forms, and Θ contains at most one form. Sets of variables are represented by X, Y, L, R. Multisets of forms are represented by Σ, Θ, etc. Substitutions are represented by the letter U, V, W, etc. We omit the definition of *Unify*$(A; B; U; V)$ while noting that V is required to be a unifier for A and B that extends U. That is, V is of the form $U' \circ U$ for some substitution U'. If such a unifier does not exist, then the result V is **No**. The parameter U is not used in any essential way in the definition below, but does permit a variation of *Search* in which the unifier returned from one successful branch of the search can be given as the input to the other branch. The parameter U also turns out to be useful for the proof of completeness in Lemma 4.8. The relation *Search* can be defined inductively by the following scheme:

1. Unifier: $\mathit{Search}(\Sigma \vdash \Theta;\ L;\ R;\ \mathbf{No};\ \mathbf{No})$.

2. Axiom:

$$\mathit{Search}([X\colon A], \Sigma \vdash [Y\colon B];\ L;\ R;\ U;\ V) \text{ if}$$
$$\mathit{Unify}(A; B; U; V) \text{ and } V \neq \mathbf{No}.$$

3. Negation:

(a) Left: Note that X is augmented with $L \cup R$ to encode the impermutabilities of the form $(\neg\vdash)$/R2.

$$\begin{aligned} &Search([X\colon \neg A], \Sigma \vdash \Theta;\ L;\ R;\ U;\ V) \text{ if} \\ &\qquad Search([X\colon \neg A], \Sigma \vdash [X \cup L \cup R\colon\ A];\ L;\ R;\ U;\ V). \end{aligned}$$

(b) Right: Note that R is augmented with X to encode the impermutabilities of the form R1/$(\vdash\neg)$.

$$\begin{aligned} &Search(\Sigma \vdash [X\colon \neg B];\ L;\ R;\ U;\ V) \text{ if} \\ &\qquad Search([X;\ B], \Sigma \vdash\ ;\ L;\ X \cup R;\ U;\ V). \end{aligned}$$

4. Implication:

(a) Left: X is augmented with $L \cup R$ to encode the impermutabilities of the form $(\supset\vdash)$/R2. The construction of the unifier V can be carried out in stages by renaming V in one subgoal and U in the other to both be W. The same applies to the left-disjunction and right-conjunction steps.

$$\begin{aligned} &Search([X\colon (A \supset B)], \Sigma \vdash \Theta;\ L;\ R;\ U;\ V) \text{ if} \\ &\qquad\qquad Search([X\colon (A \supset B)], \Sigma \vdash [X \cup L \cup R\colon\ A];\ L;\ R;\ U;\ V), \\ &\qquad \text{and}\quad Search([X \cup L \cup R\colon\ B], [X\colon (A \supset B)], \Sigma \vdash \Theta;\ L;\ R;\ U;\ V). \end{aligned}$$

(b) Right: R is augmented with X to encode the impermutabilities of the form R1/$(\vdash\supset)$.

$$\begin{aligned} &Search(\Sigma \vdash [X\colon (A \supset B)];\ L;\ R;\ U;\ V) \text{ if} \\ &\qquad Search([X\colon A], \Sigma \vdash [X\colon B];\ L;\ X \cup R;\ U;\ V). \end{aligned}$$

5. Disjunction:

(a) Left: L is augmented with X to encode the impermutabilities of the form R1/$(\vee\vdash)$.

$$\begin{aligned} &Search([X\colon (A \vee B)], \Sigma \vdash \Theta;\ L;\ R;\ U;\ V) \text{ if} \\ &\qquad\qquad Search([X\colon B], [X\colon (A \vee B)], \Sigma \vdash \Theta;\ X \cup L;\ R;\ U;\ V), \\ &\qquad \text{and}\quad Search([X\colon A], [X\colon (A \vee B)], \Sigma \vdash \Theta;\ X \cup L;\ R;\ U;\ V). \end{aligned}$$

(b) Right: X is augmented with L to encode the impermutability $(\vdash\vee)/(\vee\vdash)$.

