Cut-elimination and a permutation-free sequent calculus for intuitionistic logic.

Roy Dyckhoff[$] & Luis Pinto[†]

School of Mathematical & Computational Sciences,
St Andrews University, St Andrews, Scotland

{rd, luis}@dcs.st-and.ac.uk

Abstract. We describe a sequent calculus, based on work of Herbelin, of which the cut-free derivations are in 1-1 correspondence with the normal natural deduction proofs of intuitionistic logic. We present a simple proof of Herbelin's strong cut-elimination theorem for the calculus, using the recursive path ordering theorem of Dershowitz.

Keywords. Cut-elimination, normalisation, natural deduction, intuitionistic logic, recursive path ordering, termination.

1. Introduction

Herbelin introduced [16, 17] an elegant sequent calculus **LJT** and proved for it a strong cut-elimination theorem by Dragalin's method [8], using structural induction on the associated proof-terms supported by inductions on measures of the strong normalisability. The proof is complex: there is more than one cut rule to consider. The calculus is of special interest because its cut-free derivations are in natural 1-1 correspondence with the dp-normal [28] natural deduction proofs of first-order intuitionistic logic.

The main purpose, and the novelty, of the present paper is to illustrate the use of the recursive path ordering (r.p.o.) theorem of Dershowitz [7] by giving a simple proof of Herbelin's cut-elimination theorem. We begin with a routine reformulation of the calculus in our own notation (developed as a basis for our work [12] on the analysis of permutations in **LJ**); this is detailed elsewhere in [11], which includes both a minor simplification of the Dragalin-style proof and the r.p.o. proof in more detail.

Our subsidiary purpose is to draw attention to Herbelin's calculus as a good alternative to the traditional formalisations of natural deduction, such as typed lambda calculus and the proof system $\mathbf{NJ}^{\mathbf{cut}}$. Being close to the formalisations (with a "stoup" formula [14]) used [23] in logic programming, Herbelin's calculus is attractive as a basis for proof search (being a "sequent calculus" but avoiding the problems in **LJ** arising from the permutations [19, 12]). It underlies, for example, the implementation of uniform proof search in

$ Both authors were supported by the European Commission via the ESPRIT Basic Research Action 7232 "GENTZEN"; the second author was supported by the JNICT (Portugal).

† New address: Departamento de Matemática, Universidade do Minho, Braga, Portugal.

hereditary Harrop logic [22]. It may also be used as a basis for some inductive proofs about derivations in **LJ** or natural deductions in **NJ** (e.g. those in [12], where the strong eliminability of cut, rather than just the admissibility, is required; and, we anticipate, those in [5] and [15]).

Herbelin called (following [6]) his calculus "**LJT**", a name we avoid in case of confusion with that in the first author's [9]: we call its cut-free fragment "**MJ**" because it is intermediate between [13] Gentzen's cut-free **LJ** and **NJ**. We apologise to Herbelin for not adopting his nomenclature. We use "$\mathbf{MJ^{cut}}$" for the extension of **MJ** with cut rules; similarly, we use $\mathbf{NJ^{cut}}$ for the usual natural deduction calculus and **NJ** for the fragment consisiting only of normal deductions. As in [17], we cover here only the implicational case; other connectives (dealt with in [16, 11]) pose no significant extra difficulty.

Herbelin's cut-free calculus is here called "permutation-free", because there are no semantically trivial permutations of the inference rules, where by "semantically trivial" we mean "interpreted in **NJ** as true equations between two (normal) deductions". In contrast, the interpretation [24] of **LJ** into **NJ** is many-one, because of the permutations [19] in **LJ**. (Note however that there are some permutations in **LJ,** involving disjunction or the existential quantifier, whose interpretations in **NJ** are false. For details see [12].)

