E LLO Proof Rules

I $$\frac{}{\overset{mgu(A,B,U)}{U} \vdash_{R,S,T} ?\Delta, [X : A], [Y : B^{\perp}]}$$

$\otimes$ $$\frac{\overset{W}{U} \vdash_{R,S,T} ?\Delta, \Sigma, [XST : A] \qquad \overset{V}{W} \vdash_{R,S,T} ?\Delta, [XST : B], ?}{\overset{V}{U} \vdash_{R,S,T} ?\Delta, \Sigma, [X : (A \otimes B)], ?}$$

$\parr$ $$\frac{\overset{V}{U} \vdash_{R,S \cup X,T} \Sigma, [X : A], [X : B]}{\overset{V}{U} \vdash_{R,S,T} \Sigma, [X : (A \parr B)]}$$

$\oplus$ $$\frac{\overset{V}{U} \vdash_{R,S,T} \Sigma, [XT : A]}{\overset{V}{U} \vdash_{R,S,T} \Sigma, [X : (A \oplus B)]}$$

$\oplus$ $$\frac{\overset{V}{U} \vdash_{R,S,T} \Sigma, [XT : B]}{\overset{V}{U} \vdash_{R,S,T} \Sigma, [X : (A \oplus B)]}$$

& $$\frac{\overset{W}{U} \vdash_{R,S,T \cup X} \Sigma, [X : A] \qquad \overset{V}{W} \vdash_{R,S,T \cup X} \Sigma, [X : B]}{\overset{V}{U} \vdash_{R,S,T} \Sigma, [X : (A \,\&\, B)]}$$

$?D$ $$\frac{\overset{V}{U} \vdash_{R,S,T} \Sigma, [X :?A], [XTR : A]}{\overset{V}{U} \vdash_{R,S,T} \Sigma, [X :?A]}$$

$!S$ $$\frac{\overset{V}{U} \vdash_{R \cup X,S,T} ?\Sigma, [Y : A]}{\overset{V}{U} \vdash_{R,S,T} ?\Sigma, [X :!A]}$$

$\perp$ $$\frac{\overset{V}{U} \vdash_{R,S,T} \Sigma}{\overset{V}{U} \vdash_{R,S,T} \Sigma, [X : \perp]}$$

1 $$\frac{}{\overset{U}{U} \vdash_{R,S,T} [X : 1]}$$

$\top$ $$\frac{}{\overset{U}{U} \vdash_{R,S,T} \Sigma, [X : \top]}$$

$\forall$ $$\frac{\overset{V}{U} \vdash_{R,S,T} \Sigma, [X : A\{h(X)/y\}]}{\overset{V}{U} \vdash_{R,S,T} \Sigma, [X : \forall y.A]}$$

$\exists$ $$\frac{\overset{V}{U} \vdash_{R,S,T} \Sigma, [XvT : A\{v/y\}]}{\overset{V}{U} \vdash_{R,S,T} \Sigma, [X : \exists y.A]}$$

Where h is new in the $\forall$ rules, v is new in the $\exists$ rules, and $mgu(A, B, U)$ returns the composition with U of a most general unifier of $U(A)$ and $U(B)$. The Y governing the formula A in the hypothesis of the **!S** rule is the union of S, T, and every Z such that $[Z : B]$ occurs in the conclusion.

C LLG Proof Rules

I $\dfrac{U(A) \equiv U(B)}{{}_U \vdash_T ?\Delta, A, B^{\perp}}$

Cut $\dfrac{{}_U \vdash_T \Sigma, A \qquad {}_U \vdash_T ?, A^{\perp}}{{}_U \vdash_T \Sigma, ?}$

$\otimes$ $\dfrac{{}_U \vdash_T ?\Delta, \Sigma, A \qquad {}_U \vdash_T ?\Delta, B, ?}{{}_U \vdash_T ?\Delta, \Sigma, (A \otimes B), ?}$

$\parr$ $\dfrac{{}_U \vdash_T \Sigma, A, B}{{}_U \vdash_T \Sigma, (A \parr B)}$

$\oplus$ $\dfrac{{}_U \vdash_T \Sigma, A}{{}_U \vdash_T \Sigma, (A \oplus B)} \qquad \dfrac{{}_U \vdash_T \Sigma, B}{{}_U \vdash_T \Sigma, (A \oplus B)}$

& $\dfrac{{}_U \vdash_T \Sigma, A \qquad {}_U \vdash_T \Sigma, B}{{}_U \vdash_T \Sigma, (A \,\&\, B)}$

$?D$ $\dfrac{{}_U \vdash_T \Sigma, ?A, A}{{}_U \vdash_T \Sigma, ?A}$

$!S$ $\dfrac{{}_U \vdash_T ?\Sigma, A}{{}_U \vdash_T ?\Sigma, !A}$

$\bot$ $\dfrac{{}_U \vdash_T \Sigma}{{}_U \vdash_T \Sigma, \bot}$

1 $\dfrac{}{{}_U \vdash_T ?\Delta, 1}$

$\top$ $\dfrac{}{{}_U \vdash_T \Sigma, \top}$

$\forall$ $\dfrac{{}_U \vdash_T \Sigma, A\{h(T)/x\}}{{}_U \vdash_T \Sigma, \forall x.A}$

$\exists$ $\dfrac{{}_{\{t/v\} \circ U} \vdash_{T \cup \{v\}} A\{v/x\}, \Sigma}{{}_U \vdash_T \Sigma, \exists x.A}$

D LLV Proof Rules

I $\dfrac{}{{}^{mgu(A,B,U)}_{U} \vdash_T ?\Delta, A, B^{\perp}}$

$\otimes$ $\dfrac{{}^{W}_{U} \vdash_T ?\Delta, \Sigma, A \qquad {}^{V}_{W} \vdash_T ?\Delta, B, ?}{{}^{V}_{U} \vdash_T ?\Delta, \Sigma, (A \otimes B), ?}$

$\parr$ $\dfrac{{}^{V}_{U} \vdash_T \Sigma, A, B}{{}^{V}_{U} \vdash_T \Sigma, (A \parr B)}$

$\oplus$ $\dfrac{{}^{V}_{U} \vdash_T \Sigma, A}{{}^{V}_{U} \vdash_T \Sigma, (A \oplus B)}$