$$\begin{aligned} &Search(\Sigma \vdash [X\colon (A \vee B)];\ L;\ R;\ U;\ V) \text{ if} \\ &\qquad\qquad\qquad Search(\Sigma \vdash [L \cup X\colon\ A];\ L;\ R;\ U;\ V) \text{ and } V \neq \textbf{No} \\ &\qquad \text{or}\qquad Search(\Sigma \vdash [L \cup X\colon\ B];\ L;\ R;\ U;\ V) \text{ and } V \neq \textbf{No} \\ &\quad \text{otherwise}\quad V = \textbf{No}. \end{aligned}$$

6. Conjunction: Since neither $(\wedge\vdash)$ nor $(\vdash\wedge)$ is involved in any impermutability, the $\wedge$-steps behave normally.

(a) Left:

$$\begin{aligned} &Search([X\colon (A \wedge B)], \Sigma \vdash \Theta;\ L;\ R;\ U;\ V) \text{ if} \\ &\qquad Search([X\colon A], [X\colon B], [X\colon (A \wedge B)], \Sigma \vdash \Theta;\ L;\ R;\ U;\ V). \end{aligned}$$

(b) Right:

$$\begin{aligned} &Search(\Sigma \vdash [X\colon (A \wedge B)];\ L;\ R;\ U;\ V) \text{ if} \\ &\qquad Search(\Sigma \vdash [X\colon B];\ L;\ R;\ U;\ V) \\ &\text{and}\quad Search(\Sigma \vdash [X\colon A];\ L;\ R;\ U;\ V). \end{aligned}$$

7. Existential Quantifiers:

(a) Left:

$$\begin{aligned} &Search([X\colon (\exists x\colon A)], \Sigma \vdash \Theta;\ L;\ R;\ U;\ V) \text{ if} \\ &\qquad Search([X\colon A\{x \leftarrow f(X)\}], \Sigma \vdash \Theta;\ L;\ R;\ U;\ V), \end{aligned}$$

where f is a new n-ary function symbol, and if X is $\{u_1, \ldots, u_n\}$ then $f(X)$ represents $f(u_1, \ldots, u_n)$.

(b) Right: X is augmented with u since u now becomes a governing Herbrand variable, and with L due to the impermutability $(\vdash \exists)/(\vee \vdash)$.

$$\begin{aligned} &Search(\Sigma \vdash [X\colon (\exists x\colon A)];\ L;\ R;\ U;\ V) \text{ if} \\ &\qquad Search(\Sigma \vdash [\{u\} \cup L \cup X\colon A\{x \leftarrow u\}];\ L;\ R;\ U;\ V), \end{aligned}$$

where u is a new Herbrand variable.

8. Universal Quantifiers:

(a) Left:

$$\begin{aligned} &Search([X\colon (\forall x\colon A)], \Sigma \vdash \Theta;\ L;\ R;\ U;\ V) \text{ if} \\ &\qquad Search([\{y\} \cup X\colon A\{x \leftarrow y\}], [X\colon (\forall x\colon A)], \Sigma \vdash \Theta;\ L;\ R;\ U;\ V), \end{aligned}$$

where y is a new variable.

(b) Right:

$$\begin{aligned} &Search(\Sigma \vdash [X\colon (\forall x\colon A)];\ L;\ R;\ U;\ V) \text{ if} \\ &\qquad Search(\Sigma \vdash [X\colon A\{x \leftarrow f(X)\}];\ L;\ R;\ U;\ V), \end{aligned}$$

where f, X, and $f(X)$ are as in $(\exists \vdash)$.

The reader might wish to apply the procedure *Search* to the following sequents that are classically valid, but not intuitionistically.

1. $(A \supset (\exists x\colon B(x))) \vdash (\exists x\colon A \supset B(x))$.
2. $\neg(\forall x\colon A(x)) \vdash (\exists x\colon \neg A(x))$.
3. $((\forall x\colon A(x)) \supset B) \vdash (\exists x\colon A(x) \supset B)$.