We refer to [29] for basic proof theory and detailed descriptions (§3.3) of the relationships between systems such as **LJ** and **NJ.** The system we call **LJ** is roughly the cut-free Gentzen-Kleene system **GK3i** of [29], p 70. We also refer to [29] (§1.3.5) for the Curry-Howard correspondence between typed lambda terms and natural deduction proofs: this correspondence allows us to alternate between the type-theoretic view (based on lambda calculus) and the proof-theoretic view of natural deduction. We follow [29] in first naming the cut-free fragments of various sequent calculi and then designating the extensions with cut by a superscript; **NJ** is treated similarly.

2. Herbelin's calculus (in the cut-free case)

Consider first a routine (but rarely written down) description of the normal terms of the untyped lambda calculus:

$$A \ ::= \ ap(A,N) \mid var(V)$$

$$N \ ::= \ \lambda V.N \mid an(A)$$

where $\boldsymbol{V}$ is some set of variables, $\boldsymbol{N}$ is the set of *normal* terms and $\boldsymbol{A}$ is the set of *normal non-abstraction* terms. Variable binding conventions are, as usual, that, in $\lambda V.N$, λV binds free occurrences of V in N. We use explicit constructors var and an to ensure consistency with our type-checked implementations. This description restricts the terms N so constructed to be normal in the traditional sense, because the first argument A in $ap(A,N)$ cannot be an abstraction $\lambda V.N'$. Another routine description $N ::= \lambda\vec{x}.x\vec{N}$ or, equivalently, $N ::= \lambda x.N \mid x\vec{N}$, of the normal lambda terms leads to essentially the same ideas.

The normal terms are thus of the form

$$\lambda x_1.\lambda x_2 \ldots \lambda x_n.an(ap(\ldots ap(var(x), N_1), \ldots, N_m))$$

in which x is called the *head variable*.. The head variable of such a term N is, for a large term, buried deep inside: Herbelin's representation brings it to the surface. So, following ideas in [17], we make the

Definition. The set $\boldsymbol{M}$ of *untyped deduction terms* and the set $\boldsymbol{Ms}$ of "lists" of such terms are defined simultaneously as follows:

$$\begin{aligned} M &::= (V;Ms) \mid \lambda V.M \\ Ms &::= [] \mid M::Ms \end{aligned}$$

We use again the same symbol λ where we should really use another symbol. When other connectives are added, Ms will no longer be a list. $[M]$ abbreviates $M::[]$, and so on. Variable binding conventions are as before. Terms are *equal* iff they are alpha-convertible. *Closed* terms are those with no free variable occurrences. These terms M are the "normal $\bar{\lambda}$-expressions" of [16, 17] in a minor variant of the notation.

Adding type restrictions in the usual way gives us a description of the *typable* terms in contexts, where *contexts* Γ associate formulae (i.e. types) P to variables x. We use both the judgment form $\Gamma \Rightarrow M : P$ (read as "M is a term of type P in the context Γ") and the judgment form $\Gamma \xrightarrow[P]{} Ms : Q$ (read as "Ms is a term-list based on P of type Q in the context Γ"). The idea here is that Ms is a list of terms representing minor premisses of implication elimination rule instances on the main (introduction-free) branch [24] from the head formula P down to Q; and the assumptions on which these minor premisses depend are all declared in Γ. (The *head formula* of a normal deduction is that occurring at the top left of the main branch, i.e. the type of the head variable.)

The schematic rules for deriving judgments of this kind are (in the implicational fragment) as follows, where the *Abstract* rule has an implicit side-condition about "newness" of the variable:

$$\frac{\Gamma, x{:}P \xrightarrow[P]{} Ms : R}{\Gamma, x{:}P \Rightarrow (x;Ms) : R}\,Select \qquad \frac{}{\Gamma \xrightarrow[P]{} [\,] : P}\,Meet$$

$$\frac{\Gamma, x{:}P \Rightarrow M : Q}{\Gamma \Rightarrow \lambda x.M : P \supset Q}\,Abstract \qquad \frac{\Gamma \Rightarrow M : P \quad \Gamma \xrightarrow[Q]{} Ms : R}{\Gamma \xrightarrow[P \supset Q]{} (M{::}Ms) : R}\,Split$$