$\oplus$ $\dfrac{{}^{V}_{U} \vdash_T \Sigma, B}{{}^{V}_{U} \vdash_T \Sigma, (A \oplus B)}$

& $\dfrac{{}^{W}_{U} \vdash_T \Sigma, A \qquad {}^{V}_{W} \vdash_T \Sigma, B}{{}^{V}_{U} \vdash_T \Sigma, (A \,\&\, B)}$

$?D$ $\dfrac{{}^{V}_{U} \vdash_T \Sigma, ?A, A}{{}^{V}_{U} \vdash_T \Sigma, ?A}$

$!S$ $\dfrac{{}^{V}_{U} \vdash_T ?\Sigma, A}{{}^{V}_{U} \vdash_T ?\Sigma, !A}$

$\bot$ $\dfrac{{}^{V}_{U} \vdash_T \Sigma}{{}^{V}_{U} \vdash_T \Sigma, \bot}$

1 $\dfrac{}{{}^{U}_{U} \vdash_T 1}$

$\top$ $\dfrac{}{{}^{U}_{U} \vdash_T \Sigma, \top}$

$\forall$ $\dfrac{{}^{V}_{U} \vdash_T \Sigma, A\{h(T)/y\}}{{}^{V}_{U} \vdash_T \Sigma, \forall y.A}$

$\exists$ $\dfrac{{}^{V}_{U} \vdash_{T \cup \{v\}} \Sigma, A\{v/y\}}{{}^{V}_{U} \vdash_T \Sigma, \exists y.A}$

A LL Permutabilities

An impermutability $R1/R2$ is represented in the table below on column labeled $R1$ and row labeled $R2$. For example $\otimes/\parr$ is indicated by the numeral 1. A numeral in the table should be read as "the connective of this column cannot always permute below the connective of this row". Below the table are a short list of examples of sequents which exhibit each impermutability. Impermutability 7 is present in most first-order logics. Classical logic enjoys all other possible permutabilities.

	$\otimes$	$\parr$	&	$\oplus$	$?D$	!	$\bot$	$\forall$	$\exists$
$\otimes$						0			
$\parr$	1					0			
&	2			3	4	0			5
$?D$									
!					6				
$\bot$						0			
$\forall$						0			7
$\exists$						0			

0 various examples
1 $\vdash (A \parr B), (A^\bot \otimes B^\bot)$
2 $\vdash (A\&B), (\bot \otimes \top), A^\bot, B^\bot$
3 $\vdash (A\&B), (A^\bot \oplus B^\bot)$
4 $\vdash ((!A)\&A), ?A^\bot$
5 $\vdash (A(t)^\bot \& A(u)^\bot), \exists x.A(x)$
6 $\vdash !(A^\bot \oplus B), ?A$
7 $\vdash \forall y.(A(y) \oplus B), \exists x.A(x)^\bot$

B LL Proof Rules

I $\dfrac{}{\vdash ?\Delta, A, A^\bot}$

Cut $\dfrac{\vdash \Sigma, A \qquad \vdash ?, A^\bot}{\vdash \Sigma, ?}$

$\otimes$ $\dfrac{\vdash ?\Delta, \Sigma, A \qquad \vdash ?\Delta, B, ?}{\vdash ?\Delta, \Sigma, (A \otimes B), ?}$

$\parr$ $\dfrac{\vdash \Sigma, A, B}{\vdash \Sigma, (A \parr B)}$

$\oplus$ $\dfrac{\vdash \Sigma, A}{\vdash \Sigma, (A \oplus B)} \qquad \dfrac{\vdash \Sigma, B}{\vdash \Sigma, (A \oplus B)}$

& $\dfrac{\vdash \Sigma, A \qquad \vdash \Sigma, B}{\vdash \Sigma, (A \,\&\, B)}$

$?D$ $\dfrac{\vdash \Sigma, ?A, A}{\vdash \Sigma, ?A}$

$!S$ $\dfrac{\vdash ?\Sigma, A}{\vdash ?\Sigma, !A}$

$\bot$ $\dfrac{\vdash \Sigma}{\vdash \Sigma, \bot}$

1 $\dfrac{}{\vdash ?\Delta, 1}$

$\top$ $\dfrac{}{\vdash \Sigma, \top}$

$\forall$ $\dfrac{\vdash \Sigma, A\{a/x\}}{\vdash \Sigma, \forall x.A}$

$\exists$ $\dfrac{\vdash \Sigma, A\{t/x\}}{\vdash \Sigma, \exists x.A}$

the context and to pass along the unused part of the context to the right branch of the $\otimes$ rule.

We show the timings for these procedures on some theorems and non-theorems in order to demonstrate the efficacy of the above optimizations. We first compare these procedures on provable sequents of the form $\vdash p^{\perp}, (\exists x.q_1^{\perp}(x)), \ldots, (\exists x.q_n^{\perp}(x)), (\forall x.q_1(x) \otimes \ldots q_n(x) \otimes p)$, for $n = 1, 2, 3, 4$. The run times are given to the nearest millisecond on BINProlog on a Sparcstation 10-41. The Prolog code is anonymously FTPable from `FTP.CSL.SRI.COM` in the file `pub/shankar/mall.pl`.

n	LLV	LLO	LLOpt
1	17	0	0
2	200	0	0
3	10433	17	0
4	1117383	83	17

The table above does not sufficiently emphasize the utility of LLOpt over LLO since there is a small overhead cost for LLOpt over LLO. The value of LLOpt is more apparent when the same experiment is attempted with the unprovable sequent $\vdash p^{\perp}, (\exists x.q_1^{\perp}(x)), \ldots, (\exists x.q_n^{\perp}(x)), (\forall x.q_1(x) \otimes \ldots q_n(x) \otimes r)$, for $n = 1, 2, 3, 4$.

n	LLV	LLO	LLOpt
1	33	17	0
2	916	1700	83
3	80783	199634	1300
4	-	-	33300

Note that in the case of a failed proof search, LLO actually performs worse than LLV because both procedures carry out the same search but LLO has the additional overhead of tracking impermutabilities. Note also that we have deliberately ordered the sequent so as to trigger the worst backtracking behavior with LLV in order to compare the worst-case performances of these procedures. Several further optimizations are currently being studied.