The next step is to argue that *Search* is sound and complete.

4 Soundness and Completeness

Search is *sound* if whenever it returns a unifier other than **No** on a sequent, there is an LJ proof of that sequent. *Search* is *complete* if when given any sequent provable in LJ, *Search* nondeterministically returns a unifier (other than **No**). It is easy to see that *Search* is both sound and complete on the propositional part of LJ since it is a nondeterministic search procedure for cut-free proofs in this fragment.

For demonstrating completeness, we restrict the Herbrand functions to be taken from $h_1, h_2, \ldots$, and to each h_i we associate the parameter a_i. We restrict LJ proofs to only contain parameters a_i. If Σ is a multiset of forms, let $\underline{\Sigma}$ represent the corresponding multiset of formulas, but where any terms of the form $h_i(\ldots)$ in Σ have been replaced by the corresponding parameter a_i. Applying the substitution U to Σ yields the multiset of forms $U(\Sigma)$.

Lemma 4.1 *If $\Gamma \vdash \Delta$ is provable in LJ, then for any multisets of forms Σ and Θ such that Γ is $\underline{U(\Sigma)}$ and Δ is $\underline{U(\Theta)}$, we can find a V distinct from* **No** *where $Search(\Sigma \vdash \Theta;\ L;\ R;\ U;\ V)$.*

Proof. By induction on cut-free LJ proofs of $\Gamma \vdash \Delta$. Note that L and R do not play any role in this proof. We consider only a single case of the proof.

In the $(\vdash \exists)$ case, the given proof derives $\Gamma \vdash (\exists x\colon A)$ from $\Gamma \vdash A\{t/x\}$. The corresponding step in the search replaces $(\exists x\colon A)$ with $A\{u/x\}$. Let U in the induction hypothesis be replaced with $U\{t/u\}$ so that $\underline{U\{t/u\}([\{y\} \cup L \cup X\colon\ A\{u/x\}])}$ is just $A\{t/x\}$ and it can be shown that the required V is provided by the induction hypothesis.

The other cases are similar ■

When all the forms in Σ and Θ are of the form $[\{\}\colon\ A]$, and U is the empty substitution $\emptyset$ (which, of course, is idempotent), then $\underline{U(\Sigma)} \vdash \underline{U(\Theta)}$ is just the original sequent, and we get the completeness theorem below, where $\overline{\Gamma}$ represents the multiset of forms got by replacing each A in Γ with $[\{\}\colon\ A]$.

Theorem 4.2 (Completeness) *If $\Gamma \vdash \Delta$ is provable in LJ, then there is a V distinct from* **No** *such that $Search(\overline{\Gamma} \vdash \overline{\Delta};\ L;\ R;\ \emptyset;\ V)$ holds.*

The soundness argument is more delicate since it sheds light on the use of Herbrand functions. The goal is to show that if proof search succeeds on a sequent, then that sequent has a proof in LJ. The proof proceeds by constructing a proof search tree from a successful proof search and extracting an ordering on the quantifiers from the output unifier. We then show that the proof search tree can be permuted so as to respect the ordering on the quantifiers and that the resulting proof obeys the eigenvariable condition. Let Π be an *LJ proof structure* if it is an LJ proof that possibly violates the eigenvariable condition. A proof structure Π is *hygienic* in its use of parameter names if each parameter name a_i is associated with at most one quantifier step in Π.

Lemma 4.3 *If $Search(\Sigma \vdash \Theta;\ L;\ R;\ U;\ V)$ succeeds with $V \neq$* **No***, we can construct a corresponding hygienic LJ proof structure Π of $\underline{V(\Sigma)} \vdash \underline{V(\Theta)}$.*

Proof. By induction on the computation corresponding to *Search*. We only deal with a single case.