There is a bijective translation between $\boldsymbol{M}$ and $\boldsymbol{N}$, mentioned but not detailed in [17]: briefly, $(x;[M_1,\ldots,M_n])$ translates into the normal term $ap(\ldots ap(x,M_1),\ldots,M_n)$, usually written as $xM_1\ldots M_n$, and abstraction terms translate in the obvious way. Formally, $\theta : \boldsymbol{M} \to \boldsymbol{N}$ and $\psi : \boldsymbol{N} \to \boldsymbol{M}$ may be defined as follows:

$$\begin{array}{l}
\theta : \boldsymbol{M} \to \boldsymbol{N} \\
\theta(x;Ms) =_{def} \theta'(var(x), Ms) \\
\theta(\lambda x.M) =_{def} \lambda x.(\theta M) \\
\theta' : \boldsymbol{A} \times \boldsymbol{Ms} \to \boldsymbol{N} \\
\theta'(A,[\,]) =_{def} an(A) \\
\theta'(A, M{::}Ms) =_{def} \theta'(ap(A,\theta M), Ms)
\end{array}
\qquad
\begin{array}{l}
\psi : \boldsymbol{N} \to \boldsymbol{M} \\
\psi(an(A)) =_{def} \psi'(A,[\,]) \\
\psi(\lambda x.N) =_{def} \lambda x.\psi N \\
\psi' : \boldsymbol{A} \times \boldsymbol{Ms} \to \boldsymbol{M} \\
\psi'(var(x), Ms) =_{def} (x;Ms) \\
\psi'(ap(A,N), Ms) =_{def} \psi'(A, \psi N{::}Ms)
\end{array}$$

Proposition 1. (i) $\psi \circ \theta = id_M : \boldsymbol{M} \to \boldsymbol{M}$.
(ii) $\psi \circ \theta' = \psi' : \boldsymbol{A} \times \boldsymbol{Ms} \to \boldsymbol{M}$.

Proof. By simultaneous induction on the structures of the argument and the second argument respectively. **QED**.

Proposition 2. (i) $\theta \circ \psi = id_N : \boldsymbol{N} \to \boldsymbol{N}$.
(ii) $\theta \circ \psi' = \theta' : \boldsymbol{A} \times \boldsymbol{Ms} \to \boldsymbol{N}$.

Proof. By simultaneous induction on the structures of the argument and the first argument respectively. **QED**.

It follows that $\boldsymbol{M}$ and $\boldsymbol{N}$ are in 1-1 correspondence. We must however check that the correspondences θ and ψ work well at the typed level. Here is an appropriate proof system for the typed version **NJ** of $\boldsymbol{N}$:

$$\frac{\Gamma \rhd A : P}{\Gamma \rhd\!\rhd an(A) : P} \qquad \frac{}{x{:}P, \Gamma \rhd var(x) : P}\,ax.$$

$$\frac{x{:}P, \Gamma \rhd\!\rhd N : Q}{\Gamma \rhd\!\rhd \lambda x.N : P \supset Q}\supset\!\mathrm{I} \qquad \frac{\Gamma \rhd A{:}P \supset Q \quad \Gamma \rhd\!\rhd N : P}{\Gamma \rhd ap(A,N) : Q}\supset\!\mathrm{E}$$

Note again that we are considering just the fragment of **NJ** which allows only normal terms, or (equivalently) the normal fragment of the simply typed lambda calculus.