6 Conclusion

We have presented some optimized proof search procedures that can be applied to a wide variety of cut-free sequent calculi. We have presented rigorous informal arguments for the soundness and completeness of these optimizations. These optimized search procedures have been naively implemented in Prolog. The full paper will contain more informative versions of the proofs and more detailed empirical comparisons of the various optimizations.

Acknowledgments. We thank Sam Owre, Dale Miller, Andre Scedrov, Gopalan Nadathur, and James Harland for their help and guidance, and Serenella Cerrito for sharing an early draft of her paper [2] with us.

References

[1] J.-M. Andreoli. Logic programming with focusing proofs in linear logic. Journal of Logic and Computation, 1992. To appear.

[2] S. Cerrito. Herbrand methods in linear logic. 1992.

[3] C.-L. Chang and R.C-T. Lee. *Symbolic Logic and Mechanical Theorem Proving.* Academic Press, 1973.

[4] D. Galmiche and G. Perrier. Foundations of proof search strategies design in linear logic. In *Symposium on Logical Foundations of Computer Science*, St. Petersburg, 1994. To appear.

[5] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[6] J. Herbrand. Investigations in proof theory. In J. van Heijenoort, editor, *From Frege to Gödel: A Sourcebook of Mathematical Logic, 1879–1931*, pages 525–581. Harvard University Press, Cambridge, MA, 1967. First published 1930.

[7] J.S. Hodas and D. Miller. Logic programming in a fragment of intuitionistic linear logic. In *Proc. 6-th Annual IEEE Symposium on Logic in Computer Science, Amsterdam*, pages 32–42. IEEE Computer Society Press, Los Alamitos, California, July 1991. Full paper to appear in *Information and Computation.*

[8] M. Kanovich. Horn programming in linear logic is NP-complete. In *Proc. 7-th Annual IEEE Symposium on Logic in Computer Science, Santa Cruz, California*, pages 200–210. IEEE Computer Society Press, Los Alamitos, California, June 1992.

[9] S.C. Kleene. Permutability of inferences in Gentzen's calculi LK and LJ. *Memoirs of the AMS*, 1952.

[10] P. Lincoln, J. Mitchell, A. Scedrov, and N. Shankar. Decision problems for propositional linear logic. *Annals Pure Appl. Logic*, 56:239–311, 1992. Special Volume dedicated to the memory of John Myhill.

[11] P. Lincoln and A. Scedrov. First order linear logic without modalities is NEXPTIME-hard. To Appear in TCS, 1993. Available using anonymous ftp from host ftp.cis.upenn.edu and the file pub/papers/scedrov/mall1.dvi.

[12] P. Lincoln, A. Scedrov, and N. Shankar. Linearizing intuitionistic implication. In *Proc. 6-th Annual IEEE Symposium on Logic in Computer Science, Amsterdam*, pages 51–62. IEEE Computer Society Press, Los Alamitos, California, July 1991. Full paper to appear in Annals of Pure and Applied Logic.

[13] P. Lincoln and T. Winkler. Constant-Only Multiplicative Linear Logic is NP-Complete. To Appear in TCS, 1993.

[14] A. Martelli and U. Montanari. An efficient unification algorithm. *ACM Trans. on Prog. Lang. and Systems*, 4(2), Feb. 1982.

[15] N. Shankar. Proof search in the intuitionistic sequent calculus. In D. Kapur, editor, *Automated Deduction: CADE-11*, volume 607 of *lncs*, pages 522–536, Berlin, 1992. Springer-Verlag.

[16] T. Tammet. Proof search strategies in linear logic. Programming Methodology Group Report 70, Chalmers University, 1993.

At the cost of duplicating the proof of $\vdash B, \Delta$. For structural reasons, this kind of duplication of subproofs can only arise when a binary (2-premise) rule is permuted above another binary rule.

Since the permutation of binary steps can introduce duplication of subproofs, the permutation of bad proofs into good ones must be done carefully to achieve termination. In the first stage of the procedure, the binary high steps are permuted above the injured $\forall$ steps, with the highest binary high step first, so that no high binary steps below an injured $\forall$ are duplicated in the process. All the high binary steps can be eliminated in this manner, but this might create duplicates of the injured $\forall$ steps or the low steps. Next, we permute all the low binary steps below the bad $\exists$ step, with the lowest low steps first in order to avoid duplication of the low binary steps. This last step could create copies of the bad $\exists$ step. The only remaining high and low steps are unary steps. It is easy to permute these unary steps so that we get an LL proof of ${}^{V}_{U} \vdash_{R,S,T} (\exists x.A), ?$ that does not contain any bad $\exists$ steps. No bad $\exists$ steps are introduced by these permutations since each permutation preserves the consistency of the strict partial ordering derived from the LLO proof.

Given an LLO sequent ${}^{V}_{U} \vdash_{R,S,T} ?$, let $\underline{{}^{V}_{U} \vdash_{R,S,T} ?}$ represent the LL sequent $\vdash \underline{V(A_1)}, \ldots, \underline{V(A_n)}$, for each $[X_i : A_i]$ in $?$, where $\underline{B}$ represents the result of replacing each Herbrand term of the form $h_i(\ldots)$ in B by the corresponding eigenvariable a_i. The following lemmas are proved by induction on LLO proofs.

Lemma 4.2 *Given any idempotent substitution U, and an LLO proof of ${}^{V}_{U} \vdash_{R,S,T} ?$, there is a strict partial ordering on the steps in this proof respecting the subformula property, the eigenvariable condition, and the impermutabilities in this proof.*

Lemma 4.3 *An LLO proof of ${}^{V}_{U} \vdash_{R,S,T} ?$ can be transformed to an LL proof of $\underline{{}^{V}_{U} \vdash_{R,S,T} ?}$ with no bad $\exists$ steps but with possible violations of the eigenvariable condition.*

The soundness theorem is an easy consequence of the above lemma.

Theorem 4.4 *If $?$ does not contain any Herbrand functions or variables, then an LLO proof of ${}^{V}_{U} \vdash_{R,S,T} ?$ can be transformed to an LL proof of $\underline{{}^{V}_{U} \vdash_{R,S,T} ?}$.*

The soundness argument as outlined above is quite general and applies to any cut-free sequent calculus with the subformula property and conventional quantifier rules.