Consider the $(\vdash \forall)$ case when the induction hypothesis yields a proof structure Π_1 for $V(\Sigma) \vdash V([X\colon\ A\{x \leftarrow h_i(X)\}])$ corresponding to $Search(\Sigma \vdash [X\colon\ A\{x \leftarrow h_i(X)\}];\ L;\ R;\ U;\ V)$. Note that $\underline{V([X\colon A\{x \leftarrow h_i(X)\}])}$ is just $\underline{V([X\colon A\{x \leftarrow a_i\}])}$. It is, however, possible for $\underline{V(\Sigma)}$ to contain occurrences of a_i so that the proof tree below could violate the eigenvariable condition.

$$\dfrac{\begin{array}{c}\Pi_1\\ \vdots \\ \underline{V(\Sigma)} \vdash \underline{V([X\colon A\{x \leftarrow h_i(X)\}])}\end{array}}{\underline{V(\Sigma)} \vdash \underline{V([X\colon (\forall x\colon A)])}}(\vdash \forall)$$

The other cases are similar. ∎

Given a proof structure Π, if a rule occurrence R1 occurs above a rule occurrence R2 in Π, then R1>R2 in Π iff either

- R1/R2 is one of the *LJ* impermutabilities 1–11
- the principal formula of R1 is a proper subformula of the principal formula of R2
- or there is a rule occurrence R3 such that R1>R3>R2

Lemma 4.4 *If for any n, rule* R1 *occurs n steps above* R2 *in proof structure* Π *and* R1$\not>$R2 *in* Π, *then the order of rules in* Π *can be permuted to yield a new proof structure* Π' *with the same conclusion as* Π, *so that* R2 *occurs above* R1 *in* Π'. *If* Π *is hygienic, then there are no new pairs of quantifier rule occurrences that violate the eigenvariable condition in* Π'.

Proof. By induction on n. The details are omitted.

In a hygienic proof structure, the only way to introduce new violations of the eigenvariable condition would be to invert the order of an existential quantifier step R1 that was above an universal quantifier step R2 in Π. This never happens since we would have R1>R2 in this case. ∎

Lemma 4.5 *Let U be an idempotent substitution such that for any i, the search sequent $U(\Sigma) \vdash U(\Theta)$ contains (zero or more occurrences of) at most one term of the form $h_i(X)$. If Search($\Sigma \vdash \Theta$; L; R; U; V) returns an idempotent substitution V such that $V \neq$* **No**, *then we can find an idempotent V' such that Search($\Sigma \vdash \Theta$; L; R; U; V') and V' contains at most one term of the form $h_i(\ldots)$ for any Herbrand function h_i, namely $h_i(V'(X))$.*

Proof. By induction on the computation of *Search* taking care to return the most general idempotent unifier at every stage. ∎

Denote the quantifier step corresponding to the Herbrand term $h_i(X)$ as $\forall_i$, and the quantifer step corresponding to the Herbrand variables u_j as $\exists_j$.

Lemma 4.6 *If $\forall_i > \exists_j$ in a proof structure Π corresponding to a computation of* Search, *then $u_j \in X$ for the Herbrand term $h_i(X)$ introduced in the computation of* Search.

Proof. By induction according to the definition of R1>R2. ∎

Lemma 4.7 *Given that U satisfies the conditions of Lemma 4.5, Search($\Sigma \vdash \Theta$; L; R; U; V) holds, and V is an idempotent unifier, if $u_j \in X$ for the Herbrand term $h_i(X)$ in the computation of* Search, *then h_i does not occur in $V(u_j)$.*

Proof. Since by Lemma 4.5, we can ensure that V is such that h_i only occurs in the form $h_i(V(X))$ in V. If h_i occurs in $V(u_j)$, then either V is not idempotent or contains an infinite term since no finite term can be a proper subterm of itself. Both these possibilities are ruled out. ∎

Theorem 4.8 (Soundness) *If* $\underline{U(\Sigma)} \vdash \underline{U(\Theta)}$ *contains no parameters and* $Search(\Sigma \vdash \Theta;\ L;\ R;\ U;\ V)$ *succeeds with* $V \neq \mathbf{No}$, *then* $\underline{V(\Sigma)} \vdash \underline{V(\Theta)}$ *is provable in LJ.*