Proposition 3. The following rules are admissible:

$$\text{(i)}\quad \frac{\Gamma \Rightarrow M : R}{\Gamma \rhd\!\rhd\, \theta M : R} \qquad\qquad \text{(ii)}\quad \frac{\Gamma \rhd A : P \quad \Gamma \xrightarrow[P]{} Ms : R}{\Gamma \rhd\!\rhd\, \theta'(A, Ms) : R}$$

Proof. By simultaneous induction on the structures of M and Ms respectively. For example, in the proof of (ii), in the case $Ms = (M{::}Mss)$, the second premiss must be the conclusion of a *Split* rule, with P of the form $P' \supset Q'$, with premisses $\Gamma \Rightarrow M : P'$ and $\Gamma \xrightarrow[Q']{} Mss : R$. We can now build the proof

$$\dfrac{\dfrac{\Gamma \rhd A{:}P \quad \dfrac{\Gamma \Rightarrow M : P'}{\Gamma \rhd\!\rhd\, \theta M : P'}(i)}{\Gamma \rhd ap(A, \theta M) : Q'}\supset E \quad \Gamma \xrightarrow[Q']{} Mss : R}{\Gamma \rhd\!\rhd\, \theta'(ap(A, \theta M), Mss) : R}(ii)$$

where *(i)* refers to an inductive use of (i), M being a substructure of Ms, and *(ii)* refers to an inductive use of (ii), Mss being a substructure of Ms. From this, using the definition of θ', we conclude that $\Gamma \rhd\!\rhd\, \theta'(A, M{::}Mss) : R$. **QED**.

Proposition 4. The following rules are admissible:

$$\text{(i)}\quad \frac{\Gamma \rhd\!\rhd\, N : R}{\Gamma \Rightarrow \psi N : R} \qquad\qquad \text{(ii)}\quad \frac{\Gamma \rhd A : P \quad \Gamma \xrightarrow[P]{} Ms : R}{\Gamma \Rightarrow \psi'(A, Ms) : R}$$

Proof. By simultaneous induction on the structures of N and A respectively. **QED**.

Using the Curry-Howard correspondence between [normal] terms of the simply typed lambda calculus and the [normal] natural deductions of intuitionistic implicational logic, we thus have a 1-1 correspondence between the typed terms M and the normal natural deductions N of the same logic. There are well-known problems [20] of ensuring that the Curry-Howard correspondence is 1–1; so we use assumption classes in our natural deductions, and then the correspondence is (for closed terms and assumption-free proofs) 1–1 modulo α-convertibility (more generally, it is 1–1 modulo choice of variable names).

More precisely, we are considering a version of natural deduction where *contexts* Γ are multisets of formulae, *judgments* are sequents $\Gamma \Rightarrow P$ and *deductions* are trees; *assumption discharge* is achieved by discharging at most one assumption class, i.e. an occurrence of the assumption in the context. Deductions are *normal* if they contain no introduction step immediately followed by an elimination. Search for the normal natural deductions of a sequent $\Gamma \Rightarrow P$ is thus transformed by the above results into the problem of searching in the calculus **MJ**, which, in contrast to **NJ**, has (like **LJ**) the immediate sub-formula property and, in contrast to **LJ**, has its derivations in 1-1 correspondence with the (normal) deductions of **NJ**.

3. Termination of the cut reduction rules

We now consider the syntax of **MJ** extended to $\mathbf{MJ^{cut}}$ by allowing constructors for terms representing derivations using a cut rule. Since there are two kinds of sequent, there are several (in fact, four) cut rules. For convenience in proving cut-elimination, the constructors cut_i have an extra argument, the cut formula. The context-free syntax is given by adding the productions

$$Ms ::= cut_1(P, Ms, Mss) \mid cut_2(P, M, V.Ms)$$
$$M ::= cut_3(P, M, Ms) \mid cut_4(P, M, V.M)$$

and the typed syntax is as follows:

$$\frac{\Gamma \xrightarrow[Q]{} Ms : P \quad \Gamma \xrightarrow[P]{} Mss : R}{\Gamma \xrightarrow[Q]{} cut_1(P, Ms, Mss) : R}\,Cut_1$$

$$\frac{\Gamma \Rightarrow M : P \quad \Gamma \xrightarrow[P]{} Ms : R}{\Gamma \Rightarrow cut_3(P, M, Ms) : R}\,Cut_3$$
$$\frac{\Gamma \Rightarrow M : P \quad \Gamma, x{:}P \xrightarrow[Q]{} Ms : R}{\Gamma \xrightarrow[Q]{} cut_2(P, M, x.Ms) : R}\,Cut_2$$

