Strong cut-elimination systems for Hudelmaier's depth-bounded sequent calculus for implicational logic

Roy Dyckhoff[1], Delia Kesner[2], and Stéphane Lengrand[1,2]

[1] PPS, CNRS and Université Paris 7, France
[2] School of Computer Science, University of St Andrews, Scotland

Abstract. Inspired by the Curry-Howard correspondence, we study *normalisation* procedures in the depth-bounded intuitionistic sequent calculus of Hudelmaier (1988) for the implicational case, thus strengthening existing approaches to *Cut*-admissibility. We decorate proofs with proof-terms and introduce various term-reduction systems representing proof transformations. In contrast to previous papers which gave different arguments for *Cut*-admissibility suggesting *weakly normalising* procedures for *Cut*-elimination, our main reduction system and all its variations are *strongly normalising*, with the variations corresponding to different optimisations, some of them with good properties such as confluence.

1 Introduction

The sequent calculus G4ip (as it is called in [TS00]) for intuitionistic propositional logic was independently developed by Hudelmaier [Hud89,Hud92], and the first author [Dyc92]; see also Lincoln, Scedrov & Shankar [LSS91]; it has the strong property of being *depth-bounded*, in that proofs are of bounded depth and thus (for root-first proof search) no loop-checking is required. This contrasts with other calculi for this logic such as Kleene's G3ip, where proofs can be of unbounded depth. Its essential ingredients appeared already in 1952 work of Vorob'ev, published in detail in [Vor70].

Its completeness can be shown by various means, either indirectly, using the completeness of another calculus and a permutation argument [Dyc92], or directly, such as in the work of Negri and the first author [DN00] where cut-admissibility is proved without reference to the completeness of any other sequent calculus. This admissibility proof could be seen, via the Curry-Howard correspondence, as a *weakly normalising* proof-reduction system. Developing this idea, this paper presents a formulation of implicational G4ip with derivations represented by (proof-)terms; strong (instead of weak) normalisation is proved by the use of a multi-set path ordering. Several variations, *all of them being strongly normalising*, are considered, depending on whether we want to have a system as general as possible or a system more restricted (but simpler) implementing some reduction strategy.

The merits of G4ip for proof-search and automated reasoning have been discussed in many papers (see [ORK05] for some recent pointers; note its use of an old name LJT for G4ip), because the property of being depth-bounded makes the space of derivations of a given sequent finite.

However, a question that has been less investigated, natural though it is, is the following: which proofs are produced by proof-search in G4ip and what are their properties? Our approach to cut-elimination in this paper, with a strongly normalising reduction system internal to G4ip, tackles this question in terms of the behaviour of these proofs when they are combined together with cuts. In other words, we give them an operational semantics.

A complementary approach is to give these proofs a denotational semantics and to relate them (and their reductions) to simply-typed λ-terms. We leave this approach for future work.

In contrast to previous work, this paper presents G4ip with a proof-term syntax, so sequents are of the form $\Gamma \Rightarrow M : A$ where A is a type, M is a (proof-)term and Γ is a consistent finite set of "declarations" of the form $x : B$, where x is a variable and B a type. Results about such sequents translate directly to results about traditional "logical sequents".

Our approach to cut-elimination using proof-terms differs from that in [DN00], which showed (in the context of logical sequents) first the admissibility of *Contraction* and then the admissibility of "context-splitting" (or "multiplicative") *Cut*. Given our interest in the term calculi, it is appropriate to use rather a "context-sharing" *Cut*; admissibility of *Contraction* then follows as a special case of that of *Cut*.

Matthes [Mat02] tackled a similar problem, with a variety of motivations, such as that of understanding better Pitts' algorithm [Pit92] for uniform interpolation; but his approach has not yet been brought to a successful conclusion. His work is similar to ours in using terms to represent derivations; but it differs conceptually from ours by considering not the use of explicit operators to encode the *Cut*-rule but the closure of the syntax under (implicit) substitution, as in pure λ-calculus, where the general syntax of λ-terms may be considered as the extension of the normal lambda terms by such an implicit closure. His reduction rules are global (using implicit substitutions) rather than local (using explicit operators); strong normalisation is shown for a subset of the reductions, but not for all that are required.

Structure of the paper The paper is organised as follows. Section 2 presents the term syntax and typing rules of our calculus for G4ip and its auxiliary (admissible) rules. Section 3 studies proof transformations and reduction rules of the calculus. Section 4 shows a translation from the calculus to a first-order syntax and Section 5 shows that every reduction step satisfies subject reduction and decreases first-order terms associated to derivations with respect to a multi-set path ordering, thus proving strong normalisation. In Section 6 we give different variants for the reduction system introduced in Section 3, some of them being confluent. Finally we conclude and give some ideas for further work.

2 Syntax

2.1 Grammar

We assume we are given an infinite set of *base types* P (known as *proposition variables* or *atomic formulae* in the logical interpretation) and an infinite set of *variables* x. We consider the following grammars for *types* (also known as *formulae*) and *terms*:

Definition 1 (Grammar of Types and Terms).

$$\begin{array}{lll} A, B, C, D, E, F & ::= & P \mid A \supset B \\ M, N, L & ::= & x \mid \lambda x.M \mid x(y, z.M) \mid x(u.v.M, z.N) \mid \\ & & \mathtt{inv}(x, y.M) \mid \mathtt{of}(M, x) \mid \mathtt{dec}(x, y, z.M) \mid \mathtt{cut}(M, x.N) \end{array}$$

In this definition, the first line defines the syntax for types, the second gives the syntax for *normal* or *constructor* terms (corresponding to primitive derivations) and the third gives the extra syntax for *auxiliary* terms, which may be built up using also the "auxiliary constructors" that appear in bold teletype font, such as `cut`. Six of the eight term constructors use variable binding: in $\lambda x.M$, x binds in M; in $x(y, z.M)$, z binds in M; in $x(u.v.M, z.N)$, u and v bind in M and z binds in N; in $\mathtt{inv}(x, y.M)$, y binds in M; in $\mathtt{dec}(x, y, z.M)$, z binds in M; and in $\mathtt{cut}(M, x.N)$, x binds in N.

Standard conventions are used to avoid confusion of free and bound variables, and α-convertible terms are regarded as identical.

Certain constraints on the use of the term syntax will be evident once we present the typing rules; these constraints are captured by the following notion of *well-formed term*:

Definition 2. *A term L is* well-formed *if in any sub-term of the form*

- $x(y, z.M)$, *we have $x \neq y$, with x not free in M;*
- $x(u.v.M, z.N)$, *we have $u \neq v$, with x not free in M and not free in N;*
- $\mathtt{inv}(x, y.M)$, *we have x not free in M;*
- $\mathtt{of}(M, x)$, *we have x not free in M;*
- $\mathtt{dec}(x, y, z.M)$, *we have $x \neq y$, with both of them not free in M.*

Definition 3 (Ordering on (multi-sets of) types). *The* weight $w(A)$ *of a type A is defined by: $w(P) = 1$ for any base type P and $w(A \supset B) = 1 + w(A) + w(B)$. Types are compared by their weight, i.e. we say that A is* smaller *than B iff $w(A) < w(B)$.*

We shall then compare multi-sets of types, equipped with the traditional multi-set ordering *[DM79], denoted $<_{\mathsf{mul}}$, generated by the order relation on types.*

The weight is chosen to ensure that, for every rule of the logical sequent calculus **G4ip**, the multi-set of types appearing in the conclusion is greater than that of any given premiss. Hence, we say that **G4ip** is *depth-bounded*. See [Dyc92] or [TS00] for details, and see the next section for the corresponding property in our version of **G4ip** with proof-terms.

2.2 Typing

A *context* Γ is a consistent finite set of *declarations*, i.e. expressions $x{:}A$ (where x is a variable and A is a type) *declaring* x to be of type A; by *consistent* is meant that if $x{:}A$ and $x{:}B$ are in Γ, then $A = B$. When we write a context in the form $\Gamma, x{:}A$ it is always implicit that there is no declaration $x{:}B$ in Γ of the same variable x. By removing the variable names from a context Γ, but keeping the types, we obtain the multiset $m(\Gamma)$ of types that is *associated with* the context.

A *sequent* consists of a context Γ, a term M and a type A; it is written $\Gamma \Rightarrow M{:}A$.

The next definition adds term notation to the rules for implication of **G4ip**; another view is that it shows how the untyped normal terms of the above grammar may be typed.

Definition 4 (Typing Rules for Normal Terms).

$$\frac{}{\Gamma, x{:}A \Rightarrow x{:}A}\ Ax \qquad \frac{\Gamma, x{:}A \Rightarrow M{:}B}{\Gamma \Rightarrow \lambda x.M{:}A{\supset}B}\ R{\supset}$$

$$\frac{\Gamma, y{:}A, z{:}B \Rightarrow M{:}E}{\Gamma, x{:}A{\supset}B, y{:}A \Rightarrow x(y, z.M){:}E}\ L0{\supset}$$

$$\frac{\Gamma, u{:}C, v{:}D{\supset}B \Rightarrow M{:}D \qquad \Gamma, z{:}B \Rightarrow N{:}E}{\Gamma, x{:}(C{\supset}D){\supset}B \Rightarrow x(u.v.M, z.N){:}E}\ L{\supset}{\supset}$$

As remarked before these rules only construct well-formed terms; for example the notation $\Gamma, x{:}A{\supset}B, y{:}A$ in the conclusion of rule $L0{\supset}$ forces x to be not free in M and $x \neq y$.

Note that we use a slight variant of the $L{\supset}{\supset}$ rule used in [Dyc92] and [TS00], and that both in axioms $\Gamma, x{:}A \Rightarrow x{:}A$ and in the rule $L0{\supset}$ the type A need not be atomic. In the

rules $R{\supset}$, $L0{\supset}$ and $L{\supset}{\supset}$ the types $A{\supset}B, A{\supset}B$ and $(C{\supset}D){\supset}B$ respectively are *principal*; in $L0{\supset}$ the type A is *auxiliary*. (This use of "auxiliary" is not to be confused with its use in Definition 1 to describe certain kinds of term.)

Notice that in every instance of a rule in Definition 4 with conclusion $\Gamma \Rightarrow M : A$, each premiss $\Gamma' \Rightarrow N : B$ is such that $m(\Gamma) \cup A >_{\mathsf{mul}} m(\Gamma') \cup B$, where $\cup$ denotes the union of multi-sets. As a consequence, given Γ and A, there are finitely many derivations concluding $\Gamma \Rightarrow M : A$ for some (normal) term M.

De¯nition 5 (Typing Rules for Auxiliary Terms).

$$\frac{\Gamma, y{:}C{\supset}D \Rightarrow M : E}{\Gamma, x{:}D \Rightarrow \mathtt{inv}(x, y.M) : E}\ Inv \qquad \frac{\Gamma \Rightarrow M : A{\supset}B}{\Gamma, x{:}A \Rightarrow \mathtt{of}(M, x) : B}\ Of$$

$$\frac{\Gamma, z{:}(C{\supset}D){\supset}B \Rightarrow M : A}{\Gamma, x{:}C, y{:}D{\supset}B \Rightarrow \mathtt{dec}(x, y, z.M) : A}\ Dec \qquad \frac{\Gamma \Rightarrow M : A \quad x{:}A, \Gamma \Rightarrow N : B}{\Gamma \Rightarrow \mathtt{cut}(M, x.N) : B}\ Cut$$

As remarked before these rules only construct well-formed terms; for example the notation $\Gamma, x{:}A$ in the conclusion of rule Inv forces x to be not free in M.

In the Cut-rule, we say that A is the *cut-type*. *Derivations* are the labelled trees whose leaves are axioms and whose internal nodes match rules: each label is a sequent and each internal node is also labelled by the name of the rule. A derivation is *normal* if it uses only the primitive rules, i.e. those of Definition 4. The *height* of a derivation is just its height as a tree; so a tree with one node has height 0.

Remark 1. Notice that for each proved sequent $\Gamma \Rightarrow M : A$ there is a unique derivation tree, which can be reconstructed using the information of the term M that *represents* the proof (hence the notion of *proof-term*).

We will occasionally find it necessary to rename free variables. The *renaming* by the variable y of all the free occurrences of x in M, written $\{y/x\}M$, is defined whenever y and x are distinct variables, M is a well-formed term and y is not free in M.