Proof. By Lemma 4.3, we can derive a proof structure Π corresponding to $Search(\Sigma \vdash \Theta;\ L;\ R;\ U;\ V)$. The proof of Theorem 4.8 is by induction on the number of pairs of quantifier rules that violate the eigenvariable condition in the proof structure Π. By Lemma 4.6, for any rule occurrences $\forall_i$ and $\exists_j$ in Π, if $\forall_i > \exists_j$ then $u_j \in h_i(X)$. Here $h_i(X)$ is the Herbrand term introduced in the quantifier step $\forall_i$. By Lemma 4.7, h_i does not occur in $V(u_j)$. Therefore, if h_i does occur in $V(u_j)$, then $\forall_i \not> \exists_j$ and by Lemma 4.4, $\forall_i$ can be permuted below $\exists_j$ in Π to yield the proof structure Π' with the same conclusion as Π.

Note that by Lemma 4.4, no new violations of the eigenvariable condition are introduced into Π'. We have thus succeeded in reducing the number of violations of eigenvariable condition and the induction hypothesis can be applied to the resulting proof structure Π'. ∎

5 Observations

We have given a search procedure for proofs in the intuitionistic sequent calculus and proved it to be sound and complete. The key aspect of the search procedure is that we have generalized the notion of a Herbrand function so that the Herbrand term introduced for a universal quantifier depends only on those Herbrand variables corresponding to existential quantifier steps that cannot always be permuted above the universal quantifier step. This makes it possible to reconstruct the proof from the search structure and the unifier by permuting the search structure to reorder the quantifier steps to satisfy the order induced by the unifier.

The first observation is that both the search algorithm and proof can be generalized for any sequent calculus with a cut elimination theorem and conventional quantifier rules. The generalized procedure is to first identify the impermutabilities other than those given by the eigenvariable conditions in the given sequent calculus, and fashion a search procedure similar to *Search* to take note of these impermutabilities in forming Herbrand terms corresponding to universal quantifier steps. The above proof sketch would essentially apply without change. Note that in *Search*, we have taken advantage of certain patterns in the impermutability pairs to reduce the number of auxiliary parameters for accumulating Herbrand variables, and such optimizations may not always be possible.

The second observation is that we cannot improve upon this form of Herbrandization without examining the bodies of the quantified formulas themselves. In other words, if any Herbrand variables that are included in Herbrand terms in the search procedure above are dropped, then the procedure is unsound. It is possible to construct a counterexample of an unprovable sequent on which the search procedure would succeed. This does not preclude optimizations where the internal structure of the quantified formula is examined in order to rule out such counterexamples.

The generalization above explains why in classical logic, the goal sequent can be Herbrandized prior to proof search. In the classical sequent calculus, there are no impermutabilities other than those given by the eigenvariable conditions. So the Herbrand term corresponding to a universal quantifier depends on a Herbrand variable for an existential quantifier only if the universally quantified formula is a subformula of the existentially quantified formula. It might, of course, be necessary to make copies of formulas (while relabelling certain Herbrand variables) corresponding to the use of the contraction rule. Copying and relabelling are complicated issues that are beyond the scope of the current paper.

We can thus identify certain classes of formulas where Herbrandization will always succeed. These are classes of formulas that are defined solely by syntactic restrictions on the positive and negative occurrences of various connectives and quantifiers. Herbrandization will always work for a class of formulas that does not admit an LJ impermutability R1/R2, where an existential quantifier can govern the principal formula of R2, and the principal formula of R1 contains a universally quantified subformula. For example, Herbrandization fails for the class of hereditary Harrop formulas [MNPS91] since the unprovable formula

$$((\forall x\colon p(x)) \supset q) \supset (\exists y\colon (p(y) \supset q))$$

has the Herbrandized form

$$(p(c) \supset q) \supset (p(u) \supset q)$$

which is easily seen to be intuitionistically provable with c instantiating Herbrand variable u. This counterexample arises from the $(\supset\vdash)/(\vdash\supset)$ impermutability permitted by the class of hereditary Harrop formulas.

