A Note on the Proof Theory of the $\lambda\Pi$–Calculus

David J. Pym
A NOTE ON THE PROOF THEORY OF
THE $\lambda\Pi$-CALCULUS

David J. Pym
Queen Mary and Westfield College
University of London
England, U.K.

Abstract

The $\lambda\Pi$-calculus, a theory of first-order dependent function types in Curry-Howard-de Bruijn correspondence with a fragment of minimal first-order logic, is defined as a system of (linearized) natural deduction. In this paper, we present a Gentzen-style sequent calculus for the $\lambda\Pi$-calculus and prove the cut-elimination theorem.

The cut-elimination result builds upon the existence of normal forms for the natural deduction system and can be considered to be analogous to a proof provided by Prawitz for first-order logic. The type-theoretic setting considered here elegantly illustrates the distinction between the processes of normalization in a natural deduction system and cut-elimination in a Gentzen-style sequent calculus.

We consider an application of the cut-free calculus, via the subformula property, to proof-search in the $\lambda\Pi$-calculus. For this application, the normalization result for the natural deduction calculus alone is inadequate, a (cut-free) calculus with the subformula property being required.

1 Introduction

In a natural deduction presentation of the proofs of the implicational fragment of minimal propositional or predicate calculus, the introduction and elimination rules for implication can be formulated as

$$
\Gamma, \phi \vdash \psi \\
\frac{\Gamma \vdash \psi \triangleright \psi}{}
$$

and

$$
\Gamma \vdash \phi \triangleright \psi \\
\frac{\Gamma \vdash \phi \triangleright \psi, \Gamma \vdash \phi}{\Gamma \vdash \psi}
$$

Such a system is said to be linearized, or sequential, because the hypotheses are recorded in the antecedent (or context) of the sequent, and said to be natural deduction because the connectives receive pairs of introduction rules, which introduce the connective to the right hand side of the sequent, and elimination rules, which eliminate the connective from the right hand side of the sequent.

It is well-known [11, 22] that such a system is equivalent to a sequent calculus, in which the elimination rules are replaced by left rules, which introduce the connective to the left hand side of the sequent, provided the sequent calculus includes the structural rule of cut,

$$
\frac{\Gamma \vdash \phi, \Gamma, \phi \vdash \psi}{\Gamma \vdash \psi}
$$

*This paper was written whilst the author was affiliated to the University of Edinburgh, Scotland, U.K. and revised for publication whilst he was affiliated to the University of Birmingham, England, U.K.
In the implicational fragment of the minimal propositional or predicate calculus, the left rule for implication can be formulated as follows:

\[ \Gamma, \phi \vdash \chi \quad \vdash \quad \Gamma, \phi, \psi \vdash \chi. \]

The right rule for implication is identical to the introduction rule.

For any proof in the natural deduction system there is a corresponding proof in the Gentzen-style system with the cut rule (and vice versa). Moreover, for any proof in the natural deduction system which is in normal form there is a corresponding proof the Gentzen-style system without the cut rule (and vice versa) [11, 22]. We remark that normalization and cut-elimination are not identical [34, 22] — an observation which adumbrates our work in § 3.3.

Considering the antecedents of sequents to be multisets of formulae, and taking axioms of the form \( \Gamma, \phi \vdash \phi \), the structural rules of contraction and weakening,

\[
C \quad \frac{\Gamma, \phi, \psi \vdash \psi}{\Gamma, \phi \vdash \psi}, \quad \text{and} \quad \frac{\Gamma \vdash \psi}{\Gamma, \phi \vdash \psi},
\]

are admissible. (Note that by considering antecedents to be multisets rather than sequences of formulae, we are able to suppress all occurrences of the exchange rules [11].)

A simple partial consequence of the admissibility of contraction is that we can express the left rule for implication as follows:

\[ \Gamma, \phi \vdash \psi \quad \vdash \quad \Gamma, \phi, \psi \vdash \chi. \]

This form is of value when the sequent calculus is used as a basis for proof-search. Specifically, requiring \( \psi \) to match with \( \chi \) yields a resolution (or backchaining) rule, thereby providing a suitable basis for a simple notion of logic programming, whilst the inclusion of the contraction permits the re-use of program clauses. We return to these points in the sequel.

Our purpose in this paper is to demonstrate that a similar analysis obtains for the \( \Lambda \Pi \)-calculus, a theory of first-order dependent function types in Curry-Howard de Bruijn correspondence with a fragment of first-order minimal logic. The \( \Lambda \Pi \)-calculus is the type theory of the logical framework \( LF \) [13, 14, 2, 23, 28], a general theory of natural deduction and Hilbert-style logics based on Martin-Löf’s theory of judgements [18]. \( LF \), a tool for the metatheoretic study of such logics, consists of two components: (1) its type theory; (2) its theory of the representation of logics.\(^1\) The \( \Lambda \Pi \)-calculus is in Curry-Howard-de Bruijn with a small fragment of minimal first-order logic.\(^2\)

The paper is organized as follows: in § 2, we present the \( \Lambda \Pi \)-calculus as a system of natural deduction; in § 3, we give a presentation of the \( \Lambda \Pi \)-calculus as a sequent calculus, proving both its equivalence to the natural deduction presentation and the cut-elimination theorem; finally, in § 4, we discuss the theory of proof-search in the \( \Lambda \Pi \)-calculus, paying particular attention to systems of resolution.

2 The \( \Lambda \Pi \)-calculus

We present the \( \Lambda \Pi \)-calculus, a language introduced in [13, 14]. It is a language with entities of three levels: objects, types and families of types, and kinds. Objects are classified by types, types and

\(^1\) Since the \( \Lambda \Pi \)-calculus has good computational properties, e.g., it is decidable, \( LF \) can be sensibly used as a basis for logic programming systems and theorem provers, e.g., [21, 23, 28, 29].

\(^2\) The extension of the Curry-Howard-de Bruijn, formulae-as-types, correspondence to universal quantifiers and dependent types in the setting of Generalized Type Systems is reported in [3]: the \( \Lambda \Pi \)-calculus is very similar to \( \Lambda P \).
families of types by kinds. The kind Type classifies the types; the other kinds classify functions \( f \) which yield a type \( f(M_1) \ldots (M_n) \) when applied to objects \( M_1, \ldots, M_n \) of certain types determined by the kind of \( f \). Any function definable in the system has a type as domain, while its range can either be a type, if it is an object, or a kind, if it is a family of types. The \( \lambda \Pi \)-calculus is therefore predicative.

The theory we shall deal with is a formal system for deriving assertions of one of the following shapes:

\[ \vdash \Sigma \text{ sig} \quad \Sigma \text{ is a signature} \]
\[ \Gamma \vdash \Sigma \text{ context} \quad \Gamma \text{ is a context} \]
\[ \Gamma \vdash \Sigma \text{ K kind} \quad K \text{ is a kind} \]
\[ \Gamma \vdash \Sigma \text{ A:K} \quad A \text{ has kind } K \]
\[ \Gamma \vdash \Sigma \text{ M:A} \quad M \text{ has type } A \]

where the syntax is specified by the following grammar:

\[ \Sigma ::= \emptyset \mid \Sigma, c : K \mid \Sigma, c : A \]

\[ \Gamma ::= \emptyset \mid \Gamma, x : A \]

\[ K ::= \text{Type} \mid \Pi x : A. K \]

\[ A ::= c \mid \Pi x : A. B \mid \lambda x : A. B \mid A \cdot M \]

\[ M ::= c \mid x \mid \lambda x : A. M \mid MN \]

We let \( M \) and \( N \) range over expressions for objects, \( A \) and \( B \) for types and families of types, \( K \) for kinds, \( x \) and \( y \) over variables, and \( c \) over constants. We also allow \( f, g \) to range variables where the intention is that, in general, these have higher types.

We refer to the collection of (constants) variables declared in a (signature) context \( (\Sigma) \Gamma \) as \( (\text{Dom}(\Sigma)) \text{Dom}(\Gamma) \). We assume \( \alpha \)-conversion throughout. The inference rules of the \( \lambda \Pi \)-calculus appear in Table 1.\(^3\) We shall refer to this system as \( N \) because it is a system of natural deduction. We write \( N \) proves \( \Gamma \vdash \Sigma M : A \), etc., to denote that the assertion \( \Gamma \vdash \Sigma M : A \) is provable in the system \( N \) and we shall sometimes write simply \( \Gamma \vdash \Sigma M : A \), where no confusion can arise.

A term is said to be well-typed in a signature and context if it can be shown to either be a kind, have a kind, or have a type in that signature and context. A term is well-typed if it is well-typed in some signature and context. The notion of \( \beta(\eta) \)-reduction\(^4\), written \( \rightarrow_{\beta(\eta)} \), can be defined both at the level of objects and at the level of types and families of types in the obvious way; for the details, see [14]. \( M =_{\beta(\eta)} N \) if \( M \rightarrow_{\beta(\eta)} P \) and \( N \rightarrow_{\beta(\eta)} P \) for some term \( P \), where * denotes transitive closure. For simplicity we shall write \( \rightarrow_{\beta(\eta)} \) for \( \rightarrow_{\beta(\eta)} \). We write \( NF(U) \) to denote the \( \beta(\eta) \)-normal form of the expression \( U \). We write \( U = V \) to denote the \( \alpha \)-equality of the expressions \( U \) and \( V \) and \( U \equiv V \) to denote their syntactic identity (with the subscript "def" to denote definitions).

We write \( A \rightarrow B \) for \( \Pi x : A. B \) when \( x \) does not occur free in \( B \) and \( A \rightarrow K \) for \( \Pi x : A. K \) when \( x \) does not occur free in \( K \). With the inference rules given in Table 2, this use of \( \rightarrow \) constitutes a conservative extension on the language.

A summary of the major metatheorems pertaining to \( N \) and its reduction properties are given by Theorem 2.1 [14].

**Theorem 2.1 (The Basic Metatheory of the \( \lambda \Pi \)-calculus)**

\(^3\)Note that the inclusion of the premise \( \Gamma \vdash \Pi A : \text{Type} \) in the (III) rule (and in other similar rules) in the system \( N \) is inessential in this definition of the proofs of the \( \lambda \Pi \)-calculus. However, it is consistent with [13] and indeed with the algorithmic definition of the \( \lambda \Pi \)-calculus given in Appendix A of [14], on which the proof of Theorem 2.1 depends. Certain inductive proofs, such as the correctness of the algorithmic formulation, are (technically) simplified by this harmless addition.

\(^4\)Theorem 2.1 holds for both \( \beta \) and \( \beta \eta \) reduction, q.v. [13, 14, 30, 6]. We write \( \beta(\eta) \) where both \( \beta \) and \( \beta \eta \) are possible.
Let $X$ range over basic assertions of the form $A:K$ and $M:A$.

1. Thinning (weakening) is an admissible rule: if $N$ proves $\Gamma \vdash_{\Sigma} X$ and $N$ proves $\vdash_{\Sigma, \Sigma'} \Gamma, \Gamma'$ context, then $N$ proves $\Gamma, \Gamma' \vdash_{\Sigma, \Sigma'} X$.