This is an implicit operation on terms, not an explicit term constructor. In other words, renaming is a transformation of terms, and it is sound with respect to typing, as shown by the first of the two results of admissibility of Lemma 1. Admissibility is considered in the standard sense (see for instance [TS00]):

De¯nition 6. *A rule R is* admissible *in an inference system S if and only if, for each instance whose premisses are all derivable in S, the conclusion is also derivable in S.*

Lemma 1. *The following rules are admissible both in the system of normal derivations and in the full system with auxiliary terms, with the proviso that $y \neq x$ in the* (Ren) *rule.*

$$\frac{\Gamma, x{:}B \Rightarrow M : A}{\Gamma, y{:}B \Rightarrow \{y/x\}M : A}\ (Ren) \qquad \frac{\Gamma \Rightarrow M : A}{\Gamma, y{:}B \Rightarrow M : A}\ (W)$$

Proof: Routine induction on the height of the derivation of the premiss. Some swapping of bound variable names may be necessary: recall our convention about α-conversion and identity of terms. Remember that the notation $\Gamma, y{:}B$ forces y to be not free in M. □

We parenthesise the names of those two rules to indicate their admissibility.

3 Proof Transformations and Reduction Rules

The starting point of this section is the admissibility in the (cut-free) logical sequent calculus G4ip of the following inference rules (i.e. the logical counter-part of the typing rules for auxiliary terms given in Definition 5):

$$\frac{\Gamma, C{\supset}D \Rightarrow E}{\Gamma, D \Rightarrow E}\ Inv \qquad\qquad \frac{\Gamma \Rightarrow A{\supset}B}{\Gamma, A \Rightarrow B}\ Of$$

$$\frac{\Gamma, (C{\supset}D){\supset}B \Rightarrow A}{\Gamma, C, D{\supset}B \Rightarrow A}\ Dec \qquad\qquad \frac{\Gamma \Rightarrow A \qquad A, \Gamma \Rightarrow B}{\Gamma \Rightarrow B}\ Cut$$

The admissibility of Inv and Of in G4ip can be proved, independently, by induction on the height of the derivation. For the admissibility of Dec and Cut we can use a simultaneous induction, the admissibility of one rule being recursively used for the admissibility of the other. The measure is now the multi-set of types appearing in the unique premiss for Dec and in the second premiss for Cut. In other words, the induction can be done on $\{\!\{\Gamma, (C{\supset}D){\supset}B, A\}\!\}$ for Dec and on $\{\!\{\Gamma, A, B\}\!\}$ for Cut.

We do not include here the detail of those proofs of admissibility, because they become a corollary (Corollary 2) of the properties that we show for our calculus with proof-terms.

With proof-terms, those admissibility properties mean that a proof-term M with auxiliary constructors $\mathtt{inv}(\_, \_.\_)$, $\mathtt{of}(\_, \_)$, $\mathtt{dec}(\_, \_, \_.\_)$ or $\mathtt{cut}(\_, \_.\_)$ can be transformed into another proof-term M' with the same type in the same context that does not use these constructors.

This motivates the notion of *logical admissibility* in a system with proof-terms:

Definition 7. *A rule R is* logically admissible *in system S if, given an instance with conclusion $\Gamma \Rightarrow M{:}A$ and derivations in system S of its premiss(es), there exists a derivation in S of $\Gamma \Rightarrow M'{:}A$ for some proof-term M'.*

Remark that this notion corresponds to the standard notion of admissibility (Definition 6) when proof-term annotations are erased.

Indeed, the proofs of admissibility above can be seen as *weakly normalising* term reduction systems that specify how to eliminate the auxiliary constructors $\mathtt{inv}(\_, \_.\_)$, $\mathtt{of}(\_, \_)$, $\mathtt{dec}(\_, \_, \_.\_)$ and $\mathtt{cut}(\_, \_.\_)$.

The reduction systems, given hereafter, must satisfy the following properties:

1. A term containing an auxiliary constructor is reducible by these systems.
2. They satisfy the Subject Reduction property, i.e. preservation of typing.
3. They satisfy some termination property.

Concerning point 3, the weak normalisation property of these systems suffices to prove the results of admissibility, and the proofs suggested above can be expressed as a terminating innermost strategy for these reduction systems. Nevertheless, we give in this paper reduction systems that are in fact *strongly normalising*. While this can be inferred for the orthogonal systems that we present in Section 6 (since weak innermost normalisation is equivalent to strong normalisation for orthogonal systems [O'D77]), the result is not straightforward for the non-orthogonal ones. However, the measures for induction mentioned above can be taken as part of a *Multi-Set Path Ordering* [KL80,BN98] in order to conclude strong normalisation as well (see Section 4).

We now give in Tables 1, 2 and 3 the reduction systems that eliminate the auxiliary constructors $\mathtt{of}$, $\mathtt{inv}$ and $\mathtt{dec}$. All these rules that we call system oid will be part of the different variants that we are going to introduce.

$\mathtt{of}(y, x)$	$\longrightarrow_{\mathsf{o}1}$	$y(x, z.z)$
$\mathtt{of}(\lambda y.M, x)$	$\longrightarrow_{\mathsf{o}2}$	$\{x/y\}M$
$\mathtt{of}(y(z, w.N), x)$	$\longrightarrow_{\mathsf{o}3}$	$y(z, w.\mathtt{of}(N, x))$
$\mathtt{of}(y(u.v.M, w.N), x)$	$\longrightarrow_{\mathsf{o}4}$	$y(u.v.M, w.\mathtt{of}(N, x))$

Table 1. Reduction Rules for $\mathtt{of}$-terms

$\mathtt{inv}(x, y.z)$	$\longrightarrow_{\mathsf{i}1}$	z
$\mathtt{inv}(x, y.y)$	$\longrightarrow_{\mathsf{i}2}$	$\lambda z.x$
$\mathtt{inv}(x, y.\lambda z.M)$	$\longrightarrow_{\mathsf{i}3}$	$\lambda z.\mathtt{inv}(x, y.M)$
$\mathtt{inv}(x, y.y(w, z.N))$	$\longrightarrow_{\mathsf{i}4}$	$\{x/z\}N$
$\mathtt{inv}(x, y.y(u.v.M, z.N))$	$\longrightarrow_{\mathsf{i}5}$	$\{x/z\}N$
$\mathtt{inv}(x, y.w(y, z.N))$	$\longrightarrow_{\mathsf{i}6}$	$w(u.v.x, z.\mathtt{inv}(x, y.N))$
$\mathtt{inv}(x, y.y'(w, z.N))$	$\longrightarrow_{\mathsf{i}7}$	$y'(w, z.\mathtt{inv}(x, y.N))$
$\mathtt{inv}(x, y.y'(u.v.M, z.N))$	$\longrightarrow_{\mathsf{i}8}$	$y'(u.v.\mathtt{inv}(x, y.M), z.\mathtt{inv}(x, y.N))$

Table 2. Reduction Rules for $\mathtt{inv}$-terms

$\mathtt{dec}(x, y, z.w)$	$\longrightarrow_{\mathsf{d}1}$	w
$\mathtt{dec}(x, y, z.z)$	$\longrightarrow_{\mathsf{d}2}$	$\lambda v.v(x, w.y(w, u.u))$
$\mathtt{dec}(x, y, z.\lambda w.M)$	$\longrightarrow_{\mathsf{d}3}$	$\lambda w.\mathtt{dec}(x, y, z.M)$
$\mathtt{dec}(x, y, z.w(u.v.M, w'.N))$	$\longrightarrow_{\mathsf{d}4}$	$w(u.v.\mathtt{dec}(x, y, z.M), w'.\mathtt{dec}(x, y, z.N))$
$\mathtt{dec}(x, y, z.w(y', z'.M))$	$\longrightarrow_{\mathsf{d}5}$	$w(y', z'.\mathtt{dec}(x, y, z.M))$
$\mathtt{dec}(x, y, z.z(y', z'.M))$	$\longrightarrow_{\mathsf{d}6}$	$y'(x, z''.y(z'', z'.\mathtt{inv}(z'', y'.M)))$
$\mathtt{dec}(x, y, z.x'(z, z'.M))$	$\longrightarrow_{\mathsf{d}7}$	$x(u.v.v(x, z''.y(z'', w.w)), z'.\mathtt{dec}(x, y, z.M))$
$\mathtt{dec}(x, y, z.z(u.v.M, z'.N))$	$\longrightarrow_{\mathsf{d}8}$	$\mathtt{cut}(\{x/u\}\{y/v\}M, y'.y(y', z'.N))$

Table 3. Reduction Rules for $\mathtt{dec}$-terms

In order to reduce the cuts we now suggest a general system called cegs for cut-elimination in Tables 4 and 5 (variants are presented in Section 6). The whole system is called gs and contains the reduction rules in cegs (Tables 4 and 5) plus the ones in oid (Tables 1, 2 and 3).

Summing up :

$\mathsf{Kind_1}$		
$\mathtt{cut}(M, x.x)$	$\longrightarrow_a$	M
$\mathtt{cut}(M, x.y)$	$\longrightarrow_b$	y
$\mathtt{cut}(M, x.\lambda y.N)$	$\longrightarrow_c$	$\lambda y.\mathtt{cut}(M, x.N)$
$\mathtt{cut}(M, x.y(z, w.N))$	$\longrightarrow_d$	$y(z, w.\mathtt{cut}(\mathtt{inv}(w, y.M), x.N))$
$\mathtt{cut}(M, x.y(u.v.N', w.N))$	$\longrightarrow_e$	$y(u.v.\mathtt{cut}(\mathtt{dec}(u, v, y.M), x.N'), w.\mathtt{cut}(\mathtt{inv}(w, y.M), x.N))$
$\mathtt{cut}(\lambda z.M, x.y(x, w.N))$	$\longrightarrow_f$	$y(u.v.\mathtt{cut}(u, z.\mathtt{dec}(u, v, y.M)), w.\mathtt{cut}(\mathtt{inv}(w, y.\lambda z.M), x.N))$
$\mathtt{cut}(z, x.y(x, w.N))$	$\longrightarrow_g$	$y(z, w.\mathtt{cut}(z, x.N))$
$\mathsf{Kind_2}$		
$\mathtt{cut}(y(z, w.M), x.N)$	$\longrightarrow_\pi$	$y(z, w.\mathtt{cut}(M, x.\mathtt{inv}(w, y.N)))$
$\mathtt{cut}(y(u.v.M', w.M), x.N)$	$\longrightarrow_\phi$	$y(u.v.M', w.\mathtt{cut}(M, x.\mathtt{inv}(w, y.N)))$

Table 4. Cut Elimination Rules cegs ($\mathsf{Kind_1}$ and $\mathsf{Kind_2}$)

$\mathsf{Kind_3}$		
$\mathtt{cut}(M, x.x(z, w.N))$	$\longrightarrow_A$	$\mathtt{cut}(\mathtt{cut}(z, y.\mathtt{of}(M, y)), w.N)$
$\mathtt{cut}(M, x.x(u.v.N', w.N))$	$\longrightarrow_B$	$\mathtt{cut}(\mathtt{cut}(\lambda u.\mathtt{cut}(\lambda z.\mathtt{inv}(z, y.\mathtt{of}(M, y)), v.N'), y.\mathtt{of}(M, y)), w.N)$

Table 5. Cut Elimination Rules cegs ($\mathsf{Kind_3}$)

Name of the System	Reduction Rules
oid	Tables 1, 2 and 3
cegs	Tables 4, 5
gs	oid ∪ cegs

As in most cut-elimination systems, the cut-reduction rules can be split into three kinds ($\mathsf{Kind_1}, \mathsf{Kind_2}, \mathsf{Kind_3}$), according to whether they push cuts to the right, to the left, or they break a cut into cuts on smaller types.

Here, owing to the particular inference rules of G4ip and the linearity constraints they impose on free variables, the first two kinds must use the auxiliary constructs $\mathtt{inv}(\_, \_.\_)$ and $\mathtt{dec}(\_, \_, \_.\_)$, rather than just propagate the cuts.

For the third kind of cut-reduction rules, we usually expect both sub-proofs of the cut to introduce the cut-type (on the right and on the left, respectively). In particular, this requires the first argument of the cut-constructor to be a *value*, i.e. a variable or an abstraction, with a functional type, i.e. an implication $A \supset B$. However, just as *any* λ-term can be turned into a value by an η-expansion, here *any* term can be turned into a value by the use of the $\mathtt{of}(\_, \_)$ constructor, with the following rule, which we also call η:

$$M \longrightarrow_\eta \lambda x.\mathtt{of}(M, x) \qquad \text{if } x \notin FV(M)$$

Notice that in both cases this is only sound with respect to typing if the type of the original term is an implication.