6 Conclusion

Proof search in the intuitionistic sequent calculus is relevant to logic programming [MNPS91]. We have presented a notion of Herbrand functions that is appropriate for proof search in the intuitionistic sequent calculus. A proof search procedure was described and proved to be sound and complete. This procedure has been implemented in Common Lisp. The implementation optimizes *Search* in some small ways but there are a number of deeper optimizations that remain to be implemented. The extent to which *Search* can be optimized without destroying the transparency of the proof search deserves further exploration.

Acknowledgements: The ideas and comments of Michael Beeson, Jussi Ketonen, Lincoln Wallen, Gianluigi Bellin, Dale Miller, Patrick Lincoln, and Sam Owre have contributed to this presentation. Sol Feferman's course on proof theory provided the initial stimulus for this work.

References

[BB74] W.W. Bledsoe and P. Bruell. A man-machine theorem-proving system. In *Advance Papers of Third International Joint Conference on Artificial Intelligence*. W.W. Bledsoe, 1974.

[Bee88] M. J. Beeson. Some applications of Gentzen's proof theory in automated deduction. Manuscript, 1988.

[Bow82] K. A. Bowen. Programming with full first-order logic. In J. E. Hayes, D. Michie, and Y.-H. Pao, editors, *Machine Intelligence 10*, pages 421–440. Halsted Press, 1982.

[Fit90] M. Fitting. *First-Order Logic and Automated Theorem Proving*. Springer-Verlag, 1990.

[FM88] A. Felty and D. Miller. Specifying theorem provers in a higher-order logic programming language. In E. Lusk and R. Overbeek, editors, *Ninth International Conference on Automated Deduction*, pages 61–80, Argonne, Illinois, May 1988.

[Göd86a] K. Gödel. An interpretation of the intuitionistic propositional calculus. In S. Feferman, editor, *Kurt Gödel: Collected Works Vol. 1*, pages 301–303. Oxford University Press, 1986.

[Göd86b] K. Gödel. On intuitionistic arithmetic and number theory. In S. Feferman, editor, *Kurt Gödel: Collected Works Vol. 1*, pages 287–295. Oxford University Press, 1986.

[Her67] J. Herbrand. Investigations in proof theory. In J. van Heijenoort, editor, *From Frege to Gödel: A Source Book of Mathematical Logic*, pages 525–581. Harvard University Press, Cambridge, Mass., 1967.

[Kle52a] S. C. Kleene. *Introduction to Metamathematics*. North-Holland, Amsterdam, 1952.

[Kle52b] S. C. Kleene. Permutability of inferences in Gentzen's calculi LK and LJ. *Memoirs of the A.M.S.*, 10, 1952.

[KW84] J. Ketonen and R. Weyhrauch. A decidable fragment of predicate calculus. *The Journal of Theoretical Computer Science*, 32:297–307, 1984.

[Min88] G. Mints. Gentzen-type systems and resolution rules, Part I: Propositional logic. In P. Martin-Löf and G. Mints, editors, *COLOG-88, Lecture Notes in Computer Science 417*, pages 198–231. Springer-Verlag, 1988.

[MNPS91] D. Miller, G. Nadathur, F. Pfenning, and A. Scedrov. Uniform proofs as a foundation for logic programming. *Annals of Pure and Applied Logic*, 51:125–157, 1991.

[NM90] G. Nadathur and D. Miller. Higher-order Horn clauses. *Journal of the ACM*, 37(4):777–814, 1990.

[PW90] D. J. Pym and L. A. Wallen. Investigations into proof-search in a system of first-order dependent function types. In M. E. Stickel, editor, *10th International Conference on Automated Deduction*, pages 236–250. Springer-Verlag, 1990.

[Sza69] M. E. Szabo, editor. *The Collected Papers of Gerhard Gentzen*. North-Holland, 1969.

[Wal90] L. A. Wallen. *Automated Proof Search in Non-Classical Logics*. MIT Press, 1990.