4.1 Other Cut-Free Sequent Calculi

One can generate a dynamic skolemization system along the lines of LLV for many other sequent calculi. For sequent calculi with cut-elimination theorem, subformula property, and the usual quantifier rules, one can replace the quantifier rules with the quantifier rules of LLV, decorate every sequent with set of governing variables, and input and output unifiers, and allow unification at any identity-like rules (that is, any rules requiring two formulas to be identical). The resulting LLV-like system replaces the guesswork of the usual existential rule with unification. Thus any cut-free first-order sequent system with the subformula property, the usual quantifier rules, and some finite depth bound on the sequent proof tree is decidable. Note that full linear logic (with !,?), relevance logic, first-order intuitionistic logic, and first-order classical logic, do not exhibit a finite depth bound on sequent proofs, due to the contraction rule. However, a proof system like LLV may still be of direct interest when considering logic programming in non-classical sequent systems.

It is also straightfoward to generate an optimized sequent system along the lines of LLO for any first-order cut-free sequent-calculi with the subformula property and the usual quantifier rules. The process begins with the analysis of the propositional permutabilities (such as displayed in Appendix A for linear logic). Then for each R2 rule (rules which do not always permute upward) a list of variables is created (like the R,S, and T for $!S$, $\parr$, and & in LLO). Then for each rule R2, for each rule R1 that does not permute below R2, the R2 variables are added to the governing variables of the subformulas of the principal formula. A system generated in this way will have no quantifier impermutabilities.

5 Implementation

We have implemented Prolog proof search procedures for the constant-free multiplicative-additive fragments of LLV, LLO, and a version of LLO called LLOpt where we have optimized the backtracking to exploit some propositional permutabilities. These optimizations are similar to those proposed by Andreoli [1], Galmiche and Perrier [4], Hodas and Miller [7], and Tammet [16]. The combination of LLO with optimizations based on propositional permutabilities is a delicate matter. The usual optimization is that if we have a rule R2 that is always permutable below any other rule, then it is safe to apply this rule immediately in the proof search. In the context of LLO, this would mean that for an impermutability R1/R2, the Herbrand variables governing R2 would always govern R1 so that proof attempts that might have succeeded with R1 below R2 now fail because of the occur check. For instance, given the provable sequent $\vdash (\exists x.p(x) \parr q(x)), r, (r^{\perp} \otimes (\forall y.p^{\perp}(y) \otimes ({}^{\perp}q(y))))$, the LLO proof search fails if the $\parr$ step occurs below all the $\otimes$ steps.

The LLO search procedure backtracks on all the rules. We can optimize this to avoid backtracking on $\parr$, $\forall$, and & rules provided there is backtracking on applications of the $\exists$ rule. The $\parr$ and $\forall$ steps can be applied immediately, but immediate reductions of the & steps could lead to larger proofs. The most significant optimization is in the treatment of the $\otimes$ rule. LLO tries all possible partitions of the context, whereas it turns out to be far more effective to allow the left branch of the rule to use up some part of

variables, and T, the & Herbrand variables. In addition, each sequent formula is decorated with the Herbrand variables that govern it. The quantifier rules have the form

$$\frac{\overset{V}{U} \vdash_{R,S,T} \Gamma, [X \, v \, T : A\{v/x\}]}{\overset{V}{U} \vdash_{R,S,T} \Gamma, [X : (\exists x.A)]}\exists$$

$$\frac{\overset{V}{U} \vdash_{R,S,T} \Gamma, [X : A\{h(X)/x\}]}{\overset{V}{U} \vdash_{R,S,T} \Gamma, [X : (\forall x.A)]}\forall$$

where v and h are new. The $\exists$ rule encodes the fact that the formula A is governed by the newly introduced Herbrand variable v and by those Herbrand variables that govern any & connective that is analyzed below the $\exists$ steps, thus encoding the impermutability $\exists/\&$. The $\forall$ rule is more straightforward. The other rules similarly encode the various impermutabilities by suitably augmenting either the sets R, S, or T, or the variable sets that are local to the sequent formulas.

The completeness of LLO is established by the following lemma which can be proved by induction on LLV proofs.

Theorem 4.1 *Let ? be of the form $A_1, \ldots, A_n$. If LLV proves $\overset{V}{U} \vdash_{T'} ?$, then for any $R, S, T, X_1, \ldots, X_n \subseteq T'$, there exists a W such that LLO proves $\overset{W}{U} \vdash_{R,S,T} [X_1 : A_1], \ldots, [X_n : A_n]$.*

We outline the soundness argument, and refer to a forthcoming full paper for the details. To prove the soundness of LLO, we need to show that when the LLO proof search succeeds and returns a unifier V, then the original goal sequent is provable in LL. The main obstacle is that if LLO proves a sequent, the resulting "proof" can contain violations of the eigenvariable condition in the following sense: the conclusion $\overset{V}{U} \vdash_{R,S,T} [X : (\forall x.A)], ?$ of a $\forall$ step introducing the Herbrand function h can be such that $V(?)$ contains occurrences of h. This must be because some Herbrand variable v in ? is introduced by an $\exists$ step that occurs below the $\forall$ step introducing h, where $V(v)$ contains h. Let such an $\exists$ step be labeled *bad*, and let a proof containing bad $\exists$ steps be labeled a *bad* proof. We clearly need to permute the order of such quantifier steps in order to rid the given LLO proof of bad $\exists$ steps to construct an LL proof.

We will be constructing the LL proof by induction on the structure of the given LLO proof. The LL proof is constructed by applying permutations to the LL subproofs obtained from the induction hypothesis. We therefore need to keep track of which inference steps in the constructed LL proof remain permutable. This information is actually implicit in the original LLO proof. We therefore set up a correspondence between the steps in the LLO proof and those in the resulting LL proof. This many-to-many correspondence is preserved by the permutations. It is possible to obtain a strict partial ordering on the steps in the given LLO proof so that step $Q1 < Q2$ if and only if one of the following is true in the LLO proof:

1. $Q1$ occurs below $Q2$ and $Q2/Q1$ is an LL impermutability.
2. The principle formula of $Q2$ is a subformula of $Q1$.
3. $Q1$ is a $\forall$ step introducing Herbrand function h, and $Q2$ is an $\exists$ step introducing Herbrand variable v, where $V(v)$ contains h.
4. There is some $Q3$ such that $Q1 < Q3$ and $Q3 < Q2$.