Remark 2. All rules of system gs are such that well-formed terms reduce to well-formed terms.

4 A First-Order Syntax for Typed G4ip-Terms

Termination of the above rewrite systems on typed terms will be proved by the decrease of a measure associated to typing derivations. The latter are mapped to a first-order syntax with the following infinite signature:

$$\Sigma = \{\star/0, \mathsf{I}/1, \mathsf{K}/2, \mathsf{J}/1\} \cup \{\mathsf{D}^m/1, \mathsf{C}^m/2 \mid m \text{ is a multiset of types}\}$$

where the notation f/n is used to say that the symbol f has arity n, and the symbols have the following precedence relation:

$$\mathsf{C}^n \succ \mathsf{D}^n \succ \cdots \succ \cdots \succ \mathsf{C}^m \succ \mathsf{D}^m \succ \mathsf{J} \succ \mathsf{K} \succ \mathsf{I} \succ \star \qquad \text{if } n >_{\mathsf{mul}} m$$

Remark 3.

1. The order on types (Definition 3) is well-founded, so $>_{\mathsf{mul}}$ is well-founded [DM79].
2. The order $>_{\mathsf{mul}}$ is well-founded, so $\succ$ is also well-founded.
3. The order $\succ$ is well-founded, so the Multi-Set Path Ordering $\gg_{mpo}$ is also well-founded.

We now consider the *Multi-set Path Ordering* (mpo) [KL80,BN98] on first-order terms induced by the above precedence relation on symbols. This is the relation defined inductively as follows:

$$\frac{}{t_i \ll_{mpo} f(t_1, \ldots, t_n)} \qquad \frac{s \ll_{mpo} t_i}{s \ll_{mpo} f(t_1, \ldots, t_n)}$$

$$\frac{u_i \ll_{mpo} f(t_1, \ldots, t_n) \text{ for all } i}{g(u_1, \ldots, u_m) \ll_{mpo} f(t_1, \ldots, t_n)}\ g \prec f \qquad \frac{\{\!\{t'_1, \ldots, t'_n\}\!\} \ll_{mpo\ mul} \{\!\{t_1, \ldots, t_n\}\!\}}{f(t'_1, \ldots, t'_n) \ll_{mpo} f(t_1, \ldots, t_n)}$$

where g and f are first-order symbols with arities m and n, respectively, and $t_1, \ldots, t_n, t'_1, \ldots, t'_n, u_1, \ldots, u_m, s$ are first-order terms.

It can be shown that $\ll_{mpo}$ is a well-founded order on first-order terms satisfying the *subterm property*, i.e. if s is a subterm of t then $s \ll_{mpo} t$.

Derivations are mapped to this first-order syntax. In particular, since each sequent $\Gamma \Rightarrow M : A$ has at most one derivation, we write $\overline{\Gamma \Rightarrow M : A}$ for such a translation, and even $\overline{M}$ when the context and type are clear from the text, as in the right-hand sides of the following definition.

$$\begin{array}{ll}
\overline{\Gamma, x : A \Rightarrow x : A} & = \star \\
\overline{\Gamma \Rightarrow \lambda x.M : A \supset B} & = \mathsf{I}(\overline{M}) \\
\overline{\Gamma, x : A \supset B, y : A \Rightarrow x(y, z.M) : E} & = \mathsf{I}(\overline{M}) \\
\overline{\Gamma, x : (C \supset D) \supset B \Rightarrow x(u.v.M, z.N) : E} & = \mathsf{K}(\overline{M}, \overline{N}) \\
\overline{\Gamma, x : D \Rightarrow \mathtt{inv}(x, y.M) : E} & = \mathsf{J}(\overline{M}) \\
\overline{\Gamma, x : A \Rightarrow \mathtt{of}(M, x) : B} & = \mathsf{J}(\overline{M}) \\
\overline{\Gamma, x : C, y : D \supset B \Rightarrow \mathtt{dec}(x, y, z.M) : A} & = \mathsf{D}^{\{\!\{\Gamma, (C \supset D) \supset B, A\}\!\}}(\overline{\Gamma, z : (C \supset D) \supset B \Rightarrow M : A}) \\
\overline{\Gamma \Rightarrow \mathtt{cut}(M, x.N) : B} & = \mathsf{C}^{\{\!\{\Gamma, A, B\}\!\}}(\overline{\Gamma \Rightarrow M : A}, \overline{x : A, \Gamma \Rightarrow N : B})
\end{array}$$

Observe that $\overline{M} = \overline{\{x/y\}M}$ for any renaming of M.

5 Subject Reduction and Strong Normalisation

In this section we show two fundamental properties of system gs. The first one is subject reduction and it guarantees that types are preserved by the reduction system. The second one is strong normalisation and it guarantees that there is no infinite reduction sequence starting from a typed term. Strong normalisation is shown by a decreasing measure given by the Multi-Set Path Ordering of Section 4.

Theorem 1. *If* $\Gamma \Rightarrow L : E$ *and* $L \longrightarrow_{\mathtt{gs}} L'$*, then* $\Gamma \Rightarrow L' : E$ *and* $\overline{L} \gg_{mpo} \overline{L'}$*.*

Proof: By induction on the proof $\Gamma \vdash t : A$. We consider only the cases where reduction takes place at the root.

o1 $\mathtt{of}(y, x) \longrightarrow_{\mathtt{o1}} y(x, z.z)$
The derivation

$$\dfrac{\dfrac{}{\Gamma', y : A \supset B \Rightarrow y : A \supset B}\, Ax}{\Gamma', y : A \supset B, x : A \Rightarrow \mathtt{of}(y, x) : B}\, Of$$

rewrites to

$$\dfrac{\dfrac{}{\Gamma, x : A, z : B \Rightarrow z : B}\, Ax}{\Gamma', y : A \supset B, x : A \Rightarrow y(x, z.z) : B}\, L0\supset$$

Also, $\overline{L} = \mathsf{J}(\star) \gg_{mpo} \mathsf{I}(\star) = \overline{L'}$ since $\mathsf{J} \succ \mathsf{I}$.

o2 $\mathtt{of}(\lambda y.M, x) \longrightarrow_{\mathtt{o2}} \{x/y\}M$
The derivation

$$\dfrac{\dfrac{\Gamma, y : A \Rightarrow M : B}{\Gamma \Rightarrow \lambda y.M : A \supset B}\, R\supset}{\Gamma, x : A \Rightarrow \mathtt{of}(\lambda y.M, x) : B}\, Of$$

rewrites to

$$\dfrac{\Gamma, y : A \Rightarrow M : B}{\Gamma, x : A \Rightarrow \{x/y\}M : B}\, Ren$$

Also, $\overline{L} = \mathsf{J}(\mathsf{I}(\overline{M})) \gg_{mpo} \overline{M} = \overline{L'}$ by the subterm property of $\ll_{mpo}$.

o3 $\mathtt{of}(y(z, w.N), x) \longrightarrow_{\mathtt{o3}} y(z, w.\mathtt{of}(N, x))$
The derivation

$$\dfrac{\dfrac{\Gamma', z : C, w : D \Rightarrow N : A \supset B}{\Gamma', z : C, y : C \supset D \Rightarrow y(z, w.N) : A \supset B}\, L0\supset}{\Gamma', z : C, y : C \supset D, x : A \Rightarrow \mathtt{of}(y(z, w.N), x) : B}\, Of$$

rewrites to

$$\dfrac{\dfrac{\Gamma', z : C, w : D \Rightarrow N : A \supset B}{\Gamma', z : C, w : D, x : A \Rightarrow \mathtt{of}(N, x) : B}\, Of}{\Gamma', z : C, y : C \supset D, x : A \Rightarrow y(z, w.\mathtt{of}(N, x)) : B}\, L0\supset$$

Also, $\overline{L} = \mathsf{J}(\mathsf{I}(\overline{N})) \gg_{mpo} \mathsf{I}(\mathsf{J}(\overline{N})) = \overline{L'}$ since $\mathsf{J} \succ \mathsf{I}$.

o4 $\mathtt{of}(y(u.v.M, w.N), x) \longrightarrow_{\mathsf{o4}} y(u.v.M, w.\mathtt{of}(N, x))$ So A is of the form $C \supset D$.
The derivation

$$\dfrac{\dfrac{\Gamma', u{:}C, v{:}D \supset B' \Rightarrow M{:}D \qquad \Gamma', w{:}B' \Rightarrow N{:}A \supset B}{\Gamma', y{:}(C \supset D) \supset B' \Rightarrow y(u.v.M, w.N){:}A \supset B}\, L\supset\supset}{\Gamma', y{:}(C \supset D) \supset B', x{:}A \Rightarrow \mathtt{of}(y(u.v.M, w.N), x){:}B}\, Of$$

rewrites to

$$\dfrac{\dfrac{\dfrac{\Gamma', u{:}C, v{:}D \supset B' \Rightarrow M{:}D}{\Gamma', u{:}C, v{:}D \supset B', x{:}A \Rightarrow M{:}D}\,(W) \quad \dfrac{\Gamma', w{:}B' \Rightarrow N{:}A \supset B}{\Gamma', w{:}B', x{:}A \Rightarrow \mathtt{of}(N, x){:}B}\, Of}{\Gamma', y{:}(C \supset D) \supset B', x{:}A \Rightarrow y(u.v.M, w.\mathtt{of}(N, x)){:}B}\, L\supset\supset$$

Also, $\overline{L} = \mathsf{J}(\mathsf{K}(\overline{M}, \overline{N})) \gg_{mpo} \mathsf{K}(\overline{M}, \mathsf{J}(\overline{N})) = \overline{L'}$ since $\mathsf{J} \succ \mathsf{K}$,.

i1 $\mathtt{inv}(x, y.z) \longrightarrow_{\mathsf{i1}} z$
The derivation

$$\dfrac{\dfrac{}{\Gamma', z{:}E, y{:}A \supset B \Rightarrow z{:}E}\, Ax}{\Gamma', z{:}E, x{:}B \Rightarrow \mathtt{inv}(x, y.z){:}E}\, Inv$$

rewrites to

$$\dfrac{}{\Gamma', z{:}E, x{:}B \Rightarrow z{:}E}\, Ax$$

Also, $\overline{L} = \mathsf{J}(\star) \gg_{mpo} \star = \overline{L'}$ holds by the subterm property of $\ll_{mpo}$.

i2 $\mathtt{inv}(x, y.y) \longrightarrow_{\mathsf{i2}} \lambda z.x$
The derivation

$$\dfrac{\dfrac{}{\Gamma', y{:}A \supset B \Rightarrow y{:}A \supset B}\, Ax}{\Gamma', x{:}B \Rightarrow \mathtt{inv}(x, y.y){:}A \supset B}\, Inv$$

rewrites to

$$\dfrac{\dfrac{}{\Gamma', x{:}B, z{:}A \Rightarrow x{:}B}\, Ax}{\Gamma', x{:}B \Rightarrow \lambda z.x{:}A \supset B}\, R\supset$$

Also, $\overline{L} = \mathsf{J}(\star) \succ \mathsf{I}(\star) = \overline{L'}$ holds by $\mathsf{J} \succ \mathsf{I}$.

i3 $\mathtt{inv}(x, y.\lambda z.M) \longrightarrow_{\mathsf{i3}} \lambda z.\mathtt{inv}(x, y.M)$ with $E = C \supset D$
The derivation

$$\dfrac{\dfrac{\dfrac{\Gamma', y{:}A \supset B, z{:}C \Rightarrow M{:}D}{\Gamma', y{:}A \supset B \Rightarrow \lambda z.M{:}C \supset D}\, R\supset}{\Gamma', x{:}B \Rightarrow \mathtt{inv}(x, y.\lambda z.M){:}C \supset D}\, Inv$$

rewrites to

$$\dfrac{\dfrac{\dfrac{\Gamma', y{:}A \supset B, z{:}C \Rightarrow M{:}D}{\Gamma', x{:}B, z{:}C \Rightarrow \mathtt{inv}(x, y.M){:}D}\, Inv}{\Gamma', x{:}B \Rightarrow \lambda z.\mathtt{inv}(x, y.M){:}C \supset D}\, R\supset$$

Also, $\overline{L} = \mathsf{J}(\mathsf{I}(\overline{M})) \gg_{mpo} \mathsf{I}(\mathsf{J}(\overline{M})) = \overline{L'}$ by $\mathsf{J} \succ \mathsf{I}$.