2. Transitivity is an admissible rule: if $N$ proves $\Gamma \vdash_{\Sigma} M: A$ and $N$ proves $\Gamma, x:A, \Delta \vdash_{\Sigma} X$, then $N$ proves $\Gamma, \Delta[M/x] \vdash_{\Sigma} X[M/x]$.

3. Uniqueness of types and kinds: if $N$ proves $\Gamma \vdash_{\Sigma} M: A$ and $N$ proves $\Gamma \vdash_{\Sigma} M: A'$, then $A =_{\beta(n)} A'$, and similarly for kinds.

4. Subject reduction: if $N$ proves $\Gamma \vdash_{\Sigma} M: A$ and $M \rightarrow^{*}_{\beta(n)} M'$, then $N$ proves $\Gamma \vdash_{\Sigma} M': A$, and similarly for types.

5. All well-typed terms are strongly normalizing.

6. All well-typed terms are Church-Rosser.

7. Each of the five relations defined by the inference system of Table 1 is decidable, as is the property of being well-typed.

8. Predicativity: if $N$ proves $\Gamma \vdash_{\Sigma} M: A$ then the type-free $\lambda$-term obtained by erasing all type information from $M$ can be typed in the Curry type assignment system ([15], Ch. 15, pp. 203-223).

9. Strengthening is an admissible rule: if $N$ proves $\Gamma, x:A, \Gamma' \vdash_{\Sigma} X$ and if $x \not\in \text{FV}(\Gamma') \cup \text{FV}(X)$ then $N$ proves $\Gamma, \Gamma' \vdash_{\Sigma} X$. □
Valid Signature

\( \vdash \emptyset \ \sigma \) \quad (1)
\( \vdash \Sigma \ \sigma \quad \vdash K \ \text{kind} \quad c \notin \text{Dom}(\Sigma) \) \quad (2)
\( \vdash \Sigma, c : K \ \sigma \) \quad (3)
\( \vdash \Sigma \ \sigma \quad \vdash A \ \text{Type} \quad c \notin \text{Dom}(\Sigma) \)
\( \vdash \Sigma, c : A \ \sigma \) \quad (4)

Valid Context

\( \vdash \Sigma \ \sigma \) \quad (5)
\( \vdash \emptyset \ \text{context} \) \quad (6)
\( \vdash \Gamma \ \text{context} \quad \vdash \Pi \ A : \text{Type} \quad z \notin \text{Dom}(\Gamma) \)
\( \vdash \Gamma, z : A \ \text{context} \) \quad (7)
\( \vdash \Sigma \ \sigma \quad \vdash \Gamma \ \text{context} \)
\( \vdash \Pi \ \text{Type} \ \text{kind} \)
\( \vdash \Gamma, z : A \vdash K \ \text{kind} \)
\( \vdash \Sigma, \Pi z : A.K \ \text{kind} \) \quad (8)

Valid Kinds

\( \vdash \Sigma \ \sigma \quad \vdash \Gamma \ \text{context} \quad c : K \in \Sigma \)
\( \vdash \Sigma, c : K \) \quad (9)
\( \vdash \Gamma \ A : \text{Type} \quad \vdash \Pi z : A.B : \text{Type} \)
\( \vdash \Gamma, z : A \Pi z : A.K \) \quad (10)
\( \vdash \Gamma, \lambda z : A.B : \Pi z : A.K \)
\( \Gamma, A \vdash N : A \) \quad (11)
\( \Gamma \vdash B.N : K[N/z] \)
\( \Gamma, A : K \vdash \Gamma \vdash K' \ \text{kind} \quad K' = \beta(z) K' \)
\( \vdash \Gamma \ A : K' \) \quad (12)

Valid Elements of a Kind

\( \vdash \Sigma \ \sigma \quad \vdash \Gamma \ \text{context} \quad c : A \in \Sigma \)
\( \vdash \Sigma, c : A \) \quad (13)
\( \vdash \Sigma \ \sigma \quad \vdash \Gamma \ \text{context} \quad c : A \in \Gamma \)
\( \vdash \Sigma, z : A \) \quad (14)
\( \vdash \Sigma \ A : \text{Type} \quad \vdash \Pi z : A.B \)
\( \vdash \Gamma, z : A \Pi z : A.B \) \quad (15)
\( \vdash \Gamma, \lambda z : A.M : \Pi z : A.B \)
\( \Gamma, A \vdash N : A \) \quad (16)
\( \Gamma \vdash M : N : B[N/z] \)
\( \Gamma, A : A' : \text{Type} \quad A = \beta(z) A' \)
\( \vdash \Sigma, A' : A' \) \quad (17)

Table 1
The proof of Theorem 2.1 is rather complicated. One method, due to Salvesen [30] (see also [14]) adapts the methods developed by van Daalen in his thesis [7] to this type theory. The main difficulty here lies in obtaining the Church-Rosser property in the presence of η-conversion. The essential step in obtaining the proof of this property is to first reformulate the λΠ-calculus as a system with equality judgements in which type labels are explicit, i.e., the assertions of equality have shape Γ ⊢ Π M = N : A, etc. This step is sufficient to allow the methods of van Daalen to go through. The reader is referred to [30] for the details of the proof.\footnote{Similar properties are proved for the system with just β-reduction in [14]. We note that in this proof in order to obtain Part 8, we first prove decidability. In this proof, we prove the Church-Rosser property, strong normalization and require the presence of type labels in order to prove the decidability of the type theory. An alternative approach to the proof of the Church-Rosser property, due to Coquand, is given in [6].}

\section{The Gentzenized λΠ-calculus}

\subsection{The System G}

In this section, we present the system G, a presentation of the λΠ-calculus as sequent calculus, in which the elimination rules for → and II are replaced by left rules. In anticipation of our application to the theory of proof-search in the λΠ-calculus, we consider the left rules in the form described in §1, including a contraction on the principal formula.\footnote{Harper[12] also considers an equational formulation of the λΠ-calculus as a basis for the construction of environment models [19] of the type theory of LF.}

The inference rules of the system G are identical to those of the system N, with the crucial exception that the elimination rules for → and II of the system N are replaced by the following two left rules respectively:

\begin{align*}
\text{Valid Kinds} & \quad \Gamma, z : A \vdash \tau \quad \tau \notin \text{FV}(K) \\
\Gamma & \vdash \tau \quad \text{K kind} \\
\text{Valid Elements of a Kind} & \quad \Gamma, z : A \vdash \tau : B \quad \tau \notin \text{FV}(K) \\
\Gamma, z : A, B : A & \vdash \tau : K \\
\Gamma, z : B : A & \vdash \tau : K \\
\Gamma & \vdash \tau : B \quad \text{K kind} \\
\text{Valid Elements of a Type} & \quad \Gamma, z : A, B : B \vdash \tau \quad \tau \notin \text{FV}(B) \\
\Gamma, z : A & \vdash \tau : B \quad \text{Type} \\
\Gamma & \vdash \tau : \rightarrow \text{I} \\
\Gamma, z : A & \vdash \tau : M \quad \tau \notin \text{FV}(B) \\
\Gamma & \vdash \tau : \lambda x : A. M : A \rightarrow B \\
\Gamma & \vdash \tau : \rightarrow \text{E} \\
\Gamma, z : M : A & \vdash \tau : B \\
\Gamma, z : N : A & \vdash \tau : \text{MN} : B \\
\end{align*}

\begin{table}
\centering
\begin{tabular}{|c|}
\hline
\text{Table 2} \\
\hline
\end{tabular}
\end{table}

\footnote{The following forms for the (→ I) and (ΠI) rules (respectively), not including the contraction on the principal formula, are possible (subject to standard side-conditions on variables):

\begin{align*}
\Gamma, z : C & \vdash \tau : M : A \\
\Gamma, z : B & \vdash \tau : \text{M}[\Pi N / x] : A \\
\Gamma, z : D & \vdash \tau : \text{M}[\Pi N / x] : A \\
\Gamma, z : \Pi : B, C & \vdash \tau : \text{M}[\Pi N / x] : A \\
\end{align*}

Similar rules in which the principal formula extends the signature rather than the context are also possible.}

\section*{References}


\[
\begin{align*}
\forall A \rightarrow B \in \Sigma & \quad \Gamma \vdash \Sigma N : A & \quad \Gamma, y : B \vdash \Sigma M : D & \quad y \notin \text{FV}(D) \\
\Gamma \vdash \Sigma M[\Pi N/y] : D
\end{align*}
\]  
(24)

\[
\begin{align*}
\forall \Pi x : A.B & \in \Sigma \cup \Gamma & \quad \Gamma \vdash \Sigma N : A & \quad B[N/x] =_{\beta \eta} C & \quad \Gamma, y : C \vdash \Sigma M : D & \quad y \notin \text{FV}(D) \\
\Gamma \vdash \Sigma M[\Pi N/y] : D
\end{align*}
\]  
(25)

The right rules for \(\rightarrow\) and \(\Pi\) are identical to the introduction rules, specifically

\[
\begin{align*}
\Gamma \vdash \Sigma \ A : \text{Type} & \quad \Gamma x : A \vdash \Sigma M : B & \quad x \notin \text{FV}(B) \\
\Gamma \vdash \Sigma \lambda x : A. M : A \rightarrow B
\end{align*}
\]  
(26)

and

\[
\begin{align*}
\Gamma \vdash \Sigma \ A : \text{Type} & \quad \Gamma x : A \vdash \Sigma M : B \\
\Gamma \vdash \Sigma \lambda x : A. M : \Pi x : A. B
\end{align*}
\]  
(27)

We include the cut rule,

\[
\begin{align*}
\Gamma \vdash \Sigma M : A & \quad \Gamma, x : A, \Delta \vdash \Sigma X \\
\Gamma, \Delta[M/x] \vdash \Sigma X[M/x]
\end{align*}
\]  
(28)

explicitly in the system \(G\). Note that the rules for kind- and type-equality, (12) and (17) respectively, are the same in \(G\) as in \(N\).

In the sequel, we prove the soundness and completeness of the Gentzen-style system \(G\), with respect to the natural deduction system, \(N\), and prove a cut-elimination theorem for the \(\lambda\Pi\)-calculus by proving that the system \(G\setminus\text{cut}\) is complete with respect to the system \(N\) provided we work with inhabiting objects in \(\beta\)-normal form. While this might seem surprising at first sight, as we might expect completeness for all inhabiting objects, by analogy with usual completeness of systems without cut, it actually corresponds to the fact that in such systems we do not get completeness for proofs; rather for every proof in the system with cut which is in normal form we get a proof in the system without cut. To see this, note that the rules of \(G\setminus\text{cut}\) allow the derivation of \(\beta\)-normal forms only, whereas the cut rule allows the derivation of \(\beta\)-redexes (the term \(M\) in rule (28) can be of the form \(\lambda y : E. P\) and so can introduce a redex when substituted for \(x\)). The relationship between \(G\) and \(N\) is thus directly analogous to that between LJ and NJ [11, 22]. Indeed, our proof of the completeness of the system without cut can be considered to be analogous to that given by Prawitz [22] for proofs in normal form. Our work is closely related to Avron's account of Schroeder-Heister's analysis of natural deduction [1, 31]. The extension of the formulae-as-types correspondence to Gentzen-style Generalized Type Systems and sequent calculi is the subject of current research. The admissibility of weakening, contraction and exchange (or permutation) in \(G\) is discussed below.