We show that it is possible to construct an LL proof from a given LLO proof that is *consistent* with the above strict partial ordering so that any steps $Q1'$ and $Q2'$ in the LL proof corresponding to $Q1$ and $Q2$ in the LLO proof, $Q2'$ occurs below $Q1'$ whenever $Q1 < Q2$. Furthermore, if $Q1'$ and $Q2'$ in the LL proof are such that we have for the corresponding steps in the LLO proof that $Q1 \not< Q2$, then $Q1'$ is permutable above $Q2'$ in the LL proof. Note that an LL proof consistent with the strict partial ordering does not contain any bad $\exists$ steps.

In constructing the LL proof by induction on structure of the given LLO proof, we only examine the case when the concluding inference in the LLO proof is an $\exists$ step since the other cases are straightforward. We therefore suppose that we are given an LLO proof of the form

$$\frac{\begin{array}{c}\Pi\\ \overset{V}{U} \vdash_{R,S,T} A\{v/x\}, \Gamma\end{array}}{\overset{V}{U} \vdash_{R,S,T} (\exists x.A), \Gamma}\exists$$

where the concluding $\exists$ step is bad in that $V(v)$ contains occurrences of Herbrand functions introduced by $\forall$ steps in Π. We label the latter $\forall$ steps as *injured* by the bad $\exists$ step. By induction, we can construct an LL proof Π' from Π that is consistent with the strict partial ordering derived from the LLO proof and hence, does not contain any bad $\exists$ steps. Every proof step Q' in Π' (corresponding to Q in Π) that occurs above this bad $\exists$ step and below any corresponding injured $\forall$ step is either permutable below the bad $\exists$ step (a low step) or above the injured $\forall$ step (a high step), since otherwise, we would have a cycle in the LLO proof of the form $\forall < \exists < Q < \forall$ thus violating the strictness of the partial order. It is clear that the high steps must be permuted above the injured $\forall$ steps, and the low steps and the injured $\forall$ steps must be permuted below the bad $\exists$ step. In performing the above permutations, it should be noted that certain permutations introduce copies of subproofs. For example, the permutation of a $\otimes$ step above a & step copies one of the subproofs of the $\otimes$ step. For example, proofs of the form

$$\frac{\dfrac{\vdash A, C, \Sigma \quad \vdash A, D, \Sigma}{\vdash A, C \,\&\, D, \Sigma}\& \qquad \vdash B, \Delta}{\vdash A \otimes B, C \,\&\, D, \Sigma, \Delta}\otimes$$

can be permuted to the form

$$\frac{\dfrac{\vdash A, C, \Sigma \quad \vdash B, \Delta}{\vdash A \otimes B, C, \Sigma, \Delta}\otimes \qquad \dfrac{\vdash A, D, \Sigma \quad \vdash B, \Delta}{\vdash A \otimes B, D, \Sigma, \Delta}\otimes}{\vdash A \otimes B, C \,\&\, D, \Sigma, \Delta}\&$$

In LLG, the parameter U is given as an input substitution to the proof search and augmented at every $\exists$ step. We next introduce another variant LLV where we add another parameter, an *output* substitution V, so that a sequent in LLV has the form ${}^{V}_{U} \vdash_T ?$. The Id rule now has the form

$$\frac{}{{}^{V}_{U} \vdash_T \Pi, A, B^{\perp}} Id$$

provided V is a unifier for A and B under the substitution U. The binary rules for $\otimes$ and $\&$ are also modified so that a rule of the form $\frac{\vdash \Gamma_1 \quad \vdash \Gamma_2}{\vdash \Gamma}$ becomes

$$\frac{{}^{W}_{U} \vdash_T \Gamma_1 \qquad {}^{V}_{W} \vdash_T \Gamma_2}{{}^{V}_{U} \vdash_T \Gamma}.$$

The quantifier rules in LLV have the form:

$$\frac{{}^{V}_{U} \vdash_{T \cup \{v\}} A\{v/x\}, \Gamma}{{}^{V}_{U} \vdash_T (\exists x.A), \Gamma} \exists \qquad \frac{{}^{V}_{U} \vdash_T A\{h(T)/x\}, \Gamma}{{}^{V}_{U} \vdash_T (\forall x.A), \Gamma} \forall$$

where v and h are new. For a complete list of proof rules, see Appendix D.

Lemma 3.2 *If LLV proves* ${}^{V}_{U \circ U} \vdash_T ?$, *then it proves* ${}^{V}_{U} \vdash_T ?$.

Theorem 3.3

1. *If LLG proves* ${}_{U} \vdash_T ?$, *then for some* V, *LLV proves* ${}^{V}_{U} \vdash_T ?$.
2. *If LLV proves* ${}^{V}_{U} \vdash_T ?$, *then for some* W, *LLG proves* ${}_{W} \vdash_T ?$.

3.1 Complexity Results

Using LLV, the multiplicative-additive fragment of first-order linear logic (MALL1) can be shown to be decidable in NEXPTIME. This follows from the fact that any MALL1 proof has a depth that is bounded by the number of connectives in the conclusion. This means that the number of axiom nodes in such a proof is bounded by an exponential in the size of the conclusion. Since LLV introduces a Herbrand term of the form $h(T)$ for each universal quantifier in the conclusion, where T is itself bounded by the number of existential quantifiers in the conclusion, each axiom sequent is at most quadratic in the size of conclusion. Since unification is linear [14], one can build a nondeterministic procedure to guess and check a proof in time that is exponential in the size of the conclusion. Lincoln and Scedrov [11] have shown that the decidability of MALL1 is NEXPTIME-hard so we now have a tight bound on the complexity.

Theorem 3.4 *MALL1 is* NEXPTIME*-complete.*

The multiplicative fragment of first-order linear logic (MLL1) can be shown to be decidable in NP by the same method. The number of axiom nodes in a MLL1 proof is linear in the size of the conclusion, and each axiom node in the search has size at most quadratic in the size of the conclusion. This results in a tight bound since the propositional fragment MLL is NP-complete [8].