i4 $\mathtt{inv}(x, y.y(w, z.N)) \longrightarrow_{\mathsf{i4}} \{x/z\}N$
The derivation

$$\dfrac{\dfrac{\Gamma', w{:}A, z{:}B \Rightarrow N{:}E}{\Gamma', w{:}A, y{:}A{\supset}B \Rightarrow y(w, z.N){:}E}\, L0{\supset}}{\Gamma', w{:}A, x{:}B \Rightarrow \mathtt{inv}(x, y.y(w, z.N)){:}E}\, Inv$$

rewrites to

$$\dfrac{\Gamma', w{:}A, z{:}B \Rightarrow N{:}E}{\Gamma', w{:}A, x{:}B \Rightarrow \{x/z\}N{:}E}\, Ren$$

Also, $\overline{M} = \mathsf{J}(\mathsf{I}(\overline{N})) \gg_{mpo} \overline{N} = \overline{M'}$ holds by the subterm property of $\ll_{mpo}$.

i5 $\mathtt{inv}(x, y.y(u.v.M, z.N)) \longrightarrow_{\mathsf{i5}} \{x/z\}N$ with $A = C{\supset}D$
The derivation

$$\dfrac{\dfrac{\Gamma', u{:}C, v{:}D{\supset}B \Rightarrow M{:}D \quad \Gamma', z{:}B \Rightarrow N{:}E}{\Gamma', y{:}A{\supset}B \Rightarrow y(u.v.M, z.N){:}E}\, L{\supset}{\supset}}{\Gamma', x{:}B \Rightarrow \mathtt{inv}(x, y.y(u.v.M, z.N)){:}E}\, Inv$$

rewrites to

$$\dfrac{\Gamma', z{:}B \Rightarrow N{:}E}{\Gamma', x{:}B \Rightarrow \{x/z\}N{:}E}\, (Ren)$$

Also, $\overline{L} = \mathsf{J}(\mathsf{K}(\overline{M}, \overline{N})) \gg_{mpo} \overline{N} = \overline{L'}$ holds by the subterm property of $\ll_{mpo}$.

i6 $\mathtt{inv}(x, y.w(y, z.N)) \longrightarrow_{\mathsf{i6}} w(u.v.x, z.\mathtt{inv}(x, y.N))$
The derivation

$$\dfrac{\dfrac{\Gamma', y{:}A{\supset}B, z{:}C \Rightarrow N{:}E}{\Gamma', w{:}(A{\supset}B){\supset}C, y{:}A{\supset}B \Rightarrow w(y, z.N){:}E}\, L0{\supset}}{\Gamma', w{:}(A{\supset}B){\supset}C, x{:}B \Rightarrow \mathtt{inv}(x, y.w(y, z.N)){:}E}\, Inv$$

rewrites to

$$\dfrac{\dfrac{}{\Gamma', x{:}B, u{:}A, v{:}B{\supset}C \Rightarrow x{:}B}\, Ax \quad \dfrac{\Gamma', y{:}A{\supset}B, z{:}C \Rightarrow N{:}E}{\Gamma', x{:}B, z{:}C \Rightarrow \mathtt{inv}(x, y.N){:}E}\, Inv}{\Gamma', w{:}(A{\supset}B){\supset}C, x{:}B \Rightarrow w(u.v.x, z.\mathtt{inv}(x, y.N)){:}E}\, L{\supset}{\supset}$$

Also, $\overline{L} = \mathsf{J}(\mathsf{I}(\overline{N})) \gg_{mpo} \mathsf{K}(\star, \mathsf{J}(\overline{N})) = \overline{L'}$ by $\mathsf{J} \succ \mathsf{K}, \star$.

i7 $\mathtt{inv}(x, y.y'(w, z.N)) \longrightarrow_{\mathsf{i7}} y'(w, z.\mathtt{inv}(x, y.N))$
The derivation

$$\dfrac{\dfrac{\Gamma', w{:}C, z{:}D, y{:}A{\supset}B \Rightarrow N{:}E}{\Gamma', w{:}C, y'{:}C{\supset}D, y{:}A{\supset}B \Rightarrow y'(w, z.N){:}E}\, L0{\supset}}{\Gamma', w{:}C, y'{:}C{\supset}D, x{:}B \Rightarrow \mathtt{inv}(x, y.y'(w, z.N)){:}E}\, Inv$$

rewrites to

$$\dfrac{\dfrac{\Gamma', w{:}C, z{:}D, y{:}A{\supset}B \Rightarrow N{:}E}{\Gamma', w{:}C, z{:}D, x{:}B \Rightarrow \mathtt{inv}(x, y.N){:}E}\, Inv}{\Gamma', w{:}C, y'{:}C{\supset}D, x{:}B \Rightarrow y'(w, z.\mathtt{inv}(x, y.N)){:}E}\, L0{\supset}$$

Also, $\overline{L} = \mathsf{J}(\mathsf{I}(\overline{N})) \gg_{mpo} \mathsf{I}(\mathsf{J}(\overline{N})) = \overline{L'}$ by $\mathsf{J} \succ \mathsf{I}$.

i8 $\mathtt{inv}(x, y.y'(u.v.M, z.N)) \longrightarrow_{\mathsf{i8}} y'(u.v.\mathtt{inv}(x, y.M), z.\mathtt{inv}(x, y.N))$
The derivation

$$\dfrac{\dfrac{\Gamma', y{:}A{\supset}B, u{:}C, v{:}D{\supset}B' \Rightarrow M{:}D \qquad \Gamma', y{:}A{\supset}B, z{:}B' \Rightarrow N{:}E}{\Gamma', y'{:}(C{\supset}D){\supset}B', y{:}A{\supset}B \Rightarrow y'(u.v.M, z.N){:}E}\, L{\supset}{\supset}}{\Gamma', y'{:}(C{\supset}D){\supset}B', x{:}B \Rightarrow \mathtt{inv}(x, y.y'(u.v.M, z.N)){:}E}\, Inv$$

rewrites to

$$\dfrac{\dfrac{\Gamma', y{:}A{\supset}B, u{:}C, v{:}D{\supset}B' \Rightarrow M{:}D}{\Gamma', x{:}B, u{:}C, v{:}D{\supset}B' \Rightarrow \mathtt{inv}(x, y.M){:}D}\, Inv \quad \dfrac{\Gamma', y{:}A{\supset}B, z{:}B' \Rightarrow N{:}E}{\Gamma', x{:}B, z{:}B' \Rightarrow \mathtt{inv}(x, y.N){:}E}\, Inv}{\Gamma', y'{:}(C{\supset}D){\supset}B', x{:}B \Rightarrow y'(u.v.\mathtt{inv}(x, y.M), z.\mathtt{inv}(x, y.N)){:}E}\, L{\supset}{\supset}$$

Also, $\overline{L} = \mathsf{J}(\mathsf{K}(\overline{M}, \overline{N})) \gg_{mpo} \mathsf{K}(\mathsf{J}(\overline{M}), \mathsf{J}(\overline{N})) = \overline{L'}$ by $\mathsf{J} \succ \mathsf{K}$.

d1 $\mathtt{dec}(x, y, z.w) \longrightarrow_{\mathsf{d1}} w$, where x, y, z and w are all distinct.
The derivation

$$\dfrac{\dfrac{}{\Gamma', w{:}E, z{:}(C{\supset}D){\supset}B \Rightarrow w{:}E}\, Ax}{\Gamma', w{:}E, x{:}C, y{:}D{\supset}B \Rightarrow \mathtt{dec}(x, y, z.w){:}E}\, Dec$$

rewrites to

$$\dfrac{}{\Gamma', w{:}E, x{:}C, y{:}D{\supset}B \Rightarrow w{:}E}\, Ax$$

Also, $\overline{L} = \mathsf{D}^m(\star) \gg_{mpo} \star = \overline{L'}$, where $m = \{\!\{\Gamma', E, (C{\supset}D){\supset}B, E\}\!\}$.

d2 $\mathtt{dec}(x, y, z.z) \longrightarrow_{\mathsf{d2}} \lambda v.v(x, w.y(w, u.u))$.
The derivation

$$\dfrac{\dfrac{}{\Gamma', z{:}(C{\supset}D){\supset}B \Rightarrow z{:}(C{\supset}D){\supset}B}\, Ax}{\Gamma', x{:}C, y{:}D{\supset}B \Rightarrow \mathtt{dec}(x, y, z.z){:}E}\, Dec$$

rewrites to

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma', x{:}C, w{:}D, u{:}B \Rightarrow u{:}B}\, Ax}{\Gamma', x{:}C, w{:}D, y{:}D{\supset}B \Rightarrow y(w, u.u){:}B}\, L0{\supset}}{\Gamma', x{:}C, y{:}D{\supset}B, v{:}C{\supset}D \Rightarrow v(x, w.y(w, u.u)){:}B}\, L0{\supset}}{\Gamma', x{:}C, y{:}D{\supset}B \Rightarrow \lambda v.v(x, w.y(w, u.u)){:}(C{\supset}D){\supset}B}\, R{\supset}$$

Also, $\overline{L} = \mathsf{D}^m(\star) \gg_{mpo} \mathsf{I}(\mathsf{I}(\mathsf{I}(\star))) = \overline{L'}$, where $m = \{\!\{\Gamma', (C{\supset}D){\supset}B, (C{\supset}D){\supset}B\}\!\}$, by $\mathsf{D}^m \succ \mathsf{I}$.

d3 $\mathtt{dec}(x, y, z.\lambda w.M) \longrightarrow_{\mathsf{d3}} \lambda w.\mathtt{dec}(x, y, z.M)$.
The derivation

$$\dfrac{\dfrac{\Gamma', z{:}(C{\supset}D){\supset}B, w{:}E_1 \Rightarrow M{:}E_2}{\Gamma', z{:}(C{\supset}D){\supset}B \Rightarrow \lambda w.M{:}E_1{\supset}E_2}\ R{\supset}}{\Gamma', x{:}C, y{:}D{\supset}B \Rightarrow \mathtt{dec}(x,y,z.\lambda w.M){:}E_1{\supset}E_2}\ Dec$$

rewrites to

$$\dfrac{\dfrac{\Gamma', z{:}(C{\supset}D){\supset}B, w{:}E_1 \Rightarrow M{:}E_2}{\Gamma', x{:}C, y{:}D{\supset}B, w{:}E_1 \Rightarrow \mathtt{dec}(x,y,z.M){:}E_2}\ Dec}{\Gamma', x{:}C, y{:}D{\supset}B \Rightarrow \lambda w.\mathtt{dec}(x,y,z.M){:}E_1{\supset}E_2}\ R{\supset}$$

Let $m = \{\!\{\Gamma', (C{\supset}D){\supset}B, E_1{\supset}E_2\}\!\}$ and $n = \{\!\{\Gamma', (C{\supset}D){\supset}B, E_1, E_2\}\!\}$. We have

$$\overline{L} = \mathsf{D}^m(\mathsf{I}(\overline{M})) \gg_{mpo} \mathsf{I}(\mathsf{D}^n(\overline{M})) = \overline{L'}$$

since $\mathsf{D}^m \succ \mathsf{I}, \mathsf{D}^n$ because $m >_{\mathsf{mul}} n$.