We remark that we should like to present Gentzen-style replacements for the application rules for kinds, (20) and (11). However, for the given syntax of the \(\lambda\Pi\)-calculus this is not possible because kind declarations are not permitted in contexts. In particular, this means that we are not able to have a premise of the form \(\Gamma, x : K \vdash \Sigma X\), where \(K\) is a kind expression, which would be essential for such a rule. However, if we reformulate the \(\lambda\Pi\)-calculus without signatures but with kind declarations in contexts, such rules are available. In particular, the (ILL) rule for both kinds and types can then take the form:

\[
\begin{align*}
\forall \Pi x : A.R & \in \Sigma & \Gamma \vdash \Sigma N : A & \quad R[N/x] =_{\beta \eta} S & \quad \Gamma, y : S \vdash \Sigma U : V & \quad y \notin \text{FV}(D) \\
\Gamma \vdash \Sigma U[\Pi N/y] : V
\end{align*}
\]  
(29)

where \(R\) and \(S\) are either kind expressions or type expressions and where \(U : V\) asserts either that the type \(U\) has kind \(V\) or that the object \(U\) has type \(V\). The corresponding \((\rightarrow L)\) rule is similar.
A natural deduction presentation of such a system without signatures but with kind declarations permitted in contexts can be found in [14]; and for this system it is possible to prove all of the relevant results of this paper for a Gentzen-style system with rules of the form described above.

We conclude this section by noting that the rules of weakening, contraction and permutation are admissible in G.

**Proposition 3.1 (Admissibility of Weakening)** The weakening rule:

\[
\begin{array}{c}
\Gamma \vdash \Sigma X \quad \vdash_{\Sigma, \emptyset} \emptyset, \Gamma', \Gamma \text{ context} \\
\hline
\Gamma, \Gamma' \vdash_{\Sigma, \emptyset} X
\end{array}
\]

is admissible in G.

**Proof** The proof is by induction on the structure of proofs in G, and we omit the details. □

**Proposition 3.2 (Admissibility of Contraction)** The contraction rule:

\[
\begin{array}{c}
\Gamma, x : A, \Delta, y : A, \Theta \vdash_{\Sigma} X \\
\hline
\Gamma, x : A, \Delta, \Theta[x/y] \vdash_{\Sigma} X[x/y]
\end{array}
\]

is admissible in G.

**Proof** The proof is by induction on the structure of proofs in G, and we omit the details. We remark that the contraction rule amounts to the cut rule for the special case in which the substituting object is a variable. □

**Proposition 3.3 (Admissibility of Exchange)** The exchange rule:

\[
\begin{array}{c}
\Gamma, x : A, y : B, \Delta \vdash_{\Sigma} X \\
x \notin \text{FV}(B)
\hline
\Gamma, y : B, x : A, \Delta \vdash_{\Sigma} X
\end{array}
\]

is admissible in G.

**Proof** The proof is by induction on the structure of proofs in G; we omit the details. □

We remark that this exchange rule is a very weak one compared to that which obtains in [11]. This is because of the partial ordering of the contexts of the λΠ-calculus induced by the presence of dependent types in the language. For discussions of the algebraic structure of type-theoretic contexts see, e.g., [4]. The weakness of the exchange rule adumbrates much of the work of [26, 28] and Chapter 6 of [23, 28] which develop a certain kind of Herbrand analysis [32] for the λΠ-calculus.

### 3.2 Soundness and Completeness of G

In this section, we prove the soundness and completeness of G with respect to N. These results crucially depend upon the presence of the cut rule in G but are quite straightforward.

**Theorem 3.4 (Soundness of G)** If G proves \( \Gamma \vdash_{\Sigma} U : V \) then N proves \( \Gamma \vdash_{\Sigma} U : V \).

**Proof** The proof proceeds by induction on the structure of proofs in the system G.

Since by Theorem 2.1 the cut rule is admissible in N, the only remaining differences between N and G are that the \((\to E)\) and \((\text{II}E)\) rules of N are replaced in G by the \((\to L)\) and \((\text{II}L)\) rules.
Thus there are many trivial cases in the induction, which we omit, and two more difficult cases. We give the case in which the last rule applied is the (II) rule and remark that the case in which the last rule applied is the \((\to L)\) is similar.

Since the systems are different only for assertions that an object inhabits a type, we assume that \(U:V\) is of the form \(M:A\).

Suppose the last rule of \(G\) applied is \((\Pi L)\),

\[
\Gamma \vdash_{\Sigma} N:A @: \Pi x:A. B(x) \in \Sigma, \Gamma \vdash_{\Sigma} B[N/x] =_{\beta_\eta} C \quad \Gamma, y:C \vdash_{\Sigma} M:D \quad y \not\in \text{FV}(D)
\]

\[
\Gamma \vdash_{\Sigma} M[\downarrow N/y]:D
\]

By the induction hypothesis we have that \(N\) proves \(\Gamma \vdash_{\Sigma} N:A\), that \(N\) proves \(\Gamma, y:C \vdash_{\Sigma} M:D\), that \(N\) proves \(\Gamma \vdash_{\Sigma} @: \Pi x:A. B(x)\) and that \(B[N/x] =_{\beta_\eta} C\). By \((\Pi E)\), from \(N\) proves \(\Gamma \vdash_{\Sigma} @: \Pi x:A. B(x)\) and \(N\) proves \(\Gamma \vdash_{\Sigma} N:A\), we obtain \(N\) proves \(\Gamma \vdash_{\Sigma} @N:B[N/x]\) by (16), and so obtain \(N\) proves \(\Gamma \vdash_{\Sigma} @N:C\) by the equality rule for types (17). From this and \(N\) proves \(\Gamma \vdash_{\Sigma} M:D\) we obtain \(N\) proves \(\Gamma \vdash_{\Sigma} M[\downarrow N/y]:D\) by the cut rule (which is admissible in \(N\) by Theorem 2.1). This argument is summarized conveniently by the following proof figure:

\[
\begin{array}{c}
\frac{\beta_\eta \Gamma \vdash \Pi x:A. B(x) \in \Sigma, \Gamma \vdash_{\Sigma} B[N/x] =_{\beta_\eta} C}{\beta_\eta \Gamma, y:C \vdash_{\Sigma} M:D}
\end{array}
\]

(16)

\[
\begin{array}{c}
\Gamma \vdash_{\Sigma} \Pi x:A. B(x) \quad \Gamma \vdash_{\Sigma} N:A
\end{array}
\]

(17)

\[
\begin{array}{c}
\beta_\eta \Gamma \vdash_{\Sigma} @N:B[N/x] \\
\beta_\eta \Gamma \vdash_{\Sigma} @N:C
\end{array}
\]

(Cut).

Note that the equality rule for types requires the premise \(\Gamma \vdash_{\Sigma} C:\text{Type}\). This assertion is readily obtained by induction on the structure of the proof in \(N\) of the assertion \(\Gamma, y:C \vdash_{\Sigma} M:D\), and for clarity we omit this from the figure. □

**Theorem 3.5 (Completeness of \(G\))** If \(N\) proves \(\Gamma \vdash_{\Sigma} U:V\) then \(G\) proves \(\Gamma \vdash_{\Sigma} U:V\).

**Proof** The proof proceeds by induction on the structure of proofs in the system \(N\). The only differences between \(N\) and \(G\) are that the \((\to E)\) and \((\Pi E)\) rules of \(N\) are replaced in \(G\) by the \((\to L)\) and \((\Pi L)\) rules. Thus there are several simple cases in the induction, which we omit, and two more difficult cases. We give the case in which the last rule applied is the \((\Pi E)\) rule and remark that the case in which the last rule applied is the \((\to E)\) is similar.

Again, since the systems are only different for assertions that an object inhabits a type, we assume that \(U:V\) is of the form \(M:A\).

Suppose the last rule of \(N\) applied is the \((\Pi E)\) rule,

\[
\Gamma \vdash_{\Sigma} M : \Pi x:A. B \quad \Gamma \vdash_{\Sigma} N:A
\]

\[
\Gamma \vdash_{\Sigma} MN:B[N/x]
\]

By the induction hypothesis we have that \(G\) proves \(\Gamma \vdash_{\Sigma} M : \Pi x:A. B\) and \(G\) proves \(\Gamma \vdash_{\Sigma} N:A\), and from these two assertions we obtain, by induction on the structure of their proofs in \(G\), that \(G\) proves \(\Gamma \vdash_{\Sigma} A:\text{Type}\) and \(G\) proves \(\Gamma, x:A \vdash_{\Sigma} B:\text{Type}\), so that we obtain \(G\) proves \(\Gamma, y : \Pi x:A. B, z:A, a:B(z) \vdash_{\Sigma} \exists x:B(x)\) and \(G\) proves \(\Gamma, y : \Pi x:A. B, z:A \vdash_{\Sigma} x:A\). Therefore by the \((\Pi L)\) rule we obtain \(G\) proves \(\Gamma, y : \Pi x:A. B, x:A \vdash_{\Sigma} yz:B(x)\). From this assertion and the assertion that \(G\) proves \(\Gamma \vdash_{\Sigma} N:A\) we obtain \(G\) proves \(\Gamma, y : \Pi x:A. B \vdash_{\Sigma} yN:B[N/x]\) by the cut rule. From this assertion and the assertion that \(G\) proves \(\Gamma \vdash_{\Sigma} M : \Pi x:A. B\) we obtain \(G\) proves \(\Gamma \vdash_{\Sigma} MN:B[N/x]\) by the cut rule. □
3.3 Cut-elimination

In this section, we prove the cut-elimination theorem for the $\lambda\Pi$-calculus. This result obtains only for $\beta$-normal forms. We write $G\setminus\text{cut}$ for the system obtained by removing the cut rule from the system $G$. A brief discussion of our strategy is in order. Rather than prove that any proof in $G$ can be transformed into one in $G\setminus\text{cut}$ with the same endsequent, we prove that $\beta$-normal forms are provable in $N$ if and only if they are provable in $G\setminus\text{cut}$. Thus, as remarked in $\S$ 3.1, our proof of cut-elimination for the $\lambda\Pi$-calculus can be considered to be analogous to that of Prawitz [22] for first-order logic.

We begin by proving the soundness and completeness of $G\setminus\text{cut}$ with respect to $N$. The completeness part of this is the core of the cut-elimination result.

**Theorem 3.6 (Soundness of $G\setminus\text{cut}$)** If $G\setminus\text{cut}$ proves $\Gamma \vdash_\Sigma U : V$ then $N$ proves $\Gamma \vdash_\Sigma U : V$.