Theorem 3.5 *MLL1 is* NP*-complete.*

4 An Optimization

We now present an optimization of LLV called LLO that exploits the permutabilities of linear logic. As is clear from the table in Appendix A, there are only a few pairs of impermutable rules in LL. A rule R1 is said to be impermutable below R2, that is R1/R2, if there is a ? such that the principal formulas of R1 and R2 occur as distinct formulas in ?, but $\vdash ?$ can only be proved with R2 below R1. Otherwise, R1 is said to be permutable below R2, or equivalently, R2 is permutable above R1. One way to exploit the permutabilities is to restrict the backtracking in proof search to those rules that might not permute below others [1]. As we have already seen, the permutabilities can also be used to reduce the dependencies between Herbrand variables and Herbrand functions. For any impermutability R1/R2, if an R2 step occurs below an R1 step in the proof search, then the Herbrand variables governing the principal formula of the R2 step should also govern the principal formula of the R1 step. A Herbrand variable introduced in place of the existential quantifier in $(\exists x.A)$ also governs all subformulas of A in the proof. A Herbrand variable *governs* a formula if it dominates all the Herbrand functions replacing universal quantifiers in subformulas of the given formula in the proof.

In LLV, a universal quantifier is replaced by an application of a new Herbrand function to all the Herbrand variables that happen to have been introduced at that point in the proof. The point of the optimized system LLO is to reduce the set of Herbrand variables used in this construction to only those Herbrand variables which necessarily govern the universal formula. In a formula like $(\exists x.P(x)) \parr (\forall y.P^{\perp}(y))$ a possible LLO proof proceeds by reducing $\parr$, then $\exists$, and then $\forall$. This proof is not acceptable in LLV, as the Herbrand variable replacing x would appear in the term replacing y, causing occur-check failure of unification. However, LLO is sensitive to the fact that the Herbrand variable replacing x does not govern the universal term. The key property is that the LLO proof *could* have been carried out in such a way that the existential formula was reduced above the universal one. Another way to view this is that the LLV impermutability $\exists/\forall$ has been eliminated in LLO.

This optimization is important because proof search may succeed more quickly on true theorems as there are fewer failures of unification, and thus it is more likely in nondeterministic proof search, to find a successful proof. For example, in the pure multiplicative fragment, all quantifier rules can be applied immediately. This is in contrast to LLV where one must try all possible orderings of the quantifier rules.

One way to track the dependencies between impermutable rules is to maintain a set of Herbrand variables corresponding to every pair of impermutable rules. Since only three rules $!S$, $\parr$, and $\&$ occur in the R2 position of any LL impermutabilities of the form R1/R2, where $R1$ is not $!S$, it is sufficient to maintain three sets of variables R, S, and T, and treat $!S$ specially. A sequent in LLO therefore consists of the two substitutions U and V, and three sets of variables: R, the set of $!S$ Herbrand variables, S, the $\parr$ Herbrand

$$\dfrac{\dfrac{}{\vdash p^{\perp}, p}Id \qquad \dfrac{\dfrac{\{v \leftarrow c\}}{\vdash q^{\perp}(v), q(c)}}{\vdash (\exists x.q^{\perp}(x)), q(c)}\exists}{\dfrac{\vdash p^{\perp}, (\exists x.q^{\perp}(x)), q(c) \otimes p}{\dfrac{\vdash p^{\perp}, (\exists x.q^{\perp}(x)), (\forall y.q(y) \otimes p)}{\vdash p^{\perp} \parr (\exists x.q^{\perp}(x)), (\forall y.q(y) \otimes p)}\parr}\forall}\otimes$$

The above form of dynamic Skolemization is still inefficient since it is not flexible relative to the permutabilities of the sequent calculus. For instance, it permits a similar proof search (shown below) with a slightly different order of the rules to fail.

$$\dfrac{\dfrac{}{\vdash p^{\perp}, p}Id \qquad \dfrac{\text{unify fails}}{\vdash q^{\perp}(v), q(h(v))}}{\dfrac{\vdash p^{\perp}, q^{\perp}(v), q(h(v)) \otimes p}{\dfrac{\vdash p^{\perp}, q^{\perp}(v), (\forall y.q(y) \otimes p)}{\dfrac{\vdash p^{\perp}, (\exists x.q^{\perp}(x)), (\forall y.q(y) \otimes p)}{\vdash p^{\perp} \parr (\exists x.q^{\perp}(x)), (\forall y.q(y) \otimes p)}\parr}\exists}\forall}\otimes$$

By paying attention to the permutabilities, it is possible to obtain an optimized form of dynamic Skolemization where even the latter order of proof search can succeed. In this case, the $\exists$ rule is clearly permutable above the $\forall$ rule so that the Herbrand function introduced in place of the $\forall$ quantifier need not depend on the Herbrand variable replacing the $\exists$ quantifier. The resulting "proof" has the form:

$$\dfrac{\dfrac{}{\vdash p^{\perp}, p}Id \qquad \dfrac{\{v \leftarrow c\}}{\vdash q^{\perp}(v), q(c)}}{\dfrac{\vdash p^{\perp}, q^{\perp}(v), q(c) \otimes p}{\dfrac{\vdash p^{\perp}, q^{\perp}(v)), (\forall y.q(y) \otimes p)}{\dfrac{\vdash p^{\perp}, (\exists x.q^{\perp}(x)), (\forall y.q(y) \otimes p)}{\vdash p^{\perp} \parr (\exists x.q^{\perp}(x)), (\forall y.q(y) \otimes p)}\parr}\exists}\forall}\otimes$$

The point here is that even though the unifier is such that the proof search violates the eigenvariable condition, it is possible to permute the steps so as to obtain a valid proof. By thus allowing more of the search paths to succeed, it is possible to reduce the need for backtracking in proof search. We can also check that the same optimized procedure applied to $\vdash (\exists x.p^{\perp} \parr q^{\perp}(x)), (\forall y.q(y)) \otimes p$ does fail.

$$\dfrac{\dfrac{}{\vdash p^{\perp}, p}Id \qquad \dfrac{\dfrac{\text{unify fails}}{\vdash q^{\perp}(v), q(h(v))}}{\vdash q^{\perp}(v), (\forall y.q(y))}\forall}{\dfrac{\vdash p^{\perp}, q^{\perp}(v), (\forall y.q(y)) \otimes p}{\dfrac{\vdash p^{\perp} \parr q^{\perp}(v), (\forall y.q(y)) \otimes p}{\vdash (\exists x.p^{\perp} \parr q^{\perp}(x)), (\forall y.q(y)) \otimes p}\exists}\parr}\otimes$$

Here, since the $\exists$ quantifier governs the $\parr$ which is not permutable above the $\otimes$ governing the $\forall$, the Herbrand function introduced for the $\forall$ is dominated by the Herbrand variable replacing the $\exists$.