d4 $\mathtt{dec}(x,y,z.w(u.v.M, w'.N)) \longrightarrow_{\mathsf{d}_4} w(u.v.\mathtt{dec}(x,y,z.M), w'.\mathtt{dec}(x,y,z.N))$.
The derivation

$$\dfrac{\dfrac{\Gamma', v{:}F, u{:}G{\supset}H, z{:}(C{\supset}D){\supset}B \Rightarrow M{:}G \quad \Gamma', w'{:}H, z{:}(C{\supset}D){\supset}B \Rightarrow N{:}E}{\Gamma', w{:}(F{\supset}G){\supset}H, z{:}(C{\supset}D){\supset}B \Rightarrow w(u.v.M, w'.N){:}E}\ L{\supset}{\supset}}{\Gamma', w{:}(F{\supset}G){\supset}H, x{:}C, y{:}D{\supset}B \Rightarrow \mathtt{dec}(x,y,z.w(u.v.M, w'.N)){:}E}\ Dec$$

rewrites to

$$\dfrac{\dfrac{\Gamma', v{:}F, u{:}G{\supset}H, x{:}(C{\supset}D){\supset}B \Rightarrow M{:}G}{\Gamma', v{:}F, u{:}G{\supset}H, x{:}C, y{:}D{\supset}B \Rightarrow M'{:}G}\ Dec \quad \dfrac{\Gamma', w'{:}H, z{:}(C{\supset}D){\supset}B \Rightarrow N{:}E}{\Gamma', w'{:}H, x{:}C, y{:}D{\supset}B \Rightarrow N'{:}E}\ Dec}{\Gamma', w{:}(F{\supset}G){\supset}H, x{:}C, y{:}D{\supset}B \Rightarrow w(u.v.M', w'.N'){:}G}\ L{\supset}{\supset}$$

with $M' = \mathtt{dec}(x,y,z.M)$ and $N' = \mathtt{dec}(x,y,z.N)$.
Let $k = \{\!\{\Gamma', (F{\supset}G){\supset}H, (C{\supset}D){\supset}B, E\}\!\}$ and $m = \{\!\{\Gamma', F, G{\supset}H, (C{\supset}D){\supset}B, G\}\!\}$ and $n = \{\!\{\Gamma', H, (C{\supset}D){\supset}B, E\}\!\}$. We have

$$\overline{L} = \mathsf{D}^k(\mathsf{K}(\overline{M}, \overline{N})) \gg_{mpo} \mathsf{K}(\mathsf{D}^m(\overline{M}, \mathsf{D}^n(\overline{N})) = \overline{L'}$$

since $\mathsf{D}^k \succ \mathsf{K}, \mathsf{D}^m, \mathsf{D}^n$ because $k >_{\mathsf{mul}} m, n$.

d5 $\mathtt{dec}(x,y,z.w(y',z'.M)) \longrightarrow_{\mathsf{d}_5} w(y',z'.\mathtt{dec}(x,y,z.M))$.
The derivation

$$\dfrac{\dfrac{\Gamma', y'{:}F, z'{:}G, z{:}(C{\supset}D){\supset}B \Rightarrow M{:}E}{\Gamma', y'{:}F, w{:}F{\supset}G, z{:}(C{\supset}D){\supset}B \Rightarrow w(y',z'.M){:}E}\ L0{\supset}}{\Gamma', y'{:}F, w{:}F{\supset}G, x{:}C, y{:}D{\supset}B \Rightarrow \mathtt{dec}(x,y,z.w(y',z'.M)){:}E}\ Dec$$

rewrites to

$$\dfrac{\dfrac{\Gamma', y'{:}F, z'{:}G, z{:}(C{\supset}D){\supset}B \Rightarrow M{:}E}{\Gamma', y'{:}F, z'{:}G, x{:}C, y{:}D{\supset}B \Rightarrow \mathtt{dec}(x,y,z.M){:}E}\ Dec}{\Gamma', y'{:}F, w{:}F{\supset}G, x{:}C, y{:}D{\supset}B \Rightarrow w(y', z'.\mathtt{dec}(x,y,z.M)){:}E}\ L0{\supset}$$

Let $k = \{\!\{\Gamma', F, F{\supset}G, (C{\supset}D){\supset}B, E\}\!\}$ and $m = \{\!\{\Gamma', F, G, (C{\supset}D){\supset}B, E\}\!\}$. We have

$$\overline{L} = \mathsf{D}^k(\mathsf{I}(\overline{M})) \gg_{mpo} \mathsf{I}(\mathsf{D}^m(\overline{M})) = \overline{L'}$$

since $\mathsf{D}^k \succ \mathsf{I}, \mathsf{D}^m$ because $k >_{\mathsf{mul}} m$.

d6 $\mathtt{dec}(x,y,z.z(y',z'.M)) \longrightarrow_{\mathsf{d6}} y'(x,z''.y(z'',z'.\mathtt{inv}(z'',y'.M)))$.
The derivation

$$\dfrac{\dfrac{\Gamma', y'{:}C{\supset}D, z'{:}B \Rightarrow M{:}E}{\Gamma', z{:}(C{\supset}D){\supset}B, y'{:}C{\supset}D \Rightarrow z(y',z'.M){:}E}\ L0{\supset}}{\Gamma', x{:}C, y{:}D{\supset}B, y'{:}C{\supset}D \Rightarrow \mathtt{dec}(x,y,z.z(y',z'.M)){:}E}\ Dec$$

rewrites to

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma', y'{:}C{\supset}D, z'{:}B \Rightarrow M{:}E}{\Gamma', z''{:}D, z'{:}B \Rightarrow \mathtt{inv}(z'',y'.M){:}E}\ Inv}{\Gamma', z''{:}D, y{:}D{\supset}B \Rightarrow y(z'',z'.\mathtt{inv}(z'',y'.M)){:}E}\ L0{\supset}}{\Gamma', x{:}C, z''{:}D, y{:}D{\supset}B \Rightarrow y(z'',z'.\mathtt{inv}(z'',y'.M)){:}E}\ (W)}{\Gamma', y'{:}C{\supset}D, x{:}C, y{:}D{\supset}B \Rightarrow y'(x,z''.y(z'',z'.\mathtt{inv}(z'',y'.M))){:}E}\ L0{\supset}$$

Also, $\overline{L} = \mathsf{D}^k(\mathsf{I}(\overline{M})) \gg_{mpo} \mathsf{I}(\mathsf{I}(\mathsf{J}(\overline{M}))) = \overline{L'}$ since $\mathsf{D}^k \succ \mathsf{I}, \mathsf{J}$, where $k = \{\!\{\Gamma', (C{\supset}D){\supset}B, C{\supset}D, E\}\!\}$.

d7 $\mathtt{dec}(x,y,z.x'(z,z'.M)) \longrightarrow_{\mathsf{d7}} x(u.v.v(x,z''.y(z'',w.w)), z'.\mathtt{dec}(x,y,z.M))$.
The derivation

$$\dfrac{\dfrac{\Gamma', z{:}(C{\supset}D){\supset}B, z'{:}A \Rightarrow M{:}E}{\Gamma', x'{:}((C{\supset}D){\supset}B){\supset}A, z{:}(C{\supset}D){\supset}B \Rightarrow x'(z,z'.M){:}E}\ L0{\supset}}{\Gamma', x'{:}((C{\supset}D){\supset}B){\supset}A, x{:}C, y{:}D{\supset}B \Rightarrow \mathtt{dec}(x,y,z.x'(z,z'.M)){:}E}\ Dec$$

rewrites to

$$\dfrac{\dfrac{\cdots}{\Gamma', \ldots \Rightarrow v(x,z''.y(z'',w.w)){:}B} \quad \dfrac{\Gamma', z{:}(C{\supset}D){\supset}B, z'{:}A \Rightarrow M{:}E}{\Gamma', x{:}C, y{:}D{\supset}B, z'{:}A \Rightarrow \mathtt{dec}(x,y,z.M){:}E}\ Dec}{\Gamma', x{:}((C{\supset}D){\supset}B){\supset}A, y{:}C, z{:}D{\supset}B \Rightarrow x(u.v.v(x,z''.y(z'',w.w)), z'.\mathtt{dec}(x,y,z.M)){:}E}\ L{\supset}{\supset}$$

with first premiss is constructed as follows

$$\dfrac{\dfrac{\dfrac{}{\Gamma', x{:}C, w{:}B, u{:}B{\supset}A, z''{:}D \Rightarrow w{:}B}\ Ax}{\Gamma', x{:}C, y{:}D{\supset}B, u{:}B{\supset}A, z''{:}D \Rightarrow y(z'',w.w){:}B}\ L0{\supset}}{\Gamma', x{:}C, y{:}D{\supset}B, u{:}B{\supset}A, v{:}C{\supset}D \Rightarrow v(x,z''.y(z'',w.w)){:}B}\ L0{\supset}$$

Let $k = \{\!\{\Gamma', (C \supset D) \supset B, ((C \supset D) \supset B) \supset A, E\}\!\}$ and $m = \{\!\{\Gamma', (C \supset D) \supset B, A, E\}\!\}$. We have

$$\overline{L} = \mathsf{D}^k(\mathsf{I}(\overline{M})) \gg_{mpo} \mathsf{K}(\mathsf{I}(\mathsf{I}(\star)), \mathsf{D}^m(\overline{M})) = \overline{L'}$$

since $\mathsf{D}^k \succ \mathsf{K}, \mathsf{I}, \star, \mathsf{D}^m$ because $k >_{\mathsf{mul}} m$.

d8 $\mathtt{dec}(x, y, z.z(u.v.M, z'.N)) \longrightarrow_{\mathsf{d8}} \mathtt{cut}(\{y/u\}\{x/v\}M, y'.y(y', z'.N))$.
The derivation

$$\dfrac{\dfrac{\Gamma', v{:}C, u{:}D \supset B \Rightarrow M{:}D \qquad \Gamma', z'{:}B \Rightarrow N{:}E}{\Gamma', z{:}(C \supset D) \supset B \Rightarrow z(u.v.M, z'.N){:}E}\, L\supset\supset}{\Gamma', x{:}C, y{:}D \supset B \Rightarrow \mathtt{dec}(x, y, z.z(u.v.M, z'.N)){:}E}\, Dec$$

rewrites to

$$\dfrac{\dfrac{\dfrac{\Gamma', v{:}C, u{:}D \supset B \Rightarrow M{:}D}{\Gamma', x{:}C, y{:}D \supset B \Rightarrow \{y/u\}\{x/v\}M{:}D}\,(Ren) \quad \dfrac{\dfrac{\Gamma', z'{:}B \Rightarrow N{:}E}{y'{:}D, \Gamma', x{:}C, z'{:}B \Rightarrow N{:}E}\,(W)}{y'{:}D, \Gamma', x{:}C, y{:}D \supset B \Rightarrow y(y', z'.N){:}E}\, L0\supset}{\Gamma', x{:}C, y{:}D \supset B \Rightarrow \mathtt{cut}(\{y/u\}\{x/v\}M, y'.y(y', z'.N)){:}E}\, Cut$$

Let $k = \{\!\{\Gamma', (C \supset D) \supset B, E\}\!\}$ and $j = \{\!\{\Gamma', D, C, D \supset B, E\}\!\}$. We have

$$\overline{L} = \mathsf{D}^k(\mathsf{K}(\overline{M}, \overline{N})) \gg_{mpo} \mathsf{C}^j(\overline{M}, \mathsf{I}(\overline{N})) = \overline{L'}$$

since $\mathsf{D}^k \succ \mathsf{C}^j, \mathsf{I}$ because $k >_{\mathsf{mul}} j$.

a $\mathtt{cut}(M, x.x) \longrightarrow_a M$.
The derivation

$$\dfrac{\Gamma \Rightarrow M{:}A \quad \overline{\Gamma, x{:}A \Rightarrow x{:}A}\, Ax}{\Gamma \Rightarrow \mathtt{cut}(M, x.x){:}A}\, Cut$$

rewrites to

$$\Gamma \Rightarrow M{:}A$$

Also, $\overline{L} = \mathsf{C}^m(\overline{M}, \star) \gg_{mpo} \overline{M} = \overline{L'}$, where $m = \{\!\{\Gamma, A, A\}\!\}$.

b $\mathtt{cut}(M, x.y) \longrightarrow_b y$.
The derivation

$$\dfrac{\Gamma', y{:}E \Rightarrow M{:}A \quad \overline{\Gamma', y{:}E, x{:}A \Rightarrow y{:}E}\, Ax}{\Gamma', y{:}E \Rightarrow \mathtt{cut}(M, x.y){:}E}\, Cut$$

rewrites to

$$\Gamma', y{:}E \Rightarrow y{:}E$$

Also, $\overline{L} = \mathsf{C}^m(\overline{M}, \star) \gg_{mpo} \star = \overline{L'}$, where $m = \{\!\{\Gamma', E, A, E\}\!\}$.

c $\mathtt{cut}(M, x.\lambda y.N) \longrightarrow_c \lambda y.\mathtt{cut}(M, x.N)$.
The derivation

$$\dfrac{\Gamma \Rightarrow M : A \quad \dfrac{x : A, \Gamma, y : C \Rightarrow N : D}{x : A, \Gamma \Rightarrow \lambda y.N : C \supset D}\, R\supset}{\Gamma \Rightarrow \mathtt{cut}(M, x.\lambda y.N) : C \supset D}\, Cut$$

rewrites to

$$\dfrac{\dfrac{\Gamma \Rightarrow M : A}{\Gamma, y : C \Rightarrow M : A}\,(W) \quad x : A, \Gamma, y : C \Rightarrow N : D}{\dfrac{\Gamma, y : C \Rightarrow \mathtt{cut}(M, x.N) : D}{\Gamma \Rightarrow \lambda y.\mathtt{cut}(M, x.N) : C \supset D}\, R\supset}\, Cut$$