.

$$\frac{\Gamma \Rightarrow M : P \quad \Gamma, x{:}P \Rightarrow M' : R}{\Gamma \Rightarrow cut_4(P, M, x.M') : R}\,Cut_4$$

The theorem (attributed by Herbelin to Coquand) asserting the admissibility of these rules is a weak cut-elimination theorem: more powerfully one can prove [16, 17] by Dragalin's method the (strong) termination, i.e. that every reduction sequence is finite, of the following complete set of reduction rules:

$$cut_1(P, [], Mss) \rightarrow Mss$$

$$cut_1(P, M{::}Ms, Mss) \rightarrow M{::}cut_1(P, Ms, Mss)$$

$$cut_2(P,M,x.[]) \to []$$
$$cut_2(P,M,x.(M'::Ms)) \to cut_4(P,M,x.M')::cut_2(P,M,x.Ms)$$

$$cut_3(P,(x;Ms),Mss) \to (x;cut_1(P,Ms,Mss))$$
$$cut_3(S \supset T,\lambda y.M,[]) \to \lambda y.M$$
$$cut_3(S \supset T,\lambda y.M,M'::Ms) \to cut_3(T,cut_4(S,M',y.M),Ms)$$

$$cut_4(P,M,x.(y;Ms)) \to (y;cut_2(P,M,x.Ms)) \quad (y \neq x)$$
$$cut_4(P,M,x.(x;Ms)) \to cut_3(P,M,cut_2(P,M,x.Ms))$$
$$cut_4(P,M,x.(\lambda y.M')) \to \lambda y.cut_4(P,M,x.M')$$

The *completeness* of this set of rules is simply the fact, obvious by inspection, that every term beginning with a cut (and whose sub-terms are cut-free) matches at least one left-hand side. So, each irreducible term is cut-free. One also needs to show for each reduction rule $L \to R$ that the appropriate one of the two inference schemata

$$\frac{\Gamma \xrightarrow[P]{} L : Q}{\Gamma \xrightarrow[P]{} R : Q} \quad \frac{\Gamma \Rightarrow L : Q}{\Gamma \Rightarrow R : Q}$$

is admissible: this is routine. In fact, one can show more: w.r.t. the "obvious" interpretation Θ of the terms of the extended calculus $\mathbf{MJ^{cut}}$ into $\mathbf{NJ^{cut}}$, $\Theta(L) = \Theta(R)$ or (just the seventh rule) $\Theta(L) \to_\beta \Theta(R)$, where $\to_\beta$ is the usual β-reduction relation. [11] gives details of these two arguments in our own notation.

Our own proof of this termination result depends on the recursive path ordering (r.p.o.) theorem of Dershowitz [7]. Let $>$ be a transitive and irreflexive ordering on a set F of operators, and $T(F)$ be the set of closed terms over F. Then $>_{rpo}$ is defined recursively on $T(F)$ by

$$s = f(s_1,\ldots,s_m) >_{rpo} g(t_1,\ldots,t_n) = t$$

iff

$$s_i \geq_{rpo} t \quad \text{for some } i = 1,\ldots,m$$

or

$$f > g \text{ and } s >_{rpo} t_j \text{ for all } j = 1,\ldots,n$$

or

$$f = g \text{ and } \{s_1,\ldots,s_m\} >>_{rpo} \{t_1,\ldots,t_n\}$$

where $>>_{rpo}$ is the extension of $>_{rpo}$ to finite multisets and $\geq_{rpo}$ means $>_{rpo}$ or equivalent up to permutations of subterms. The r.p.o. theorem says that if $>$ is well-founded, then so is $>_{rpo}$.