**Proof** This is an immediate consequence of the soundness of $G$ with respect to $N$. $\Box$

The next theorem is the completeness of the system $G\setminus\text{cut}$ with respect to the system $N$. The proof of this theorem depends crucially upon a simple technical device. This device consists in the replacement of the $(\Pi\Pi)\text{ rule}$ of the system $N$ by an apparently weaker version of this rule, $(\Pi\Pi)^\circ$:

$$\frac{\Gamma \vdash_\Sigma N : A \quad \Gamma \vdash_\Sigma M : \Pi x : A . B \quad \Gamma \vdash_\Sigma B[N/x] : \text{Type}}{\Gamma \vdash_\Sigma MN : B[N/x]} \quad (30)$$

This rule appears weaker than $(\Pi\Pi)$ because it requires the extra premiss, however it is a relatively straightforward matter to prove that system $N^\circ$, which is obtained from $N$ by the replacement of $(\Pi\Pi)$ by $(\Pi\Pi)^\circ$, is equivalent to $N$.

**Lemma 3.7 (Equivalence of $(\Pi\Pi)^\circ$ and $(\Pi\Pi)$)** $N^\circ$ proves $\Gamma \vdash_\Sigma M : A$ if and only if $N$ proves $\Gamma \vdash_\Sigma M : A$.

**Proof** It follows immediately that if $N^\circ$ proves $\Gamma \vdash_\Sigma M : A$ then $N$ proves $\Gamma \vdash_\Sigma M : A$.

Conversely, suppose that $N$ proves $\Gamma \vdash_\Sigma M : A$; we must prove that $N^\circ$ proves $\Gamma \vdash_\Sigma M : A$. The proof is by induction on the structure of proofs in $N$ and the only case of interest is when the last rule applied is $(\Pi\Pi)$:

$$\frac{\Gamma \vdash_\Sigma N : A \quad \Gamma \vdash_\Sigma M : \Pi x : A . B}{\Gamma \vdash_\Sigma MN : B[N/x]} \quad (30)$$

By the induction hypothesis, we have that $N^\circ$ proves $\Gamma \vdash_\Sigma M : \Pi x : A . B$ and $N^\circ$ proves $\Gamma \vdash_\Sigma N : A$. From the first of these assertions we obtain, by induction on the structure of proofs, that $N^\circ$ proves $\Gamma, x : A \vdash_\Sigma B : \text{Type}$. It is easy to see that the cut rule is admissible in $N^\circ$; so from the last two assertions we obtain, by the cut rule, that $N^\circ$ proves $\Gamma \vdash_\Sigma B[N/x] : \text{Type}$. We then obtain $N^\circ$ proves $\Gamma \vdash_\Sigma MN : B[N/x]$ by the $(\Pi\Pi)^\circ$ rule. $\Box$

The proof of the completeness theorem is facilitated by the next lemma.

**Lemma 3.8 (Application in $\beta$-normal form)** If $M$ is a $\beta$-normal form and if the last rule applied in the proof in the system $N$ of $\Gamma \vdash_\Sigma M : A$ is either $\rightarrow E$ or $(\Pi\Pi)$, then $M$ is of the form $\oplus M_1 \ldots M_n$ for some $\oplus \in \text{Dom}(\Sigma) \cup \text{Dom}(\Gamma)$ of appropriate type.

---

7The proof is essentially the same as that for $N$. 

10
PROOF. We illustrate a simple case of the proof. Suppose the last rule applied is the \((\rightarrow E)\) rule,
\[
\frac{\Gamma \vdash \Sigma N : A \to B \quad \Gamma \vdash \Sigma P : A}{\Gamma \vdash \Sigma NP : B}
\]
If \(NP\) is a \(\beta\)-normal form, then \(N\) cannot be of the form \(\lambda x : A . Q\). Therefore \(N\) is of the form \(\oplus M_1 \ldots M_m\), for some \(\oplus\) of appropriate type. \(\square\)

We are now ready to prove the completeness of the system \(G\setminus \text{cut}\) with respect to the system \(N\). However, the proof is rather complex, so first we give an example of the essential part of the argument. By Lemma 3.7, it is sufficient to prove the completeness of the system \(G\setminus \text{cut}\) with respect to the system \(N^\oplus\). So the proof proceeds by induction on the structure of proofs in \(N^\oplus\).

Suppose that the last inference in the system \(N^\oplus\) is of the form
\[
\frac{\Gamma \vdash \Sigma \oplus N_1 N_2 : B \to C \quad \Gamma \vdash \Sigma N : B}{\Gamma \vdash \Sigma \oplus N_1 N_2 N : C}
\]
where \(\oplus : \Pi x_1 : B_1 . B_2 \to (B^* \to C^*) \in \Sigma \cup \Gamma\) and \((B^* \to C^*)[N_1/x_1] = \rho_{\eta} B \to C\).

The induction hypothesis will immediately give us that \(G\setminus \text{cut}\) proves \(\Gamma \vdash \Sigma \oplus N_1 N_2 : B \to C\) and that \(G\setminus \text{cut}\) proves \(\Gamma \vdash \Sigma N : B\); we also have that \(G\setminus \text{cut}\) proves \(\Gamma \vdash \Sigma N_1 : B_1\) and that \(G\setminus \text{cut}\) proves \(\Gamma \vdash \Sigma N_2 : B_2[N_1/x_1]\).

By a simple inductive argument from the proof in \(N^\oplus\) of \(\Gamma \vdash \Sigma \oplus N_1 N_2 : B \to C\), it is easy to see that we can obtain, noting that the rule for type-equality (17) is available in both \(N^\oplus\) and \(G\setminus \text{cut}\), shorter proofs of

\[
N^\oplus \text{ proves } \Gamma \vdash \Sigma (B_2 \to (B^* \to C^*))[N_1/x_1] : \text{Type}, \quad (31)
\]
\[
N^\oplus \text{ proves } \Gamma \vdash \Sigma B \to C : \text{Type} \quad (32)
\]
and
\[
N^\oplus \text{ proves } \Gamma \vdash \Sigma C : \text{Type}; \quad (33)
\]
so by the induction hypothesis we have that
\[
G\setminus \text{cut proves } \Gamma \vdash \Sigma (B_2 \to (B^* \to C^*))[N_1/x_1] : \text{Type}, \quad (34)
\]
\[
G\setminus \text{cut proves } \Gamma \vdash \Sigma B \to C : \text{Type} \quad (35)
\]
and
\[
G\setminus \text{cut proves } \Gamma \vdash \Sigma C : \text{Type}. \quad (36)
\]

Given these data, we can construct a proof of
\[
G\setminus \text{cut proves } \Gamma \vdash \Sigma \oplus N_1 N_2 : C. \quad (37)
\]
From (34), (35) and (36), it easy to see that.
\[ G \text{ cut proves } \Gamma, x_1: (B_2 \to (B^* \to C^*))[N_1/x_1], y: B \to C, z: C \text{ context}, \] (38)

so that we immediately obtain the axiom

\[ G \text{ cut proves } \Gamma, x_1: (B_2 \to (B^* \to C^*))[N_1/x_1], y: B \to C, z: C \vdash \Sigma \ yN : C. \] (39)

By the \((\to L)\) rule, from (39) and \(G \text{ cut proves } \Gamma \vdash \Sigma \ N : B\), we obtain

\[ G \text{ cut proves } \Gamma, x_1: (B_2 \to (B^* \to C^*))[N_1/x_1], y: B \to C \vdash \Sigma \ yN : C. \] (40)

By the \((\to L)\) rule again, from (40) and \(G \text{ cut proves } \Gamma \vdash \Sigma \ N_2 : B_2[N_1/x_1]\), we obtain

\[ G \text{ cut proves } \Gamma, x_1: (B_2 \to (B^* \to C^*))[N_1/x_1], y: B \to C \vdash \Sigma \ x_1N_2N : C. \] (41)

From (41) and \(G \text{ cut proves } \Gamma \vdash \Sigma \ N_1 : B_1\), by the \((\text{ILL})\) rule, we obtain

\[ G \text{ cut proves } \Gamma \vdash \Sigma \ @N_1N_2N : C, \] (42)

as required.

**Theorem 3.9 (Completeness of G cut)** If \(N\) proves \(\Gamma \vdash \Sigma \ U : V\) and if \(U\) is a \(\beta\)-normal form then \(G \text{ cut} \) proves \(\Gamma \vdash \Sigma \ U : V\).

**Proof** By Lemma 3.7, it is sufficient to show the completeness of \(G \text{ cut}\) with respect to \(N^\circ\).

The proof of follows a pattern that is analogous to that employed by Prawitz [22] in his proof of the completeness of a sequent calculus formulation of first-order logic (without the cut rule) with respect to a natural deduction formulation. The idea is to proceed by induction on the structure of proofs in \(N^\circ\) and to exploit the structure of the \(\beta\)-normal form \(U\).\(^8\) The only differences between \(N^\circ\) and \(G \text{ cut}\) are that the \((\to E)\) and \((\text{ILE}^\circ)\) rules of \(N^\circ\) are replaced in \(G \text{ cut}\) by the \((\to L)\) and \((\text{ILL})\) rules. In particular, note that the rule for type-equality (17) is available in both \(N^\circ\) and \(G \text{ cut}\). Thus there are several trivial cases in the induction, which we omit, and two rather more difficult cases, which we include. Again, since the systems are different only for assertions that an object inhabits a type, we assume that \(U : V\) is of the form \(M : A\).

Before proceeding with the cases for the \((\to E)\) and \((\text{ILE}^\circ)\) rules, it is convenient to introduce a notation for the type of an atom \(\&\). We denote the type of an atom \(\&\) by an expression of the form

\[ A_1 r_1 A_2 r_2 \ldots r_{n-1} A_n r_n B(x_1, \ldots, x_n), \]

in which each \(r_i, 1 \leq i \leq n\), denotes either a \(\Pi\)-connective, in which case the variable associated with it is \(x_i\), or a \(\to\)-connective, with association to the right: e.g., if \(r_1\) denotes a \(\Pi\), \(r_2\) denotes an \(\to\) and \(r_3\) denotes an \(\to\) then the expression \(A_1 r_1 A_2 r_2 A_3 r_3 B(x_1, x_2, x_3)\) denotes the type \(\Pi x_1 : A_1. (A_2(x_1) \to (A_3(x_1) \to B(x_1)))\).

Suppose the last rule of \(N^\circ\) applied is the \((\to E)\) rule (this is the first difficult case),

\(^8\)See also the remarks in §§1 - 5 of [16].
\[
\Gamma \vdash_{\Sigma} M : B \rightarrow C \quad \Gamma \vdash_{\Sigma} N : B
\]
\[
\Gamma \vdash_{\Sigma} MN : C
\]

By hypothesis, \( MN \) is a \( \beta \)-normal form, so by Lemma 3.8, \( M \) must be of the form \( \@N_1 \ldots N_m \), where \( \@ \) is in \( \Sigma \cup \Gamma \). Thus we must prove that \( G \\text{\textbackslash cut} \) proves \( \Gamma \vdash_{\Sigma} \@N_1 \ldots N_m : C \), given that \( G \\text{\textbackslash cut} \) proves \( \Gamma \vdash_{\Sigma} \@N_1 \ldots N_m : B \rightarrow C \) and \( G \\text{\textbackslash cut} \) proves \( \Gamma \vdash_{\Sigma} N : B \). This is proved by exploiting the structure of the type of \( \@ \).