Let $k = \{\!\{A, \Gamma, C \supset D\}\!\}$ and $j = \{\!\{A, \Gamma, C, D\}\!\}$. We have

$$\overline{L} = \mathsf{C}^k(\overline{M}, \mathsf{I}(\overline{N})) \gg_{mpo} \mathsf{I}(\mathsf{C}^j(\overline{M}, \overline{N})) = \overline{L'}$$

since $\mathsf{C}^k \succ \mathsf{I}, \mathsf{C}^j$ because $k >_{\mathsf{mul}} j$.

d $\mathtt{cut}(M, x.z(y, w.N)) \longrightarrow_d z(y, w.\mathtt{cut}(\mathtt{inv}(w, z.M), x.N))$.
The derivation

$$\dfrac{\Gamma', y : C, z : C \supset B \Rightarrow M : A \quad \dfrac{x : A, \Gamma', y : C, w : B \Rightarrow N : E}{x : A, \Gamma', y : C, z : C \supset B \Rightarrow z(y, w.N) : E}\, L0\supset}{\Gamma', y : C, z : C \supset B \Rightarrow \mathtt{cut}(M, x.z(y, w.N)) : E}\, Cut$$

rewrites to

$$\dfrac{\dfrac{\Gamma', y : C, z : C \supset B \Rightarrow M : A}{\Gamma', y : C, w : B \Rightarrow \mathtt{inv}(w, z.M) : A}\, Inv \quad x : A, \Gamma', y : C, w : B \Rightarrow N : E}{\dfrac{\Gamma', y : C, w : B \Rightarrow \mathtt{cut}(\mathtt{inv}(w, z.M), x.N) : E}{\Gamma', y : C, z : C \supset B \Rightarrow z(y, w.\mathtt{cut}(\mathtt{inv}(w, z.M), x.N)) : E}\, L0\supset}\, Cut$$

Let $k = \{\!\{A, \Gamma', C, C \supset B, E\}\!\}$ and $j = \{\!\{A, \Gamma', C, B, E\}\!\}$. We have

$$\overline{L} = \mathsf{C}^k(\overline{M}, \mathsf{I}(\overline{N})) \gg_{mpo} \mathsf{I}(\mathsf{C}^j(\mathsf{J}(\overline{M}), \overline{N}) = \overline{L'}$$

since $\mathsf{C}^k \succ \mathsf{I}, \mathsf{C}^j, \mathsf{J}$ because $k >_{\mathsf{mul}} j$.

e $\mathtt{cut}(M, x.y(u.v.N, z.N')) \longrightarrow_e y(u.v.\mathtt{cut}(\mathtt{dec}(v, u, y.M), x.N), z.\mathtt{cut}(\mathtt{inv}(z, y.M), x.N'))$.
The derivation

$$\dfrac{\Gamma', y : (C \supset D) \supset B \Rightarrow M : A \quad \dfrac{x : A, \Gamma', v : C, u : D \supset B \Rightarrow N : D \quad x : A, \Gamma', z : B \Rightarrow N' : E}{x : A, \Gamma', y : (C \supset D) \supset B \Rightarrow y(u.v.N, z.N') : E}\, L\supset\supset}{\Gamma', y : (C \supset D) \supset B \Rightarrow \mathtt{cut}(M, x.y(u.v.N, z.N')) : E}\, Cut$$

rewrites to

$$\dfrac{\dfrac{\mathcal{D}}{\Gamma', v : C, u : D \supset B \Rightarrow \mathtt{cut}(\mathtt{dec}(v, u, y.M), x.N) : D} \quad \dfrac{\mathcal{D}'}{\Gamma', z : B \Rightarrow \mathtt{cut}(\mathtt{inv}(z, y.M), x.N') : E}}{\Gamma', y : (C \supset D) \supset B \Rightarrow y(u.v.\mathtt{cut}(\mathtt{dec}(v, u, y.M), x.N), z.\mathtt{cut}(\mathtt{inv}(z, y.M), x.N')) : E}\, L\supset\supset$$

where $\mathcal{D}$ is the following derivation:

$$\dfrac{\dfrac{\Gamma', y{:}(C{\supset}D){\supset}B \Rightarrow M{:}A}{\Gamma', v{:}C, u{:}D{\supset}B \Rightarrow \mathtt{dec}(v,u,y.M){:}A}\ Dec \quad x{:}A, \Gamma', v{:}C, u{:}D{\supset}B \Rightarrow N{:}D}{\Gamma', v{:}C, u{:}D{\supset}B \Rightarrow \mathtt{cut}(\mathtt{dec}(v,u,y.M), x.N){:}D}\ Cut$$

and $\mathcal{D}'$ is the following derivation:

$$\dfrac{\dfrac{\Gamma', y{:}(C{\supset}D){\supset}B \Rightarrow M{:}A}{\Gamma', z{:}B \Rightarrow \mathtt{inv}(z,y.M){:}A}\ Inv \quad x{:}A, \Gamma', z{:}B \Rightarrow N'{:}E}{\Gamma', z{:}B \Rightarrow \mathtt{cut}(\mathtt{inv}(z,y.M), x.N'){:}E}\ Cut$$

Let $k = \{\!\{A, \Gamma', (C{\supset}D){\supset}B, E\}\!\}$ and $j = \{\!\{A, \Gamma', C, D{\supset}B, D\}\!\}$ and $i = \{\!\{A, \Gamma', B, E\}\!\}$ and $h = \{\!\{\Gamma', (C{\supset}D){\supset}B, A\}\!\}$.
We have

$$\overline{L} = \mathsf{C}^k(\overline{M}, \mathsf{K}(\overline{N'}, \overline{N})) \gg_{mpo} \mathsf{K}(\mathsf{C}^j(\mathsf{D}^h(\overline{M}), \overline{N'}), \mathsf{C}^i(\mathsf{J}(\overline{M}), \overline{N}) = \overline{L'}$$

since $\mathsf{C}^k \succ \mathsf{K}, \mathsf{J}, \mathsf{C}^j, \mathsf{C}^i, \mathsf{D}^h$ because $k >_{\mathsf{mul}} j, h, i$.

$f\ \ \mathtt{cut}(\lambda z.M, x.y(x, w.N)) \longrightarrow_f\ y(u.v.\mathtt{cut}(u, w.\mathtt{dec}(w, v, y.M)), w.\mathtt{cut}(\mathtt{inv}(w, y.\lambda z.M), x.N))$.
The derivation

$$\dfrac{\dfrac{z{:}C, \Gamma', y{:}(C{\supset}D){\supset}B \Rightarrow M{:}D}{\Gamma', y{:}(C{\supset}D){\supset}B \Rightarrow \lambda z.M{:}C{\supset}D}\ R{\supset} \quad \dfrac{x{:}C{\supset}D, \Gamma', w{:}B \Rightarrow N{:}E}{x{:}C{\supset}D, \Gamma', y{:}(C{\supset}D){\supset}B \Rightarrow y(x, w.N){:}E}\ L0{\supset}}{\Gamma', y{:}(C{\supset}D){\supset}B \Rightarrow \mathtt{cut}(\lambda z.M, x.y(x, w.N)){:}E}\ Cut$$

rewrites to

$$\dfrac{\dfrac{\mathcal{D}}{\Gamma', u{:}C, v{:}D{\supset}B \Rightarrow M'{:}D} \quad \dfrac{\mathcal{D}'}{\Gamma', w{:}B \Rightarrow N'{:}E}}{\Gamma', y{:}(C{\supset}D){\supset}B \Rightarrow y(u.v.M', w.N'){:}E}\ L{\supset}{\supset}$$

where $M' = \mathtt{cut}(u, w.\mathtt{dec}(w, v, y.M))$, $N' = \mathtt{cut}(\mathtt{inv}(w, y.\lambda z.M), x.N)$, $\mathcal{D}$ is the following derivation:

$$\dfrac{\dfrac{}{\Gamma', u{:}C, v{:}D{\supset}B \Rightarrow u{:}C}\ Ax \quad \dfrac{u{:}C, \Gamma', y{:}(C{\supset}D){\supset}B \Rightarrow M{:}D}{\Gamma', u{:}C, w{:}C, v{:}D{\supset}B \Rightarrow \mathtt{dec}(w, v, y.M){:}D}\ Dec}{\Gamma', u{:}C, v{:}D{\supset}B \Rightarrow \mathtt{cut}(u, w.\mathtt{dec}(w, v, y.M)){:}D}\ Cut$$

and $\mathcal{D}'$ is the following derivation:

$$\dfrac{\dfrac{\Gamma', y{:}A{\supset}B \Rightarrow \lambda z.M{:}C{\supset}D}{\Gamma', w{:}B \Rightarrow \mathtt{inv}(w, y.\lambda z.M){:}C{\supset}D}\ Inv \quad x{:}C{\supset}D, \Gamma', w{:}B \Rightarrow N{:}E}{\Gamma', w{:}B \Rightarrow \mathtt{cut}(\mathtt{inv}(w, y.\lambda z.M), x.N){:}E}\ Cut$$

Let $k = \{\!\{C{\supset}D, \Gamma', (C{\supset}D){\supset}B, E\}\!\}$ and $j = \{\!\{\Gamma', C, C, D{\supset}B, D\}\!\}$ and $h = \{\!\{C, \Gamma', (C{\supset}D){\supset}B, D\}\!\}$ and $i = \{\!\{C{\supset}D, \Gamma', B, E\}\!\}$.

We have

$$\overline{L} = \mathsf{C}^k(\mathsf{I}(\overline{M}), \mathsf{I}(\overline{N})) \gg_{mpo} \mathsf{K}(\mathsf{C}^j(\star, \mathsf{D}^h(\overline{M})), \mathsf{C}^i(\mathsf{J}(\mathsf{I}(\overline{M})), \overline{N})) = \overline{L'}$$

since $\mathsf{C}^k \succ \mathsf{K}, \star, \mathsf{J}, \mathsf{I}, \mathsf{C}^j, \mathsf{C}^i, \mathsf{D}^h$ because $k >_{\mathsf{mul}} j, i, h$.

g $\mathtt{cut}(x, y.z(y, w.M)) \longrightarrow_g z(y, w.\mathtt{cut}(x, y.M))$.
The derivation

$$\dfrac{\dfrac{}{\Gamma', x{:}A, z{:}A{\supset}B \Rightarrow x{:}A}\ Ax \quad \dfrac{x{:}A, y{:}A, w{:}B, \Gamma' \Rightarrow M{:}E}{x{:}A, y{:}A, z{:}A{\supset}B, \Gamma' \Rightarrow z(y, w.M){:}E}\ L0{\supset}}{\Gamma', x{:}A, z{:}A{\supset}B \Rightarrow \mathtt{cut}(x, y.z(y, w.M)){:}E}\ Cut$$

rewrites to

$$\dfrac{\dfrac{}{\Gamma', x{:}A, w{:}B \Rightarrow x{:}A}\ Ax \quad x{:}A, y{:}A, w{:}B, \Gamma' \Rightarrow M{:}E}{\dfrac{\Gamma', x{:}A, w{:}B \Rightarrow \mathtt{cut}(x, y.M){:}E}{\Gamma', x{:}A, z{:}A{\supset}B \Rightarrow z(y, w.\mathtt{cut}(x, y.M)){:}E}\ L0{\supset}}\ Cut$$

Let $k = \{\!\{A, A, A{\supset}B, \Gamma', E\}\!\}$ and $j = \{\!\{A, A, B, \Gamma', E\}\!\}$.
We have

$$\overline{L} = \mathsf{C}^k(\star, \mathsf{I}(\overline{M})) \gg_{mpo} \mathsf{I}(\mathsf{C}^j(\star, \overline{M})) = \overline{L'}$$

since $\mathsf{C}^k \succ \mathsf{I}, \mathsf{C}^j$ because $k >_{\mathsf{mul}} j$.