We treat the term $cut_1(P, Ms, Mss)$ as if made up of an operator $cut_1(P)$ and two arguments Ms and Mss; similarly for the other cut terms. The operators are then ordered according to the following rules:

$$cut_i(P) > cut_j(Q) \text{ for } P > Q \text{ and } i, j = 1,2,3 \text{ or } 4$$

$$cut_4(P) > cut_3(P)$$
$$cut_4(P) > cut_1(P)$$
$$cut_2(P) > cut_3(P)$$
$$cut_2(P) > cut_1(P)$$

For the non-cut terms, we have the operators *';', 'λ', '::' and '[]'*, which just need to be ordered below each of the $cut_i(P)$ operators. The formulae P can be ordered by the sub-term relation. We thus have an ordered set $(Op, >)$ of operators

$$Op = \{\ cut_i(P) : i = 1, 2, 3 \text{ or } 4, P \text{ a formula}\ \} \cup \{\ ';',\ 'λ',\ '::',\ '[]'\ \}$$

Proposition 5. The ordering > on *Op* is transitive, irreflexive and well-founded.

Proof. Transitivity follows by examination of cases. Irreflexivity is trivial. The only possibility of an infinite decreasing sequence is of the form $cut_{i_0}(P_0) > cut_{i_1}(P_1) > \ldots$ whose length must be bounded by twice the depth of P_0 since each reduction either reduces the argument P or both fixes P and reduces the suffix of the cut from *4* or *2* to *3* or *1*. **QED**.

It follows from the r.p.o. theorem that $>_{rpo}$ on the set of closed typed terms of **MJ**$^{\mathbf{cut}}$ is well-founded. "Closed" in this context means containing no free metavariables.

Theorem. The set of cut-reduction rules of **MJ**$^{\mathbf{cut}}$ is strongly terminating.

Proof. We must check for each instance of a cut-reduction rule that the *LHS* $>_{rpo}$ *RHS*. Here we check just two of the rules to illustrate the technique:

(i) $cut_3(S \supset T, \lambda y.M, M'::Ms) >_{rpo} cut_3(T, cut_4(S, M', y.M), Ms)$

because $cut_3(S \supset T) > cut_3(T)$

because $S \supset T > T$

and

$cut_3(S \supset T, \lambda y.M, M'::Ms) >_{rpo} cut_4(S, M', y.M)$

because $cut_3(S \supset T) > cut_4(S)$

because $S \supset T > S$

and $cut_3(S \supset T, \lambda y.M, M'::Ms) >_{rpo} M'$

because $M':: Ms \geq_{rpo} Ms$

and $cut_3(S \supset T, \lambda y.M, M'::Ms) >_{rpo} M$

because $\lambda y.M \geq_{rpo} M$

and

$cut_3(S \supset T, \lambda y.M, M'::Ms) >_{rpo} Ms$.

because $M'::Ms \geq_{rpo} Ms$

(ii) $cut_4(P,M, x.(x;Ms)) >_{rpo} cut_3(P,M,cut_2(P,M, x.Ms))$

because $cut_4(P) > cut_3(P)$

and $cut_4(P,M, x.(x;Ms)) >_{rpo} M$

because $M \geq_{rpo} M$

and $cut_4(P,M, x.(x;Ms)) >_{rpo} cut_2(P,M, x.Ms)$

because $cut_4(P) = cut_2(P)$

and $\{M, (x;Ms)\} >>_{rpo} \{M, Ms\}$

because $(x;Ms) >_{rpo} Ms$.

There is a minor problem: $cut_4(P)$ and $cut_2(P)$ are not necessarily equal; indeed no order between them can be inferred from the above definition of >. So one must work instead not with the operators as given but with equivalence classes generated by the conditions that $cut_4(P) = cut_2(P)$ and $cut_3(P) = cut_1(P)$. This causes no additional difficulties. **QED.**

4. Related work

Dershowitz [7], Okada (unpublished, see [3]), Cichon *et al* [3] and Tahhan Bittar [26, 27] have drawn attention to the applicability of term rewriting techniques (going back to Gentzen [13]) in proofs of cut elimination. As noted above, Herbelin's own proof of strong cut-elimination for his calculus **LJT** uses the more complex structural induction technique of Dragalin [8].