Using the notation introduced above, we represent the type of \( \@ \) by the expression

\[
B_1 r_1 B_2 r_2 B_3 r_3 \ldots r_{m-1} B_m r_m (B^* \rightarrow C^*),
\]

where \((B^* \rightarrow C^*)[N_1/x_1, \ldots, N_{m-1}/x_{m-1}] =_{\beta\eta} B \rightarrow C\). We assume that the variable associated with \( r_i \), if it occurs, is \( x_i \).

By hypothesis, we have that \( N^\o \) proves \( \Gamma \vdash_{\Sigma} \@N_1 \ldots N_m : B \rightarrow C \) and so by induction on the structure of proofs we have that, for \( 1 \leq i \leq n \),

\[
N^\o \text{ proves } \Gamma \vdash_{\Sigma} N_i : B_i[N_1/x_1, \ldots, N_{i-1}/x_{i-1}],
\]

by shorter proofs. Therefore by the induction hypothesis we have that

\[
G \\text{\textbackslash cut} \text{ proves } \Gamma \vdash_{\Sigma} B_i[N_1/x_1, \ldots, N_{i-1}/x_{i-1}],
\]

and that the \( N_i \) are \( \beta \)-normal forms, for \( 1 \leq i \leq m \). From this we obtain that

\[
G \\text{\textbackslash cut} \text{ proves } \Gamma \vdash_{\Sigma} B_i[N_1/x_1, \ldots, N_{i-1}/x_{i-1}] : \text{Type}.
\]

It follows that we can construct a finite set of contexts which progressively extend \( \Gamma \):

\[
\begin{align*}
\Delta_1 & \equiv \text{def } \Gamma, y_1 : B_2[N_1/x_1] r_2 \ldots r_{m-1} B_m[N_1/x_1] r_m (B^* \rightarrow C^*)[N_1/x_1] \\
\Delta_2 & \equiv \text{def } \Delta_1, y_2 : B_3[N_1/x_1, N_2/x_2] r_3 \ldots r_{m-1} B_m[N_1/x_1, N_2/x_2] r_m (B^* \rightarrow C^*)[N_1/x_1, N_2/x_2] \\
& \vdots \\
\Delta_m & \equiv \text{def } \Delta_{m-1}, y_{m-1} : B_m[N_1/x_1, \ldots, N_{m-1}/x_{m-1}] r_m (B^* \rightarrow C^*)[N_1/x_1, \ldots, N_{m-1}/x_{m-1}],
\end{align*}
\]

where \( y_i \notin \text{Dom}(\Gamma) \) and \( y_i \notin \text{Dom}(\Gamma), 1 \leq i \leq m-1 \). It is easy to verify, for \( 1 \leq i \leq m \), that

\[
G \\text{\textbackslash cut} \text{ proves } \Gamma \vdash_{\Sigma} \Delta_i \text{ context}.
\]

By the induction hypothesis and the admissibility of weakening we have that \( G \\text{\textbackslash cut} \) proves \( \Delta_m \vdash_{\Sigma} N : B \). Since \( N^\o \) proves \( \Gamma \vdash_{\Sigma} \@N_1 \ldots N_m : B \rightarrow C \), we obtain, by the induction hypothesis, that \( G \\text{\textbackslash cut} \) proves \( \Delta_m \vdash_{\Sigma} B \rightarrow C : \text{Type} \) and that \( G \\text{\textbackslash cut} \) proves \( \Delta_m \vdash_{\Sigma} C : \text{Type} \). Therefore we have that \( G \\text{\textbackslash cut} \) proves \( \Delta_m, y : B \rightarrow C, z : C \vdash_{\Sigma} z : C \), where \( y, z \notin \text{Dom}(\Gamma) \). Therefore, by the \((\rightarrow L)\) rule we obtain \( G \\text{\textbackslash cut} \) proves \( \Delta_m, y : B \rightarrow C \vdash_{\Sigma} yN : C \). By the induction hypothesis we have that \( G \\text{\textbackslash cut} \) proves \( \Delta_m \vdash_{\Sigma} N_m : B_m[N_1/x_1, \ldots, N_{m-1}/x_{m-1}] \) and so by either the \((\rightarrow L)\) or the \((\Pi L)\) rule, we obtain

\[
G \\text{\textbackslash cut} \text{ proves } \Delta_{m-1}, y_{m-1} : B_m[N_1/x_1, \ldots, N_{m-1}/x_{m-1}] r_m (B^* \rightarrow C^*)[N_1/x_1, \ldots, N_{m-1}/x_{m-1}] \vdash_{\Sigma} y_{m-1}N_mN : C.
\]

13
By the induction hypothesis we have that $G\setminus cut$ proves $\Delta_{m-1} \vdash_{\Sigma} N_{m-1} : B_{m-1}$ and so by either the $(\rightarrow L)$ rule or by the $(\Pi L)$ rule we obtain

$$G\setminus cut \text{ proves } \Delta_{m-2}, y_{m-2} : B_{m-1}^{*}[N_{i}/x_1, \ldots, N_{m-2}/x_{m-2}] \vdash_{m} r_{m-1}$$

$$(B^{*} \rightarrow C^{*})[N_{i}/x_1, \ldots, N_{m-2}/x_{m-2}] \vdash_{E} x_{m-2} N_{m-1} N_{m} N : C.$$ 

We proceed similarly until we obtain $G\setminus cut$ proves $\Delta_{1} \vdash_{\Sigma} y_{1} N_{2} \ldots N_{m} N : C$. By either the $(\rightarrow L)$ rule or by the $(\Pi L)$ rule we obtain $G\setminus cut$ proves $\Gamma \vdash_{\Sigma} \emptyset N_{1} \ldots N_{m} N : C$, as required.

Suppose the last rule of $N^{\emptyset}$ applied is the $(\Pi E^{\emptyset})$ rule (this is the second difficult case),

$$\frac{\Gamma \vdash_{\Sigma} M : \Pi x : B \cdot C \quad \Gamma \vdash_{\Sigma} N : B \quad \Gamma \vdash_{\Sigma} C[N/x] : Type}{\Gamma \vdash_{\Sigma} M N : C[N/x]}$$

We have, by hypothesis, that $MN$ is a $\beta$-normal form so that, by Lemma 3.8, $M$ must be of the form $\emptyset N_{1} \ldots N_{m}$, where $\emptyset$ is in $\Sigma \cup \Gamma$. Thus we must prove that

$$G\setminus cut \text{ proves } \Gamma \vdash_{\Sigma} \emptyset N_{1} \ldots N_{m} N : C[N/x],$$

given that $G\setminus cut$ proves $\Gamma \vdash_{\Sigma} \emptyset N_{1} \ldots N_{m} : \Pi x : B \cdot C$ and $G\setminus cut$ proves $\Gamma \vdash_{\Sigma} N : B$.

Again, this is proved by exploiting the structure of the type of $\emptyset$. Using the notation introduced above, we represent the type of $\emptyset$ by the expression

$$B_1 r_1 B_2 r_2 B_3 r_3 \ldots r_{m-1} B_m r_m (\Pi x : B^{*} \cdot C^{*}),$$

where $(\Pi x : B^{*} \cdot C^{*})[N_{i}/x_1, \ldots, N_{m}/x_{m}] =_{\beta \eta} \Pi x : B \cdot C$. We assume that the variable associated with $r_i$, if it occurs, is $x_i$.

By hypothesis, we have that $N^{\emptyset}$ proves $\Gamma \vdash_{\Sigma} \emptyset N_{1} \ldots N_{m} : \Pi x : B \cdot C$. By induction on the structure of proofs, it follows that, for $1 \leq i \leq n$,

$$N^{\emptyset} \text{ proves } \Gamma \vdash_{\Sigma} \emptyset N_{i} : B_{i}[N_{i}/x_1, \ldots, N_{i-1}/x_{i-1}],$$

by shorter proofs. Therefore by the induction hypothesis we have that

$$G\setminus cut \text{ proves } \Gamma \vdash_{\Sigma} \emptyset N_{i} : B_{i}[N_{i}/x_1, \ldots, N_{i-1}/x_{i-1}],$$

and that the $N_{i}$ are $\beta$-normal forms, for $1 \leq i \leq m$. From this we obtain that $G\setminus cut$ proves $\Gamma \vdash_{E} B_{i}[N_{i}/x_1, \ldots, N_{i-1}/x_{i-1}] : Type$. It follows that we are able to construct a finite set of contexts which progressively extend $\Gamma$:

$$\Delta_{1} \equiv_{\text{def}} \Gamma, y_{1} : B_{2}[N_{1}/x_1] r_{2} \ldots r_{m-1} B_{m}[N_{1}/x_1] r_{m} (\Pi x : B^{*} \cdot C^{*})[N_{1}/x_1]$$

$$\Delta_{2} \equiv_{\text{def}} \Delta_{1}, y_{2} : B_{3}[N_{1}/x_1, N_{2}/x_2] r_{3} \ldots r_{m-1} B_{m}[N_{1}/x_1, N_{2}/x_2] r_{m}$$

$$(\Pi x : B^{*} \cdot C^{*})[N_{1}/x_1, N_{2}/x_2]$$

$$\vdots$$

$$\Delta_{m} \equiv_{\text{def}} \Delta_{m-1}, y_{m-1} : B_{m}[N_{1}/x_1, \ldots, N_{m-1}/x_{m-1}] r_{m}$$

$$(\Pi x : B^{*} \cdot C^{*})[N_{1}/x_1, \ldots, N_{m-1}/x_{m-1}],$$

where $y, z \notin \text{Dom}(\Gamma)$ and $y_i \notin \text{Dom}(\Gamma), 1 \leq i \leq m - 1$. It is easy to verify, for $1 \leq i \leq m$, that

$$G\setminus cut \text{ proves } \Gamma \vdash_{\Sigma} \Delta_{i} \text{ context.}$$
By the induction hypothesis and the admissibility of weakening we have that $G\setminus\text{cut}$ proves $\Delta_m \vdash_\Sigma N : B$. Since $N^\oplus$ proves $\Gamma \vdash_\Sigma \Pi_1 \ldots \Pi_m : \Pi x : B.C : \text{Type}$, we obtain, by the induction hypothesis, that $G\setminus\text{cut}$ proves $\Delta_m \vdash_\Sigma \Pi x : B.C : \text{Type}$ and, making use of the additional premiss in the (IE$^\oplus$) rule compared to the (IE) rule, that

$$G\setminus\text{cut} \text{ proves } \Delta_m \vdash_\Sigma C[N/x]:\text{Type}.$$ 