π $\mathtt{cut}(z(y, w.M), x.N) \longrightarrow_\pi z(y, w.\mathtt{cut}(M, x.\mathtt{inv}(w, z.N)))$.
The derivation

$$\dfrac{\dfrac{\Gamma', y{:}C, w{:}B \Rightarrow M{:}A}{\Gamma', y{:}C, z{:}C{\supset}B \Rightarrow z(y, w.M){:}A}\ L0{\supset} \quad x{:}A, \Gamma', y{:}C, z{:}C{\supset}B \Rightarrow N{:}E}{\Gamma', y{:}C, z{:}C{\supset}B \Rightarrow \mathtt{cut}(z(y, w.M), x.N){:}E}\ Cut$$

rewrites to

$$\dfrac{\dfrac{\Gamma', y{:}C, w{:}B \Rightarrow M{:}A \quad \dfrac{x{:}A, \Gamma', y{:}C, z{:}C{\supset}B \Rightarrow N{:}E}{x{:}A, \Gamma', y{:}C, w{:}B \Rightarrow \mathtt{inv}(w, z.N){:}E}\ Inv}{\Gamma', y{:}C, w{:}B \Rightarrow \mathtt{cut}(M, x.\mathtt{inv}(w, z.N)){:}E}\ Cut}{\Gamma', y{:}C, z{:}C{\supset}B \Rightarrow z(y, w.\mathtt{cut}(M, x.\mathtt{inv}(w, z.N))){:}E}\ L0{\supset}$$

Let $k = \{\!\{A, \Gamma', C, C{\supset}B, E\}\!\}$ and $j = \{\!\{A, \Gamma', C, B, E\}\!\}$.
We have

$$\overline{L} = \mathsf{C}^k(\mathsf{I}(\overline{M}), \overline{N}) \gg_{mpo} \mathsf{I}(\mathsf{C}^j(\overline{M}, \mathsf{J}(\overline{N})) = \overline{L'}$$

since $\mathsf{C}^k \succ \mathsf{I}, \mathsf{C}^j, \mathsf{J}$ because $k >_{\mathsf{mul}} j$.

ϕ $\mathtt{cut}(y(u.v.M, z.M'), x.N) \longrightarrow_\phi y(u.v.M, z.\mathtt{cut}(M', x.\mathtt{inv}(z, y.N)))$.
The derivation

$$\dfrac{\dfrac{\Gamma', v{:}C, u{:}D{\supset}B \Rightarrow M{:}D \quad \Gamma', z{:}B \Rightarrow M'{:}A}{\Gamma', y{:}(C{\supset}D){\supset}B \Rightarrow y(u.v.M, z.M'){:}A}\ L{\supset}{\supset} \quad x{:}A, \Gamma', y{:}(C{\supset}D){\supset}B \Rightarrow N{:}E}{\Gamma', y{:}(C{\supset}D){\supset}B \Rightarrow \mathtt{cut}(y(u.v.M, z.M'), x.N){:}E}\ Cut$$

rewrites to

$$\dfrac{\Gamma', v{:}C, u{:}D{\supset}B \Rightarrow M{:}D \quad \dfrac{\Gamma', z{:}B \Rightarrow M'{:}A \quad \dfrac{x{:}A, \Gamma', y{:}(C{\supset}D){\supset}B \Rightarrow N{:}E}{x{:}A, \Gamma', z{:}B \Rightarrow \mathtt{inv}(z, y.N){:}E}\,Of}{\Gamma', z{:}B \Rightarrow \mathtt{cut}(M', x.\mathtt{inv}(z, y.N)){:}E}\,Cut}{\Gamma', y{:}(C{\supset}D){\supset}B \Rightarrow y(u.v.M, z.\mathtt{cut}(M', x.\mathtt{inv}(z, y.N))){:}E}\,L{\supset}{\supset}$$

Let $k = \{\!\{A, \Gamma', (C{\supset}D){\supset}B, E\}\!\}$ and $j = \{\!\{A, \Gamma', B, E\}\!\}$.
We have

$$\overline{L} = \mathsf{C}^k(\mathsf{K}(\overline{M}, \overline{M'}), \overline{N}) \gg_{mpo} \mathsf{K}(\overline{M}, \mathsf{C}^j(\overline{M'}, \mathsf{J}(\overline{N}))) = \overline{L'}$$

since $\mathsf{C}^k \succ \mathsf{K}, \mathsf{C}^j, \mathsf{J}$ because $k >_{\mathsf{mul}} j$.

A $\mathtt{cut}(M, x.x(z, w.N)) \longrightarrow_A \mathtt{cut}(\mathtt{cut}(z, y.\mathtt{of}(M, y)), w.N)$.
The derivation

$$\dfrac{\Gamma', z{:}C \Rightarrow M{:}C{\supset}B \quad \dfrac{\Gamma', z{:}C, w{:}B \Rightarrow N{:}E}{\Gamma', z{:}C, x{:}C{\supset}B \Rightarrow x(z, w.N){:}E}\,L0{\supset}}{\Gamma', z{:}C \Rightarrow \mathtt{cut}(M, x.x(z, w.N)){:}E}\,Cut$$

rewrites to

$$\dfrac{\dfrac{}{\Gamma', z{:}C \Rightarrow z{:}C}\,Ax \quad \dfrac{\Gamma', z{:}C \Rightarrow M{:}C{\supset}B}{\Gamma', z{:}C, y{:}C \Rightarrow \mathtt{of}(M, y){:}B}\,Of}{\Gamma', z{:}C \Rightarrow \mathtt{cut}(z, y.\mathtt{of}(M, y)){:}B}\,Cut \quad \Gamma', z{:}C, w{:}B \Rightarrow N{:}E}{\Gamma', z{:}C \Rightarrow \mathtt{cut}(\mathtt{cut}(z, y.\mathtt{of}(M, y)), w.N){:}E}\,Cut$$

Let $k = \{\!\{\Gamma', C, C{\supset}B, E\}\!\}$ and $j = \{\!\{\Gamma', C, B, E\}\!\}$ and $i = \{\!\{\Gamma', C, C, B\}\!\}$. We have

$$\overline{L} = \mathsf{C}^k(\overline{M}, \mathsf{I}(\overline{N})) \gg_{mpo} \mathsf{C}^j(\mathsf{C}^i(\star, \mathsf{J}(\overline{M})), \overline{N}) = \overline{L'}$$

since $k >_{\mathsf{mul}} j, i$ and $\mathsf{C}^k \succ \star, \mathsf{J}, \mathsf{C}^j, \mathsf{C}^i$.

B $\mathtt{cut}(M, x.x(u.v.N, z.N')) \longrightarrow_B \mathtt{cut}(\mathtt{cut}(\lambda u.\mathtt{cut}(\lambda y'.\mathtt{inv}(y', y.\mathtt{of}(M, y)), v.N), y.\mathtt{of}(M, y)), z.N')$.
The derivation

$$\dfrac{\Gamma \Rightarrow M{:}(C{\supset}D){\supset}B \quad \dfrac{u{:}C, v{:}D{\supset}B, \Gamma \Rightarrow N{:}D \quad z{:}B, \Gamma \Rightarrow N'{:}E}{x{:}(C{\supset}D){\supset}B, \Gamma \Rightarrow x(u.v.N, z.N'){:}E}\,L{\supset}{\supset}}{\Gamma \Rightarrow \mathtt{cut}(M, x.x(u.v.N, z.N')){:}E}\,Cut$$

rewrites to

$$\dfrac{\dfrac{\mathcal{D}}{\Gamma \Rightarrow M'{:}C{\supset}D} \quad \dfrac{\Gamma \Rightarrow M{:}(C{\supset}D){\supset}B}{\Gamma, y{:}C{\supset}D \Rightarrow \mathtt{of}(M, y){:}B}\,Of}{\Gamma \Rightarrow \mathtt{cut}(M', y.\mathtt{of}(M, y)){:}B}\,Cut \quad z{:}B, \Gamma \Rightarrow N'{:}E}{\Gamma \Rightarrow \mathtt{cut}(\mathtt{cut}(M', y.\mathtt{of}(M, y)), z.N'){:}E}\,Cut$$

where $M' = \lambda u.\mathtt{cut}(\lambda y'.\mathtt{inv}(y', y.\mathtt{of}(M, y)), v.N)$ and $\mathcal{D}$ is the following derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \Rightarrow M : (C \supset D) \supset B}{\Gamma, y : C \supset D \Rightarrow \mathtt{of}(M, y) : B}\, Of}{\Gamma, u : C, y : C \supset D \Rightarrow \mathtt{of}(M, y) : B}\, (W)}{\Gamma, u : C, y' : D \Rightarrow \mathtt{inv}(y', y.\mathtt{of}(M, y)) : B}\, Inv}{\Gamma, u : C \Rightarrow \lambda y'.\mathtt{inv}(y', y.\mathtt{of}(M, y)) : D \supset B}\, R\supset \qquad u : C, v : D \supset B, \Gamma \Rightarrow N : D}{\Gamma, u : C \Rightarrow \mathtt{cut}(\lambda y'.\mathtt{inv}(y', y.\mathtt{of}(M, y)), v.N) : D}\, Cut}{\Gamma \Rightarrow \lambda u.\mathtt{cut}(\lambda y'.\mathtt{inv}(y', y.\mathtt{of}(M, y)), v.N) : C \supset D}\, R\supset$$

Let $k = \{\!\{(C \supset D) \supset B, \Gamma, E\}\!\}$ and $j = \{\!\{B, \Gamma, E\}\!\}$ and $i = \{\!\{\Gamma, C \supset D, B\}\!\}$ and $h = \{\!\{C, D \supset B, \Gamma, D\}\!\}$. We have

$$\begin{array}{c} \overline{L} = \\ \mathsf{C}^k(\overline{M}, \mathsf{K}(\overline{N}, \overline{N'})) \end{array} \gg_{mpo} \underbrace{\mathsf{C}^j(\mathsf{C}^i(\mathsf{I}(\mathsf{C}^h(\mathsf{I}(\mathsf{J}(\mathsf{J}(\overline{M}))), \overline{N})), \mathsf{J}(\overline{M})), \overline{N'})}_{= \overline{L'}}$$

since $\mathsf{C}^k \succ \mathsf{I}, \mathsf{J}, \mathsf{C}^j, \mathsf{C}^i, \mathsf{C}^h$ because $k >_{\mathsf{mul}} j, i, h$.

□

Corollary 1 (Strong Normalisation). *System* gs *is strongly normalising on typed terms.*

Proof: This is a consequence of Theorem 1 and Remark 3. □

Corollary 2. *Rules Inv, Of, Dec, and Cut are logically admissible in the system of Definition 4.*

Proof: Every term with an auxiliary constructor is reducible by system gs. □

6 Variants of reduction systems

We investigate in this section some variants of the cut-elimination system presented in Section 3.

We discuss in Section 6.1 the rules of Kind_3, noticing that the $\mathtt{of}(\_, \_)$-constructor is only introduced by the reductions of gs in order to include η-conversion in the system. We present two variations without η-conversion, called system rs and system ars, that no longer use the $\mathtt{of}(\_, \_)$-constructor.

Without η-conversion, the only critical pairs of those variations are between the rules of Kind_1 and those of Kind_2, so in Section 6.2, which only concerns rules of Kind_1 and Kind_2, we present two ways of removing those critical pairs, i.e. of making systems rs and ars orthogonal.

All the systems presented in this paper can be summarised in the following table:

of, inv and dec	cut = (Kind$_1$ + Kind$_2$) +	Kind$_3$	Whole system
oid	cegs = Table 4 +	Table 5	gs
oid	cers = Table 4 +	Table 6	rs
oid	cears = Table 4 +	Table 7	ars
oid	cecbn = Table 8 +	(Table 6 or Table 7)	cbn
oid	cecbv = Table 9 +	(Table 6 or Table 7)	cbv

6.1 Avoiding the of-constructor

In this section we remove η-expansion from the reduction system so that the $\mathtt{of}(\_,\_)$-constructor is no more used by the cut elimination rules. We obtain two variants, depending on whether we want variables to behave like their η-expansions or we want the elimination of a cut with a variable to be simpler and closer to renaming.

The rules A and B of system gs introduce the $\mathtt{of}(\_,\_)$-constructor to model η-expansion, turning the first argument of the cut into an abstraction.