Herbelin's notation (his "normal $\bar{\lambda}$-expressions") for the terms of **MJ** is similar to ours in the cut-free case; for the terms with cut, he uses a notation involving explicit substitutions, such as $(t[x:=t'])$ where we would use $cut_4(P,M',x.M)$. Our own notation, chosen initially for the applications in [12] (and a forthcoming verification [1] thereof in Coq), made it easier for us to see how to order terms as required for the proof using the r.p.o. theorem. We have adopted a notation used in logic programming [21] for judgments with a "privileged" or "stoup" formula rather than Herbelin's $\Gamma;P \vdash Q$ (with P optional).

Howard [18] also has a calculus which allows a bijective correspondence with (normal) natural deduction; but this correspondence no longer works well when disjunction is taken into account. The intercalation calculi [25, 4] of Sieg and Cittadini are similar, in having formulae in special positions in the sequent, but with extra features to ensure (in the propositional case) termination of the proof search.

5. Conclusion and further work

For a sequent calculus so natural that Gentzen might well have discovered it (rather than $\mathbf{LJ^{cut}}$) as an alternative to natural deduction, we have shown how to use the r.p.o. theorem to prove cut-elimination. Use of this technique is not novel, but is much simpler than the Dragalin-style proof in [16, 17]. (One can even more easily apply the same technique to $\mathbf{LJ^{cut}}$.) Maybe it is possible to adapt the r.p.o. technique to prove strong normalisation of the typed lambda calculus itself, but the difficulty, noted in the last sentence of [17], of including the (in our notation) cut-reduction rule

$$cut_4(P, M', y.cut_3(Q, \lambda x.M, Ms)) \rightarrow cut_3(Q, cut_4(P, M', y.\lambda x.M), cut_2(P, M', y.Ms))$$

is unresolved.

Elsewhere we show (or plan to show) how the **MJ** calculus is well-suited both for applications to inductive arguments [5, 12, 15] about other sequent calculi and for proof search. The same methodology, of finding a calculus between a sequent calculus (admitting lots of permutations) and a natural deduction system (without the immediate subformula property), should also be applied to substructural logics. As suggested by a referee, it would be interesting to relate the Dragalin-style proofs and the r.p.o.-based proofs; in particular, to see which proof methods give effective bounds on the maximal length of cut-reduction sequences. Our suspicion is that it is the former.

6. Acknowledgments

Special thanks are due to Andrew Adams, Adam Cichon, Hugo Herbelin, Frank Pfenning and Elias Tahhan Bittar for making [1], [3], [16], [23] and [27] available before publication.

7. References

[1] Adams, A. A.: “Meta-theory of sequent calculus and natural deduction systems in Coq”, in preparation (St Andrews University).

[2] Andreoli, J.-M.: “Logic programming with focusing proofs in linear logic”, Journal of Logic and Computation, **2** (1992), 297–347.

[3] Cichon, E. A., M. Rusinowitch and S. Selhab: “Cut elimination and rewriting: termination proofs”, in preparation (preprint received in June 1996), INRIA-Lorraine, Nancy, France.

[4] Cittadini, S.: “Intercalation calculus for intuitionistic propositional logic”, Report 29 in Philosophy, Methodology, Logic Series, Carnegie Mellon University (1992)..

[5] Cubric, D.: “Interpolation property for bicartesian closed categories”, Arch. Math. Logic **33** (1994), 291–319.

[6] Danos, V., J. B. Joinet and H. Schellinx: “LKQ and LKT: Sequent calculi for second order logic based upon dual linear decompositions of classical implication”, in “Advances in Linear Logic” (Proceedings of the Cornell Workshop on Linear Logic, edited by J.–Y. Girard, Y. Lafont and L. Regnier), Cambridge University Press (1995), 211–224.

[7] Dershowitz, N.: “Orderings for term-rewriting systems”, Theoretical Computer Science **17** (1982), 279–301.