Therefore we have that $G\setminus\text{cut}$ proves $\Delta_m, y : \Pi x : B.C, z : C[N/x] \vdash_\Sigma z : C[N/x]$, where $y, z \notin \text{Dom}(\Gamma)$. Therefore, by the (ILL) rule we obtain $G\setminus\text{cut}$ proves $\Delta_m, y : \Pi x : B.C \vdash_\Sigma y : N : C[N/x]$. By the induction hypothesis we have that $G\setminus\text{cut}$ proves $\Delta_m \vdash_\Sigma N_m : B_m$ and so by either the $(\rightarrow L)$ or the (ILL) rule we obtain

$$G\setminus\text{cut} \text{ proves } \Delta_{m-1}, y_{m-1} : B_m^*[N_1/x_1, \ldots, N_{m-1}/x_{m-1}] \vdash_\Sigma y_{m-1}.$$ 

$$(\Pi x : B^*[N_1/x_1, \ldots, N_{m-1}/x_{m-1}]) \vdash_\Sigma y_{m-1} N_m : C[N/x].$$ 

By the induction hypothesis we have that $G\setminus\text{cut}$ proves $\Delta_{m-1} \vdash_\Sigma N_{m-1} : B_{m-1}$ and so by the either the $(\rightarrow L)$ rule or by the (ILL) rule we obtain

$$G\setminus\text{cut} \text{ proves } \Delta_{m-2}, y_{m-2} : B_{m-1}^*[N_1/x_1, \ldots, N_{m-2}/x_{m-2}] \vdash_\Sigma y_{m-2}.$$ 

$$(\Pi x : B_{m-1}^*[N_1/x_1, \ldots, N_{m-2}/x_{m-2}]) \vdash_\Sigma y_{m-2} N_{m-1} : C[N/x].$$ 

We proceed similarly until we obtain $G\setminus\text{cut}$ proves $\Delta_1 \vdash_\Sigma y_1 N_2 \ldots N_m : C[N/x]$. By either the $(\rightarrow L)$ rule or by the (ILL) rule we obtain $G\setminus\text{cut}$ proves $\Gamma \vdash_\Sigma \Pi_1 \ldots \Pi_m N : C[N/x]$, as required. $\square$

Cut-elimination is a simple corollary of this theorem.

**Corollary 3.10 (Cut-Elimination)** Suppose that $G$ proves $\Gamma \vdash_\Sigma M : A$. If $M$ is a $\beta$-normal form, then $G\setminus\text{cut}$ proves $\Gamma \vdash_\Sigma M : A$.

**Proof** By Theorem 3.4, if $G$ proves $\Gamma \vdash_\Sigma M : A$ then $N^\oplus$ proves $\Gamma \vdash_\Sigma M : A$. By Lemma 3.7 and Theorem 3.9, if $M$ is a $\beta$-normal form then $G\setminus\text{cut}$ proves $\Gamma \vdash_\Sigma M : A$. $\square$

An alternative way to obtain the result of Corollary 3.10 is to prove the completeness of $G\setminus\text{cut}$ with respect to $G$ directly. The focus of the proof is then the need to eliminate instances of the cut rule; it is similar in character to the proof of [11].

4 Calculi for Proof-search

4.1 Proof-search in the $\lambda\Pi$-calculus

In proof-search, the basis of logic programming and automatic theorem proving, one reads the inference rules as reduction operators from conclusion to premisses. Thus, for a given endsequent, one searches for a proof by applying the inference rules, considered as reduction operators, until one has constructed a tree whose leaves are all axioms.

[Kleene [17] explains this in the case of predicate logic.]
In this, we consider an application of the cut-free calculus to proof-search in the $\lambda\Pi$-calculus. For proof-search we are interested not in the decidable typing assertions $\Gamma \vdash_\Sigma M : A$, in which the object $M$ encodes the proof of the assertion, but rather in semi-decidable inhabitation assertions of the form $\Gamma \Rightarrow_\Sigma A$, which is to be read as "(the type) $A$ is inhabited relative to the signature $\Sigma$ and context $\Gamma$": an inhabiting object, corresponding to a proof, remains to be calculated (and may not exist).

There are two candidates for a calculus for manipulating such sequents. The first is obtained from the natural deduction calculus $N$ by deleting inhabiting objects. In such a calculus, the $\Pi$-elimination rule has the form:

$$
\Pi E \quad \frac{\Gamma \Rightarrow_\Sigma \Pi x : A \cdot B}{\Gamma \Rightarrow_\Sigma B[N/x]}
$$

(a) $N$ proves $\Gamma \vdash_\Sigma N : A$,

which is similar to the usual natural deduction rule for the first-order universal quantifier. Since the type $A$ in the premise is not a subformula of the conclusion, a proof-search procedure based on such a calculus would have to invent the type.

The second candidate is obtained from the sequent calculus $G \setminus \text{cut}$ by deleting inhabiting objects and by building in contractions on the principal formulae of the left rules, in the manner described in § 1. This gives rise to a unary ($\Pi L$) rule of the form:

$$
\Pi L \quad \frac{\Gamma, y : D \Rightarrow_\Sigma C}{\Gamma \Rightarrow_\Sigma C}
$$

(a) $\exists : \Pi x : A \cdot B \in \Sigma \cup \Gamma$

(b) $y \notin \text{Dom}(\Gamma)$, $\text{FV}(C)$

(c) $G \setminus \text{cut}$ proves $\Gamma \vdash_\Sigma N : A$

(d) $B[N/x] \rightarrow_{\rho_y} D$.

This rule is preferable to the elimination rule for proof-search because, for a given choice of principal formula, it restricts the non-determinism to the choice of object $N$, and such choices can be calculated by unification. Note also that the principal formula, $\exists : \Pi x : A \cdot B$, occurs in both the premise as well as in the conclusion. An important feature [26, 28] of this calculus, $L$, is that the $(\rightarrow L)$ rule is not a special case of the ($\Pi L$) rule (unlike in the system $G$) but is the binary rule

$$
\rightarrow L \quad \frac{\Gamma \Rightarrow_\Sigma A \quad \Gamma, y : B \Rightarrow_\Sigma C}{\Gamma \Rightarrow_\Sigma C}
$$

(a) $\exists : A \rightarrow B \in \Sigma \cup \Gamma$

(b) $y \notin \text{Dom}(\Gamma)$, $\text{FV}(C)$.

This leads us to a minor remark on our presentation. Consider the following three equivalences, proved in the sequel:

\[
\begin{array}{c}
N \\
\downarrow^{(1)} \\
(3) \\
(1) \\
\downarrow \\
L. \\
\downarrow^{(2)} \\
G \setminus \text{cut}
\end{array}
\]

Although the fact that, in the system $G$, the $(\rightarrow L)$ rule is a special case of the ($\Pi L$) rule means that for the purpose of proving equivalence (1) it would be sufficient to consider the system without the type constructor $\rightarrow$, thereby eliminating a case in the proof, we include $\rightarrow$ for this proof, despite
the consequent need to introduce a notational device in order to handle the presence of both → and ∏. We do so for the following reason: an alternative to obtaining equivalence (3) via equivalences (1) and (2) is to obtain it directly: however, in so doing it is necessary to use the notational device we introduce in the proof of equivalence (1), or something of equivalent utility. In fact, it could be argued that we should pass directly from the system N to the system L, since it is L that is a metacalculus for N, in the sense in which L3 is a metacalculus for NJ [11, 26, 28]. For a direct proof of equivalence (3) it is necessary to consider → and ∏ separately; indeed, such a proof can be read off from our proof of equivalence (1) (Theorems 3.6 and 3.9). It is our judgement that our approach represents an efficient presentation of the content §§ 3 and 4.

In § 4.2, we discuss briefly how the calculus L can be developed to exploit a notion of indeterminate in order to facilitate the integration of search in L and unification, thereby leading to the analyses of [26, 23, 25, 24, 28]. By identifying the class of clausal types, we are able, in §§4.2 and 4.3, to give (analytic) resolution calculi, for both the typing and inhabitation relations.

4.2 A Metacalculus

Following on from § 4.1, we proceed to define a sequent as a triple (Σ, Γ, A), written Γ ⇒Σ A, where Σ is a signature, Γ is a context and A is a type (family). The intended interpretation of a sequent is the following (meta-)assertion:

(∃M) N proves Γ ⇒Σ M ▸ A.

The calculus L, given in Table 3, provides an adequate basis for searching for proofs of sequents of this form. Note that we restrict our attention to inhabitation assertions for types of kind Type: the calculus is defined only at this level of the λII-calculus (but see the remarks around rule (29), § 3.1).10

Here (for simplicity) we work exclusively with types that are β(η)-normal forms, and for such terms syntactic identity (≡) is taken up to α-congruence (change of bound variable). As usual we refer to the variable x of the (ΠR) rule as the eigenvariable of the inference. We can ensure that in any derivation eigenvariables occur only in sequents above the inference at which they are introduced. Both distinguished occurrences of A are said to be the principal formula of the (AX1) and (AX2) rules. A → B is the principal formula of the (→ R) and (→ L) rules and (Πz:A.B) is the principal formula of the (ΠR) and (ΠL) rules. A and B are the side formulae of the (→ R)

10This is a very natural restriction for the use made of the λII-calculus in LF, q.v. [13, 14, 2, 23, 28].
and (ILR) rules, and $A$ and $B[N/x]$ are the side formulae of the (ILL) rule. $A$, $B$ and $C$ are the side formulae of the ($\to L$) rule. $\emptyset$ is the principal atom in the ($\to L$) and (ILR) rules. $L$-derivations are trees of sequents regulated by the operational rules, and $L$-proofs are derivations whose leaves are axioms. A sequent $\Gamma \Rightarrow \Sigma \ A$ is said to be well-formed just in case $G \setminus \text{cut}$ proves $\Gamma \vdash \Sigma \ A$. Type. From Theorem 2.1 we have:

**Proposition 4.1** The well-formedness problem for sequents is decidable.

For proof-search, derivations are constructed from the endsequent (or root), toward the leaves, in the spirit of Kleene [17] and systems of tableaux [33]. In support of this usage we have the following result:

**Proposition 4.2** For well-formed sequents $\Gamma \Rightarrow \Sigma \ A$,

$$L \text{ proves } \Gamma \Rightarrow \Sigma \ A \quad \text{iff} \quad (\exists M) \quad G \setminus \text{cut proves } \Gamma \vdash \Sigma \ M : A. \quad \square$$

We revert to the system $G \setminus \text{cut}$ to decide if the endsequent is well-formed. If so, $L$ can be used to prove inhabitation of $A$ with respect to the context $\Gamma$. Moreover the inhabiting object $M$ is obtained as the extract-object of the $L$-proof $\Gamma \Rightarrow \Sigma \ A$, constructed by replacing the inhabiting objects in the proof, starting with the constants and variables at the leaves, in the manner of the construction of objects by the system the system $G \setminus \text{cut}$, q.v., [23, 28]. It should now be clear that the equivalence of the systems $L$ and $N$ (equivalence (3) of § 4.1) can be obtained directly from the proofs of Theorems 3.6 and 3.9.