The completeness of both dynamic Skolemization and its optimization are straightforward since if there is a proof of the given conjecture, then both search procedures can succeed by nondeterministically applying the inference rules in the same order as the given proof. The soundness of the optimization is somewhat more delicate since it succeeds even when the order of inference rules in the search could violate the eigenvariable condition. We present these search procedures and demonstrate their soundness and completeness. For concreteness, these theorems will be stated in terms of linear logic, but their generalization to other cut-free sequent calculi will be apparent from the proofs. We also obtain complexity results for some first-order fragments of linear logic.

3 Linear Logic

We will be assuming the basic terminology of sequent calculus, linear logic, and unification theory [3, 5]. We will only be dealing with idempotent substitutions U so that $U(U(t)) = U(t)$, for any t. The composition of substitutions will be such that $(U \circ V)(t) = U(V(t))$, for any t. A substitution U is less general than V, in symbols $U < V$, if and only if there exists a W such that $U = W \circ V$. As our starting point, we use a small variant of linear logic shown in Appendix B where the weakening rule is built into the axiom rule, and the contraction rule is built into the dereliction and $\otimes$ rules. It is easy to see that the modified system is equivalent to the more conventional presentation of linear logic.

We then modify LL slightly to obtain LLG where each sequent additionally contains a list of free variables T and a substitution U, and the quantifier and Identity rules are modified. The $\exists$ and $\forall$ rules have the form

$$\dfrac{{}_{\{t/v\} \circ U} \vdash_{T \cup \{v\}} A\{v/x\}, \Gamma}{{}_{U} \vdash_{T} (\exists x.A), \Gamma}\exists \qquad \dfrac{{}_{U} \vdash_{T} A\{h(T)/x\}, \Gamma}{{}_{U} \vdash_{T} (\forall x.A), \Gamma}\forall$$

where v is a new[2] Herbrand variable, h is a new Herbrand function, and if $T = \{v_1, \ldots, v_n\}$, then $h(T)$ is $h(v_1, \ldots, v_n)$ so that each v_i in T dominates h. The Id rule is also modified as

$$\dfrac{U(A) \equiv U(B)}{{}_{U} \vdash_{T} \Gamma, A, B^{\perp}}Id$$

where A and B are atomic. The other rules remain unchanged and preserve the parameters U and T from conclusion to premises. For a complete list of proof rules, see Appendix C. The system LLG is introduced only as a step leading to LLV and LLO, which are the systems of real interest.

Let the eigenvariables in LL be labeled $a_1, \ldots, a_n, \ldots$ and the Herbrand functions in LLG be $h_1, \ldots, h_n, \ldots$. The notation $\underline{?}$ denotes the result of replacing every term of the form $h_i(\ldots)$ in $?$ by the corresponding a_i.

Lemma 3.1 *If T contains all the free Herbrand variables in $?$, then LL proves $\vdash \underline{U(?)}$ if and only if LLG proves ${}_{U} \vdash_{T} ?$.*

[2] By a *new* Herbrand variable or Herbrand function, we mean one that does not occur in the input substitution U, the governing Herbrand variables T, or in any of the formulas in the conclusion sequent of the rule.

search. The general ideas are illustrated with the specific case of linear logic because, as is well known, it is possible to embed many logics into linear logic while preserving the structure of proofs [5, 12]. We present a series of proof search procedures for linear logic culminating in one where the permutability properties of linear logic are exploited to introduce an optimized form of *dynamic Skolemization*[1] that significantly reduces the amount of nondeterminism in the proof search. The optimization of the quantifier steps in the proof can be very effectively combined with other optimizations that are also based on permutability. We also present the performance results of a Prolog implementation of these search procedures.

The sequent proof search paradigm was previously used to analyze various fragments of propositional linear logic to show the undecidability of propositional linear logic, the PSPACE-completeness of multiplicative-additive linear logic (MALL), and the NP-completeness of multiplicative linear logic (MLL) with unrestricted weakening [10]. Using similar techniques, Kanovich showed that multiplicative linear logic is NP-complete [8]. Lincoln and Winkler demonstrated NP-completeness of MLL using only the constants [13]. Lincoln and Scedrov had previously shown this fragment of linear logic to be NEXPTIME-hard [11]. By extending this proof search paradigm, we are able to show here that the first-order version of MALL is decidable in NEXPTIME. We also observe that first-order MLL is NP-complete and hence of the same complexity as its propositional counterpart.

The naive form of dynamic Skolemization has been studied by a number of authors. Cerrito [2] presents a related system for linear logic. Shankar [15] presents an optimized form of dynamic Skolemization for the intuitionistic sequent calculus LJ. In this paper, we generalize this technique to arbitrary cut-free sequent calculi with the subformula property and conventional quantifier rules. We also provide rigorous proofs of the soundness and completeness of this procedure for the specific case of linear logic. These proofs can easily be adapted for other cut-free sequent calculi. Finally, we present experimental evidence that the proof systems discussed here directly lead to efficient proof search procedures for linear logic.

2 Proof Search Examples from Linear Logic

Using the proof rules for linear logic given in Appendix B and the standard definition of linear negation, we can illustrate the relevant ideas underlying sequent proof search. A specific instance of a proof rule is called a *step*. First consider the example of the linear sequent $\vdash (p^\perp \parr q^\perp) \parr (q \otimes p)$. This sequent has only one possible cut-free proof, namely

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash p^\perp, p}Id \qquad \dfrac{}{\vdash q^\perp, q}Id}{\vdash p^\perp, q^\perp, q \otimes p}\otimes}{\vdash p^\perp \parr q^\perp, q \otimes p}\parr}{\vdash (p^\perp \parr q^\perp) \parr (q \otimes p)}\parr$$

By the subformula property of cut-free proofs, the $\parr$ step must be the lowest one in the proof. For the next step of the search, we have a choice between the $\parr$ rule and the $\otimes$ rule. It is easy to see that an application of the $\otimes$ rule at this point causes the proof search to fail, whereas the search succeeds with the application of the $\parr$ rule below the $\otimes$ rule. This shows that the $\otimes$ step cannot always be permuted below the $\parr$ step.