[8] Dragalin, A.: Mathematical intuitionism — Introduction to proof theory, AMS Translations of Mathematical Monographs **67** (1988).

[9] Dyckhoff, R.: "Contraction-free sequent calculi for intuitionistic logic", Journal of Symbolic Logic **57** (1992), 795–807.

[10] Dyckhoff, R. & L. Pinto: "Uniform proofs and natural deductions", in Proceedings of CADE-12 workshop on "Proof search in type theoretic languages" (edited by D. Galmiche & L. Wallen), Nancy, June 1994.

[11] Dyckhoff, R. & L. Pinto: "A permutation-free sequent calculus for intuitionistic logic", St Andrews University Computer Science Research Report CS/96/9 (August 1996).; revised version from "http://www-theory.dcs.st-and.ac.uk/~rd/".

[12] Dyckhoff, R. & L. Pinto: "Permutability of proofs in intuitionistic sequent calculi", submitted (1996) for publication; abstract appeared in Proceedings of 10th International Congress on Logic, Methodology and Philosophy of Science, held at Florence (1995).

[13] Gentzen, G.: "The collected papers of Gerhard Gentzen", (M. Szabo, editor), North-Holland, Amsterdam 1969.

[14] Girard, J.-Y.,: "A new constructive logic: classical logic", Mathematical Structures in Computer Science **1** (1991), 255–196.

[15] Girard, J.-Y., A. Scedrov & P. Scott : "Normal forms and cut-free proofs as natural transformations", in "Logic from Computer Science", MSRI publications, Y. Moschovakis (editor), **21**, Springer-Verlag, (1992), 217–241.

[16] Herbelin, H.: "A λ-calculus structure isomorphic to sequent calculus structure", preprint, (October 1994); now available at "http://capella.ibp.fr/~herbelin/LAMBDA-BAR-FULL.dvi.gz".

[17] Herbelin, H.: "A λ-calculus structure isomorphic to Gentzen-style sequent calculus structure", Proceedings of the 1994 conference on Computer Science Logic, Kazimierz (Poland), (edited by L. Pacholski & J. Tiuryn), Springer Lecture Notes in Computer Science **933** (1995), 61–75.

[18] Howard, W.A.: "The formulae-as-types notion of construction", in To H. B. Curry, Essays on Combinatory Logic, Lambda Calculus and Formalism, (edited by J.R. Hindley & J.P. Seldin), Academic Press (1980).

[19] Kleene, S. C.: "Permutability of inferences in Gentzen's calculi LK and LJ", Mem. Amer. Math. Soc. (1952), 1–26.

[20] Leivant, D.: "Assumption classes in natural deduction", Zeitschrift für math. Logik **25** (1979), 1–4.

[21] Miller, D.: "FORUM: a multiple-conclusion specification logic", Theoretical Computer Science, **165** (1996), 201–232.

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

[23] Pfenning, F.: "Notes on deductive systems", Carnegie Mellon University, (June 1994).

[24] Prawitz, D.: "Natural deduction", Almquist & Wiksell, Stockholm (1965).

[25] Sieg, W.: "Mechanisms and search", AILA preprint 14 (1992).

[26] Tahhan Bittar, E.: "Gentzen cut elimination for propositional sequent calculus by rewriting derivations", Pub. du Laboratoire de Logique, d'Algorithmique et d'Informatique de Clermont 1, Université d'Auvergne, **16** (1992).

[27] Tahhan Bittar, E.: "Strong normalisation proofs for cut elimination in Gentzen's sequent calculi", Prépub. du Laboratoire de Logique, d'Algorithmique et d'Informatique de Clermont 1, Université d'Auvergne, **56** (1996).

[28] Troelstra, A. S., and D. van Dalen: "Constructivism in mathematics: an introduction (vol 2)", North Holland, 1988.

[29] Troelstra, A. S., and H. Schwichtenberg, "Basic Proof Theory", Cambridge University Press, 1996.