We remark that a number of properties of $L$ are important for its use as the basis for proof-search in the $\lambda I$-calculus, although a detailed discussion of the theory of proof-search is beyond the scope of this paper (see [26, 27, 28]): (i) $L$ has the subformula property (recall the discussion of § 4.1); (ii) $L$ is semi-logicistic: the object $M$ required by side condition (c) of the (ILL) rule, the only appeal to a system external to $L$, can be calculated by unification; (iii) Uniform proofs can be defined: whilst the extract-objects of $L$-proofs are always $\beta$-normal forms, they need not also be $\eta$-normal forms. The requirement that an extract-object of an $L$-proof, $\Phi$, be a long $\eta$-normal form amounts to the requirement that $\Phi$ be a uniform proof in the sense of Miller et al. [20]. Such proofs are constructed, from root leaves, according to the strategy that left rules are applied only when no right rule is applicable [20, 23, 28]. To understand how this works, consider the following figure, an example of a uniform $L$-proof (of the axiom sequent $x : A \to B \Rightarrow \Sigma \ A \to B$ (sic)):

$$
\begin{array}{c}
x : A \to B, y : A \Rightarrow \Sigma \ A \\
x : A \to B, y : A \Rightarrow \Sigma \ A
\end{array}
$$

It is easy to see that the extract-object of this proof is $\lambda y : A. x y$, which is the long $\beta \eta$-normal form of $x$. Similar phenomena have also been observed by Felty [10].

We conclude our brief discussion of the application of the Gentzen-style calculus to proof-search by introducing a new calculus, $U$ which allows unification [24, 23, 28, 9] to be used to calculate terms for use with (ILL) inferences (point (ii), above). The calculation is constrained by the propositional structure of the derivation in such a way that only terms that are relevant to the formation of a proof (rather than a mere derivation) are considered. The search space of $U$ is then shown to be a proper subspace of the search space of $L$ containing representatives for all proofs in the latter. $U$-based search is therefore a complete and uniform improvement over $L$-based search. In order to define $U$, we introduce a new syntactic class of variables called indeterminates, denoted by lowercase Greek letters $\alpha, \beta, \text{ etc.}$, and extend the syntactic category of objects to include them thus:

$$
\text{Objects } \quad M := c \mid \alpha \mid \pi \mid \lambda x : A. M \mid MN.
$$

[11] Long $\eta$-normal forms have been studied carefully by Dowek [8].
Notice that indeterminates cannot appear λ-bound. By virtue of this extension, entities of all syntactic classes can now contain indeterminates as subterms. We define U by dropping the axiom schemata of L (they are recovered computationally via unification) and modifying the (ILL) rule as follows: the rules of U consist of the (→ R), (→ L) and (ΠR) rules of L, together with the rule

\[ \Pi L \quad \frac{\Gamma, z : B \vdash \alpha / x \Rightarrow \Sigma \ C}{\Gamma \Rightarrow \Sigma \ C} \]

(a) @ : Πx : A.B ∈ Σ ∪ Γ
(b) z \notin \text{Dom}(Γ), \text{FV}(C).

U-derivations are trees regulated by the rules of U such that the sequent at the root of the tree is well-formed. The calculus U is sound and complete for the calculation of L-proofs, although the proof of this result, and indeed the use of U for search, are complex and beyond the scope of the present paper. The theory of the calculus U is developed in [26, 28].

We conclude this section by remarking that the calculus U can be considered to provide the basis for logic programming with the λΠ-calculus, the indeterminates amounting to logical variables [5]. The development of both the syntactic and semantic theories of this notion of logic programming is beyond the scope of this paper. Pfenning [21] has also studied logic programming for the λΠ-calculus.

4.3 Resolution Calculi

In this section, we introduce a clausal form for both types and for sequents. This form is of particular interest: if we restrict our attention to sequents in this clausal form we are able to obtain a completeness theorem, with respect to the calculus L, for two calculi in which the (→ L) and (ILL) rules are replaced by rules which can be considered to be analytic counterparts to the resolution rule found in (classical) Horn clause logic [29]. This form is suggested by the form of Martin-Löf's hypothetico-general judgements [18] and by the encoding of logical rules in the LP [13, 14, 2]; a thorough discussion of its use in the formal definition of natural deduction inference rules is beyond the scope of this paper. In LP, clausal types can be considered to be the type-theoretic counterparts of the rule schemes of [1, 31]. The encoding of classes of introduction and elimination rules in LP is discussed in [23, 28].

A type (of kind Type, well-typed in some signature and context) is said to be in clausal form if it is of the following, recursively defined, form:

\[ \Pi x_1 : B_1 \ldots \Pi x_m : B_m \cdot C_1 \rightarrow (C_2 \rightarrow (\ldots (C_n \rightarrow D)\ldots)) \],

(43)

where D is atomic (i.e., of the form cM₁ \ldots M_p) and each B_i, 1 ≤ i ≤ m, and each C_j, 1 ≤ j ≤ n, is in clausal form.

We are now able to define clausal form for sequents: let \( \Gamma \Rightarrow \Sigma \ A \) be a well-formed sequent. The sequent is said to be in clausal form if A is in clausal form and for each \( @ : B \in \Sigma \cup \Gamma \), B is in clausal form.

We note that any type (of kind Type, well-typed in some signature and context) can be put into clausal form. To see this, consider the type

\[ E \equiv \Pi x : A \cdot \Pi y : B \cdot (C_1 \rightarrow (\Pi z : D \cdot (C_2 \rightarrow D))) \]

It is easy to see that any proof which uses some \( @ : E \) can be replaced by a proof which uses some \( @' : E' \) instead, where

\[ E' \equiv \Pi x : A \cdot \Pi y : B \cdot \Pi z : D \cdot (C_1 \rightarrow (C_2 \rightarrow D)) \],

since \( z \notin \text{FV}(C_1) \), and that any proof which uses \( @' : E' \) can be replaced by a proof which uses \( @ : E \) instead.\(^\text{12}\)

\(^{12}\)Note that all of the types declared in the signatures given in [2] are in clausal form.
4.4 A Resolution Calculus

We introduce the first of our two resolution rules. This is a single rule which combines many instances of the \( (\rightarrow L) \) and \( (\Pi L) \) rules of \( G \setminus \text{cut} \), provided that all types and sequents are in clausal form:

\[
\Gamma \vdash_{\Sigma} M_1 : A_1 \ldots \Gamma \vdash_{\Sigma} M_m : A_m \quad \Gamma \vdash_{\Sigma} N_1 : D_1 \ldots \Gamma \vdash_{\Sigma} N_n : D_n,
\]

\[
\Gamma \vdash_{\Sigma} \ominus M_1 \ldots M_m N_1 \ldots N_n : E
\]

where \( \ominus : \Pi \Delta_1 : E_1 \ldots \Pi \Delta_m : E_m \cdot B_1 \rightarrow (B_2 \rightarrow \ldots \rightarrow (B_n \rightarrow C) \ldots) \in \Sigma \cup \Gamma \), for \( 1 \leq i \leq m \) \( A_i[M_1/x_1, \ldots, M_{i-1}/x_{i-1}] = \beta_i E_i \), for \( 1 \leq i \leq n \) \( B_i[M_1/x_1, \ldots, M_m/x_m] = \beta_i D_i \) and \( C[M_1/x_1, \ldots, M_m/x_m] = \beta_i E \).

We define the system \( NR \) to be that system which is obtained from the system \( N \) by replacing the \( (\rightarrow E) \) and \( (\Pi E) \) rules by the rule (44).

In fact, the rule (44) has the additional trivial premise \( \Gamma, x : E \vdash_{\Sigma} x : E \), where \( x \notin \text{Dom}(\Gamma) \), but for simplicity of presentation we omit to write this premise.

**Theorem 4.3 (Soundness of NR)** If \( NR \) proves \( \Gamma \vdash_{\Sigma} M : A \) then \( N \) proves \( \Gamma \vdash_{\Sigma} M : A \).

**Proof** A straightforward induction on the structure of proofs. We omit the details. \( \Box \)

**Theorem 4.4 (Completeness of NR)** Let \( M \) be a \( \beta \)-normal form. If \( N \) proves \( \Gamma \vdash_{\Sigma} M : A \) then \( NR \) proves \( \Gamma \vdash_{\Sigma} M : A \).

**Proof** The proof proceeds by induction on the structure of proofs in \( N \). The only interesting cases are where the last rule of \( N \) applied is either the \( (\rightarrow E) \) rule or the \( (\Pi E) \) rule. We sketch the argument.

If the last rule of \( N \) applied is the \( (\rightarrow E) \) rule, and if the conclusion of this rule is a \( \beta \)-normal form, then the last inference in \( N \) is of the form

\[
\Gamma \vdash_{\Sigma} \ominus M_1 \ldots M_m N_1 \ldots N_n : A \rightarrow B \quad \Gamma \vdash_{\Sigma} N : A
\]

where \( \ominus : \Pi \Delta_1 : A_1 \ldots \Pi \Delta_m : A_m \cdot B_1 \rightarrow (B_2 \rightarrow \ldots \rightarrow (B_n \rightarrow (A^* \rightarrow B^*))(\ldots)) \in \Sigma \cup \Gamma \) and \( (A^* \rightarrow B^*)[M_1/x_1, \ldots, M_m/x_m] = \beta_n A \rightarrow B \). By induction on the structure of proofs in \( N \) we have that \( NR \) proves \( \Gamma \vdash_{\Sigma} N : A \) and by induction on the structure of proofs that \( NR \) proves \( \Gamma \vdash_{\Sigma} M_i : E_i \), where \( A_i[M_1/x_1, \ldots, M_{i-1}/x_{i-1}] = \beta_i E_i \), for \( 1 \leq i \leq m \), and that \( NR \) proves \( \Gamma \vdash_{\Sigma} N_i : D_i \), where \( B_i[M_1/x_1, \ldots, M_m/x_m] = \beta_i D_i \), for \( 1 \leq i \leq n \). By an argument which is similar to that used in the proof of Theorem 3.9, we can prove that \( NR \) proves \( \Gamma, x : B^*[M_1/x_1, \ldots, M_m/x_m] \vdash_{\Sigma} x : B^*[M_1/x_1, \ldots, M_m/x_m] \), where as in the proof of Theorem 3.9 the difficulty is to prove that \( NR \) proves \( \Gamma \vdash_{\Sigma} B^*[M_1/x_1, \ldots, M_m/x_m] : \text{Type} \).

We now obtain \( NR \) proves \( \Gamma \vdash_{\Sigma} \ominus M_1 \ldots M_m N_1 \ldots N_n : B \) by rule (44).