Next, consider a first-order variant of the above example that is not a theorem:
$\vdash (\exists x.q^\perp(x) \parr p^\perp), (\forall y.q(y)) \otimes p$. We have already seen that in any proof of such a formula, the $\parr$ step must occur below the $\otimes$ step. By the subformula property, the $\exists$ step must occur below the $\parr$ step and the $\forall$ step must occur above the $\otimes$ step. The only likely proof is the following:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash p^\perp, p}Id \qquad \dfrac{\dfrac{}{\vdash q^\perp(c), q(c)}Id}{\vdash q^\perp(c), (\forall y.q(y))}\forall}{\vdash p^\perp, q^\perp(c), (\forall y.q(y)) \otimes p}\otimes}{\vdash p^\perp \parr q^\perp(c), (\forall y.q(y)) \otimes p}\parr}{\vdash (\exists x.p^\perp \parr q^\perp(x)), (\forall y.q(y)) \otimes p}\exists$$

This proof clearly violates the eigenvariable condition because the parameter generalized in the $\forall$ step appears free in the conclusion of the step. Note that static Skolemization on the conclusion of this proof causes the proof search to (erroneously) succeed. The Skolemized form of the conclusion is $\vdash p^\perp \parr q^\perp(v), q(c) \otimes p$, which is easily proved in propositional linear logic when c instantiates v. With the naive form of dynamic Skolemization where quantifiers are replaced by Herbrand variables or functions *during* the proof search, the proof search rightly fails, as shown below.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash p^\perp, p}Id \qquad \dfrac{\overset{\text{unify fails}}{\overline{\vdash q^\perp(v), q(h(v))}}}{\vdash q^\perp(v), (\forall y.q(y))}\forall}{\vdash p^\perp, q(v)^\perp, (\forall y.q(y)) \otimes p}\otimes}{\vdash p^\perp \parr q^\perp(v), (\forall y.q(y)) \otimes p}\parr}{\vdash (\exists x.p^\perp \parr q^\perp(x)), (\forall y.q(y)) \otimes p}\exists$$

Note that the Herbrand variable v *dominates* the Herbrand function h since v is introduced below h in the search. The occurs check at the unification fails indicating a potential transgression of the eigenvariable condition. We can also successfully apply dynamic Skolemization to the positive example, $\vdash p^\perp \parr (\exists x.q^\perp(x)), (\forall y.q(y) \otimes p)$.

[1] The use of Herbrand functions in proof search is colloquially called Skolemization.

Proof Search in First-order Linear Logic and Other Cut-free Sequent Calculi

Reprint from: Ninth Annual IEEE Symposium on Logic in Computer Science (LICS'94) to be held in Paris, France, July 4-8 1994.
Also appears in SRI Technical Report CSL-93-11.

P. D. Lincoln*
lincoln@csl.sri.com

N. Shankar*
shankar@csl.sri.com

SRI International Computer Science Laboratory
333 Ravenswood Ave
Menlo Park CA 94025 USA

Abstract

We present a general framework for proof search in first-order cut-free sequent calculi and apply it to the specific case of linear logic. In this framework, Herbrand functions are used to encode universal quantification, and unification is used to instantiate existential quantifiers so that the eigenvariable conditions are respected. We present an optimization of this procedure that exploits the permutabilities of the subject logic. We prove the soundness and completeness of several related proof search procedures. This proof search framework is used to show that provability for first-order MALL is in NEXPTIME, and first-order MLL is in NP. Performance comparisons based on Prolog implementations of the procedures are also given. The optimization of the quantifier steps in proof search can be combined effectively with a number of other optimizations that are also based on permutability.

1 Introduction

Since proofs contain more information than the theorems they prove, the main challenge for automated reasoning is one of automatically finding proofs rather than merely proving theorems. We are therefore more interested in automated proof search than in automated theorem proving. Simply stated, the problem of proof search is to find a proof for a given conjecture.

There is a very simple but impractical approach to proof search: guess a formal proof of the given conjecture and check whether it is a correct proof. This approach is too nondeterministic. For example, it leaves open the choice of the formalism in which the proof is to be generated. We restrict our attention to sequent calculi since most logics can be formalized this way, and sequent calculi have certain structural advantages from the point of view of mechanization. We concentrate on *cut-free* sequent calculi so as to remove any guesswork regarding the choice of cut formulas. Most sequent calculi do exhibit cut-elimination theorems so that no theorems are lost by this restriction, even though some useful and compact proofs are typically ruled out. The latter disadvantage is outweighed by the ease with which cut-free sequent calculi can be mechanized.

The next level of nondeterminism is in the guessing of the term instantiating an existential quantifier. By employing *Herbrand functions* and unification, this guesswork can be eliminated. The Herbrand theorem [6] states that a formula A of first-order logic can be transformed into a quantifier-free formula A_H such that A is provable if and only if some finite disjunction of instances of A_H can be proved in classical propositional logic. The Herbrand theorem yields an effective semi-decision procedure for classical first-order logic, but cannot be applied directly to most logics, including intuitionistic, linear, and modal logics, which typically lack the transformations needed to convert A to A_H. The proof search procedures described in this paper eliminate the guesswork in the quantifier rules even in the absence of a Herbrand theorem.

The final level of nondeterminism in proof search has to do with the order in which the inference rules are applied. It is possible for the proof search to succeed with one ordering of the rules, and fail with some other ordering. A proof search procedure might therefore backtrack in order to try all possible orderings of the applicable proof rules. The relative ordering of the quantifier rules in a proof is particularly important because the rule of universal generalization must obey the *eigenvariable* condition. In many sequent calculi, the relative order of the structural and propositional rules is also crucial to the success of an attempted proof search. The permutability theorems [9] of a sequent calculus indicate when the relative order of two inference rules can be permuted without invalidating a proof. These permutabilities can be exploited to reduce the nondeterminism in the ordering of proof rules.

This paper studies the optimizations that can be applied to proof search in the general setting of cut-free sequent calculi. The point of these optimizations is to reduce the amount of backtracking in proof

*Work supported under NSF Grant CCR-9224858