The case in which the last rule of \( N \) applied is the \( (\Pi E) \) is similar to that for the \( (\rightarrow E) \) rule. \( \Box \)
A version of the rule (44) can be given for the inhabitation relation over a given signature, \( \Rightarrow_{\Sigma} \). Consider the following rule:

\[
\frac{\Gamma \Rightarrow_{\Sigma} D_1 \ldots \Gamma \Rightarrow_{\Sigma} D_n}{\Gamma \Rightarrow_{\Sigma} E}
\]  

(45)

where \( \otimes : \Pi x_1 : A_1 \ldots \Pi x_m : A_m \cdot B \rightarrow (B_2 \rightarrow (\ldots \rightarrow (B_n \rightarrow C) \ldots )) \in \Sigma \cup \Gamma \),

\( \Phi \) proves \( \Gamma \vdash_{\Sigma} M_i : E_i \), where, for \( 1 \leq i \leq m \), \( A_i[M_1/x_1, \ldots, M_{i-1}/x_{i-1}] \Rightarrow_{\beta} E_i \),

for \( 1 \leq i \leq n \), \( B_i[M_1/x_1, \ldots, M_m/x_m] \Rightarrow_{\beta} D_i \) and \( C[M_1/x_1, \ldots, M_m/x_m] \Rightarrow_{\beta} E \).

In fact, the rule (45) has the additional trivial premise \( \Gamma, x : E \Rightarrow_{\Sigma} E \), where \( x \notin \text{Dom}(\Gamma) \), but for simplicity of presentation we omit this premise. The importance of this premise is now rather clear: if \( n = 0 \) it is the only premise that is obtained.

Let \( \mathbf{LNR} \) be that calculus which is obtained from the calculus \( \mathbf{L} \) by replacing the \((\rightarrow L)\) and \((\Pi L)\) rules by the rule (45). It is easy to see that if the extract-object of the conclusion of the rule (45) is defined to be \( \otimes M_1 \ldots M_m N_1 \ldots N_n \) where, for \( 1 \leq i \leq n \), \( N_i \) is the extract-object of the premise \( \Gamma \Rightarrow_{\Sigma} D_i \), then \( \mathbf{LNR} \) is sound and complete with respect to \( \mathbf{NR} \) in the following sense:

**Proposition 4.5 (Soundness and Completeness of \( \mathbf{LNR} \))** \( \Phi \) is an \( \mathbf{LNR} \) proof of \( \Gamma \Rightarrow_{\Sigma} A \) if and only if \( \mathbf{NR} \) proves \( \Gamma \vdash_{\Sigma} M : A \), where \( M \) is the extract-object of \( \Phi \). \( \square \)

It follows immediately that \( \mathbf{LNR} \) is sound and complete with respect to \( \mathbf{L} \).

Note that the systems \( \mathbf{NR} \) and \( \mathbf{LNR} \) are syntax-directed in two senses: (i) the left rule is driven by a choice of clausal type, the principal formula of the rule, from the context (corresponding, in logic programming terms, to the choice of program clause, for which it is desirable that the rule builds in a contraction of the principal formula, thereby permitting the re-use of the clause); and (ii) the rightmost (atomic) type matches with the succedent (right hand side) of the sequent. In this sense, the rules (44) and (45) can be thought of as being both natural deduction rules and Gentzen-style rules. This leads us to conclude the paper with a brief section on a resolution calculus which is not syntax-directed, but which resides properly in the world of Gentzen-style calculi.\(^{13}\)

### 4.5 Another Resolution Calculus

We introduce the second of our two resolution rules; again we suppose that all types and sequents are in clausal form.

\[
\frac{\Gamma \Rightarrow_{\Sigma} M_1 : E_1 \ldots \Gamma \Rightarrow_{\Sigma} M_m : E_m \quad \Gamma \Rightarrow_{\Sigma} N_1 : D_1 \ldots \Gamma \Rightarrow_{\Sigma} N_n : D_n \quad \Gamma \vdash_{\Sigma} P : F \quad x \notin \text{FV}(F)}{\Gamma \Rightarrow_{\Sigma} P[\otimes M_1 \ldots M_m N_1 \ldots N_n/x] : F}
\]  

(46)

where \( \otimes : \Pi x_1 : A_1 \ldots \Pi x_m : A_m \cdot B \rightarrow (B_2 \rightarrow (\ldots \rightarrow (B_n \rightarrow C) \ldots )) \in \Sigma \cup \Gamma \),

for \( 1 \leq i \leq m \), \( A_i[M_1/x_1, \ldots, M_{i-1}/x_{i-1}] \Rightarrow_{\beta} E_i \), for \( 1 \leq i \leq n \), \( B_i[M_1/x_1, \ldots, M_m/x_m] \Rightarrow_{\beta} D_i \) and \( C[M_1/x_1, \ldots, M_m/x_m] \Rightarrow_{\beta} E \).

We define the system \( \mathbf{GR} \) to be that which is obtained from the system \( \mathbf{N} \) by replacing the \((\rightarrow E)\) and \((\Pi E)\) rules by the rule (46).

The system \( \mathbf{GR} \) can be considered to stand in the same relation to the system \( \mathbf{NR} \) as does the system \( \mathbf{G} \) to the system \( \mathbf{N} \).

**Theorem 4.6 (Soundness of \( \mathbf{GR} \))** If \( \mathbf{GR} \) proves \( \Gamma \vdash_{\Sigma} M : A \) then \( \mathbf{N} \) proves \( \Gamma \vdash_{\Sigma} M : A \).

**Proof** A straightforward induction on the structure of proofs. We omit the details. \( \square \)

\(^{13}\)Perhaps such a calculus should not be called a resolution calculus at all, but its relation to the calculus of the present section will be clear.
THEOREM 4.7 (COMPLETENESS OF GR) Let $M$ be a $\beta$-normal form. If $N$ proves $\Gamma \vdash_{\Sigma} M : A$ then GR proves $\Gamma \vdash_{\Sigma} M : A$.

PROOF The proof proceeds by induction on the structure of proofs in $N$. The only interesting cases are where the last rule of $N$ applied is either the ($\rightarrow E$) rule or the (IIE) rule. We sketch the argument.

If the last rule of $N$ applied is the (IIE) rule, and if the conclusion of this rule is a $\beta$-normal form, then the last inference in $N$ is of the form

\[
\frac{\Gamma \vdash_{\Sigma} \Pi x : A \cdot B \quad \Gamma \vdash_{\Sigma} N : A}{\Gamma \vdash_{\Sigma} \Pi x : A \cdot M \cdot N \cdot B[N/x]},
\]

where $\Pi : \Pi x_{1} : A_{1} \ldots \Pi x_{m} : A_{m} \cdot B_{1} \rightarrow (B_{2} \rightarrow (\ldots \rightarrow (B_{n} \rightarrow (\Pi x : A^{*} \cdot B^{*}) \ldots))) \in \Sigma \cup \Gamma$ and $[\Pi x : A^{*} \cdot B^{*}][M_{1}/x_{1}, \ldots, M_{m}/x_{m}] =_{\beta_{n}} \Pi x : A \cdot B$. By induction on the structure of proofs in $N$ we have that GR proves $\Gamma \vdash_{\Sigma} N : A$ and by induction on the structure of proofs that GR proves $\Gamma \vdash_{\Sigma} M_{i} : E_{i}$, where $A_{i}[M_{1}/x_{1}, \ldots, M_{m-1}/x_{m-1}] =_{\beta_{n}} E_{i}$, for $1 \leq i \leq m$, and that GR proves $\Gamma \vdash_{\Sigma} N_{i} : D_{i}$, where $B_{i}[M_{1}/x_{1}, \ldots, M_{m}/x_{m}] =_{\beta_{n}} D_{i}$, for $1 \leq i \leq n$. By an argument similar to that used in the proof of Theorem 3.9 we can prove that

GR proves $\Gamma, x : B^{*}[M_{1}/x_{1}, \ldots, M_{m}/x_{m}, N/x] \vdash_{\Sigma} x : B^{*}[M_{1}/x_{1}, \ldots, M_{m}/x_{m}, N/x]$.

where, as in the proof of Theorem 3.9, the difficulty is to prove that

GR proves $\Gamma \vdash_{\Sigma} B^{*}[M_{1}/x_{1}, \ldots, M_{m}/x_{m}, N/x] : Type$.

We now obtain GR proves $\Gamma \vdash_{\Sigma} \Pi M_{1} \ldots \Pi M_{m} N_{1} \ldots N_{n} N : B$ by rule 46.

The case in which the last rule of $N$ applied is the ($\rightarrow E$) rule is similar to that for the (IIE) rule. $\square$

As in the case of the first resolution rule, a version of the rule (46) can be given for the inhabitation relation over a given signature, $\Rightarrow_{\Sigma}$. Consider the following rule:

\[
\frac{\Gamma \Rightarrow_{\Sigma} D_{1} \ldots \Gamma \Rightarrow_{\Sigma} D_{n} \quad \Gamma, x : E \Rightarrow_{\Sigma} F}{\Gamma \Rightarrow_{\Sigma} F},
\] (47)

where $\Pi : \Pi x_{1} : A_{1} \ldots \Pi x_{m} : A_{m} \cdot B_{1} \rightarrow (B_{2} \rightarrow (\ldots \rightarrow (B_{n} \rightarrow C) \ldots)) \in \Sigma \cup \Gamma$, $\Gamma \\text{cut}$ proves $\Gamma \vdash_{\Sigma} M_{i} : E_{i}$, where $A_{i}[M_{1}/x_{1}, \ldots, M_{m-1}/x_{m-1}] =_{\beta_{n}} E_{i}$ for $1 \leq i \leq m$, $B_{i}[M_{1}/x_{1}, \ldots, M_{m}/x_{m}] =_{\beta_{n}} D_{i}$, for $1 \leq i \leq n$, and $C[M_{1}/x_{1}, \ldots, M_{m}/x_{m}] =_{\beta_{n}} E$.

Let LGR be that calculus which is obtained from the calculus $L$ by replacing the ($\rightarrow L$) and (ILL) rules by the rule (47). It is easy to see that if the extract-object of the conclusion of the rule (47) is defined to be $P[\Pi M_{1} \ldots \Pi M_{m} N_{1} \ldots N_{n}/x]$ where, for $1 \leq i \leq n$, $N_{i}$ is the extract-object of the premiss $\Gamma \Rightarrow_{\Sigma} D_{i}$ and where $P$ is the extract-object of the premiss $\Gamma, x : E \Rightarrow_{\Sigma} F$, then LGR is sound and complete with respect to GR in the following sense:

PROPOSITION 4.8 (SOUNDNESS AND COMPLETENESS OF LGR) $\Phi$ is an LGR $\Gamma \Rightarrow_{\Sigma} A$ if and only if GR proves $\Gamma \vdash_{\Sigma} M : A$, where $M$ is the extract-object of $\Phi$. $\square$

It follows immediately that LGR is sound and complete with respect to L.

We note that we have the obvious notion of uniform LGR-proof and remark that the strategy of constructing uniform LGR-proofs is complete.

5 Acknowledgments

This paper is largely based on Chapters 3 and 4 of the author's University of Edinburgh Ph.D. thesis [23, 28]. The author is grateful to his Ph.D. supervisors, Gordon Plotkin and Furio Honsell,
his Ph.D. examiners, Alan Bundy and Gérard Huet, and to Arnon Avron, Robert Harper, Ian Mason, Anne Salvesen and especially Lincoln Wallen for their help and advice. Furthermore, the author is grateful to the three anonymous referees for their assistance in the production of this published form of the paper.

References