Theorem 2. *Rule A (resp. B) can be factorised into an η-expansion followed by rule C (resp. D) below:*

$$\begin{array}{ll}\mathtt{cut}(\lambda y.M, x.x(z, w.N)) & \longrightarrow_C \ \mathtt{cut}(\mathtt{cut}(z, y.M), w.N)\\ \mathtt{cut}(\lambda y.M, x.x(u.v.N', w.N)) & \longrightarrow_D \ \mathtt{cut}(\mathtt{cut}(\lambda u.\mathtt{cut}(\lambda z.\mathtt{inv}(z, y.M), v.N'), y.M), w.N)\end{array}$$

Proof:

$$\begin{array}{lll}(\text{Rule } A) & & \mathtt{cut}(M, x.x(z, w.N))\\ & \longrightarrow_\eta & \mathtt{cut}(\lambda y.\mathtt{of}(M, y), x.x(z, w.N))\\ & \longrightarrow_C & \mathtt{cut}(\mathtt{cut}(z, y.\mathtt{of}(M, y)), w.N)\\ (\text{Rule } B) & & \mathtt{cut}(M, x.x(u.v.N', w.N))\\ & \longrightarrow_\eta & \mathtt{cut}(\lambda y.\mathtt{of}(M, y), x.x(u.v.N', w.N))\\ & \longrightarrow_D & \mathtt{cut}(\mathtt{cut}(\lambda u.\mathtt{cut}(\lambda z.\mathtt{inv}(z, y.\mathtt{of}(M, y)), v.N'), y.\mathtt{of}(M, y)), w.N)\end{array}$$

□

Note that the η-expansion of an abstraction reduces, by direct elimination of the $\mathtt{of}(\_,\_)$, to the abstraction itself:

$$\lambda y.M \longrightarrow_\eta \ \lambda x.\mathtt{of}(\lambda y.M, x) \longrightarrow_{\mathsf{o2}} \ \lambda x.\{x/y\}M =_\alpha \lambda y.M \text{ with } x \notin FV(M)$$

This justifies the following theorem:

Theorem 3. *Rules C and D can be respectively derived from rules A and B using system* oid.

Proof:

$$\begin{array}{lll}(\text{Rule } C) & & \mathtt{cut}(\lambda y.M, x.x(z, w.N))\\ & \longrightarrow_A & \mathtt{cut}(\mathtt{cut}(z, w'.\mathtt{of}(\lambda y.M, w')), w.N)\\ & \longrightarrow_{\mathsf{o2}} & \mathtt{cut}(\mathtt{cut}(z, w'.\{w'/y\}M), w.N)\\ & =_\alpha & \mathtt{cut}(\mathtt{cut}(z, y.M), w.N)\\ (\text{Rule } D) & & \mathtt{cut}(\lambda y.M, x.x(u.v.N', w.N))\\ & \longrightarrow_B & \mathtt{cut}(\mathtt{cut}(\lambda u.\mathtt{cut}(\lambda z.\mathtt{inv}(z, w'.\mathtt{of}(\lambda y.M, w')), v.N'), z'.\mathtt{of}(\lambda y.M, z')), w.N)\\ & \longrightarrow^*_{\mathsf{o2}} & \mathtt{cut}(\mathtt{cut}(\lambda u.\mathtt{cut}(\lambda z.\mathtt{inv}(z, w'.\{w'/y\}M), v.N'), z'.\{z'/y\}M), w.N)\\ & =_\alpha & \mathtt{cut}(\mathtt{cut}(\lambda u.\mathtt{cut}(\lambda z.\mathtt{inv}(z, y.M), v.N'), y.M), w.N)\end{array}$$

□

Similarly, direct elimination of the $\mathtt{of}(\_,\_)$-constructor is allowed by rule $\mathtt{o1}$ in the case of a variable ($y \longrightarrow_\eta \lambda x.\mathtt{of}(y,x) \longrightarrow_{\mathtt{o1}} y(x,z.z)$ with $x \notin FV(M)$), so this suggests that two rules E and F, treating the case of a variable, can also be derived from rules A and B:

Theorem 4. *The following rules E and F can be respectively derived from A and B using system* gs*:*

$$\begin{array}{ll}
\mathtt{cut}(y, x.x(z,w.N)) & \longrightarrow_E \ y(z, w'.\mathtt{cut}(w', w.\mathtt{inv}(w', y.N))) \\
\mathtt{cut}(y, x.x(u.v.N', w.N)) & \longrightarrow_F \ y(u'.v'.\mathtt{cut}(u', u.P), w'.\mathtt{cut}(w', w.\mathtt{inv}(w', y.N))) \\
 & \quad \textit{where } P = \mathtt{dec}(u', v', y.\mathtt{cut}(\lambda y''.y(u.v.y'', z.z), v.N'))
\end{array}$$

Proof:

$$\begin{array}{lll}
(\text{Rule } E) & & \mathtt{cut}(y, x.x(z,w.N)) \\
 & \longrightarrow_A & \mathtt{cut}(\mathtt{cut}(z, y'.\mathtt{of}(y,y')), w.N) \\
 & \longrightarrow_{\mathtt{o1}} & \mathtt{cut}(\mathtt{cut}(z, y'.y(y', w'.w')), w.N) \\
 & \longrightarrow_g & \mathtt{cut}(y(z, w'.\mathtt{cut}(z, y'.w')), w.N) \\
 & \longrightarrow_b & \mathtt{cut}(y(z, w'.w'), w.N) \\
 & \longrightarrow_\pi & y(z, w'.\mathtt{cut}(w', w.\mathtt{inv}(w', y.N))) \\
(\text{Rule } F) & & \mathtt{cut}(y, x.x(u.v.N', w.N)) \\
 & \longrightarrow_B & \mathtt{cut}(\mathtt{cut}(\lambda u.\mathtt{cut}(L, v.N'), y'.\mathtt{of}(y,y')), w.N) \\
 & \longrightarrow^* & \mathtt{cut}(\mathtt{cut}(\lambda u.\mathtt{cut}(L', v.N'), y'.\mathtt{of}(y,y')), w.N) \\
 & \longrightarrow_{\mathtt{o1}} & \mathtt{cut}(\mathtt{cut}(\lambda u.\mathtt{cut}(L', v.N'), y'.y(y', w'.w')), w.N) \\
 & \longrightarrow^* & \mathtt{cut}(y(u'.v'.\mathtt{cut}(u', u.P), w'.w'), w.N) \\
 & \longrightarrow_\phi & y(u'.v'.\mathtt{cut}(u', u.P), w'.\mathtt{cut}(w', w.\mathtt{inv}(w', y.N)))
\end{array}$$

where the first $\longrightarrow^*$ is justified by

$$\begin{array}{lll}
L & = & \lambda y''.\mathtt{inv}(y'', w'.\mathtt{of}(y, w')) \\
 & \longrightarrow_{\mathtt{o1}} & \lambda y''.\mathtt{inv}(y'', w'.y(w', z.z)) \\
 & \longrightarrow_{\mathsf{i6}} & \lambda y''.y(y_1.y_2.y'', z.\mathtt{inv}(y'', w'.z)) \\
 & \longrightarrow_{\mathsf{i1}} & \lambda y''.y(y_1.y_2.y'', z.z) \qquad\qquad = L'
\end{array}$$

and the last $\longrightarrow^*$ is justified by

$$\begin{array}{ll}
 & \mathtt{cut}(\lambda u.\mathtt{cut}(L', v.N'), y'.y(y', w'.w')) \\
\longrightarrow_f & y(u'.v'.\mathtt{cut}(u', u.\mathtt{dec}(u', v', y.\mathtt{cut}(L', v.N'))), w'.\mathtt{cut}(\mathtt{inv}(w', y.\lambda u.\mathtt{cut}(L', v.N')), y'.w')) \\
\longrightarrow_b & y(u'.v'.\mathtt{cut}(u', u.\mathtt{dec}(u', v', y.\mathtt{cut}(L', v.N'))), w'.w') \\
= & y(u'.v'.\mathtt{cut}(u', u.P), w'.w')
\end{array}$$

□

Now, *by construction*, rules E and F make variables have the same functional behaviour as their η-expansion.

Notice also that the new rules C, D, E and F (together with rules π and ϕ) can now replace any use of rules A and B, thus forming a system, called cers, that is still complete for cut-elimination and makes no use of the $\mathtt{of}(\_,\_)$-constructor. We show in Table 6 only the cut reduction rules of Kind_3, in which cegs and cers differ, the rules of Kind_1 and Kind_2 being the same. System cegs can thus be seen as system cers to which η-expansion has been integrated by the use of the auxiliary constructor $\mathtt{of}(\_,\_)$.

The behaviour of functionals is interesting in G4ip, because it is a depth-bounded calculus. For instance, among all Church's numerals, only 0 and 1 can be represented in G4ip, so

Kind_3

$$\dfrac{\Gamma \Rightarrow y{:}(C{\supset}D){\supset}B \quad \dfrac{\Gamma, u{:}C, v{:}D{\supset}B \Rightarrow N'{:}D \quad \Gamma, w{:}B \Rightarrow N{:}E}{\Gamma, x{:}(C{\supset}D){\supset}B \Rightarrow x(u.v.N', w.N){:}E}\, L{\supset}{\supset}}{\Gamma \Rightarrow \mathtt{cut}(y, x.x(u.v.N', w.N)){:}E}\, Cut$$

rewrites to

$$\dfrac{\dfrac{}{\Gamma', u'{:}C, v'{:}D{\supset}B \Rightarrow u'{:}C}\, Ax \quad \dfrac{\dfrac{L}{\Gamma', u{:}C, u'{:}C, v'{:}D{\supset}B \Rightarrow P'{:}D}\, Cut \quad \dfrac{L'}{\Gamma', w'{:}B \Rightarrow M{:}E}}{\Gamma', u'{:}C, v'{:}D{\supset}B \Rightarrow \mathtt{cut}(u', u.P'){:}D}\, Cut}{\Gamma', y{:}(C{\supset}D){\supset}B \Rightarrow y(u'.v'.\mathtt{cut}(u', u.P'), w'.L'){:}E}\, L{\supset}{\supset}$$

where $M = \mathtt{cut}(w', w.\mathtt{inv}(w', y.N))$, the premiss L is constructed as follows:

$$\dfrac{\Gamma', u{:}C, u'{:}C, v'{:}D{\supset}B \Rightarrow v'{:}D{\supset}B \quad \dfrac{\Gamma', y{:}(C{\supset}D){\supset}B, u{:}C, v{:}D{\supset}B \Rightarrow N'{:}D}{\Gamma', u{:}C, v{:}D{\supset}B, u'{:}C, v'{:}D{\supset}B \Rightarrow \mathtt{dec}(u', v', y.N'){:}D}\, Dec}{\Gamma', u{:}C, u'{:}C, v'{:}D{\supset}B \Rightarrow \mathtt{cut}(v', v.\mathtt{dec}(u', v', y.N')){:}D}\, Cut$$

and the second premiss L' is constructed as follows:

$$\dfrac{\Gamma', w'{:}B \Rightarrow w'{:}B \quad \dfrac{\Gamma', y{:}(C{\supset}D){\supset}B, w{:}B \Rightarrow N{:}E}{\Gamma', w'{:}B, w{:}B \Rightarrow \mathtt{inv}(w', y.N){:}E}\, Inv}{\Gamma', w'{:}B \Rightarrow \mathtt{cut}(w', w.\mathtt{inv}(w', y.N)){:}E}\, Cut$$

Let $k = \{\!\{\Gamma, (C{\supset}D){\supset}B, (C{\supset}D){\supset}B, E\}\!\}$ and $i = \{\!\{\Gamma', B, B, E\}\!\}$ and $j = \{\!\{\Gamma', C, C, D{\supset}B, D\}\!\}$ and $h = \{\!\{\Gamma', C, D{\supset}B, C, D{\supset}B, D\}\!\}$ and $l = \{\!\{\Gamma', (C{\supset}D){\supset}B, C, D{\supset}B, E\}\!\}$ and
We have

$$\overline{\mathtt{cut}(y, x.x(u.v.N', w.N))} \quad = \quad \mathsf{C}^k(\star, \mathsf{K}(\overline{N'}, \overline{N})) \qquad \gg_{mpo} \qquad \mathsf{K}(\mathsf{C}^j(\star, \mathsf{C}^h(\star, \mathsf{D}^l(\overline{N'}))), \mathsf{C}^i(\star, \mathsf{J}(\overline{N}))) \quad = \quad \overline{y(u'.v'.\mathtt{cut}(u', u.P'), w'.\mathtt{cut}(w', w.\mathtt{inv}(w', y.N)))}$$

since $\mathsf{C}^k \succ \mathsf{K}, \mathsf{C}^j, \mathsf{C}^h, \mathsf{D}^l, \mathsf{C}^i$ because $k >_{mul} i, j, h, l$.

6.2 Orthogonal systems

In this section we suggest two ways of restricting the rules of Kind_1 and Kind_2 to make systems rs and ars orthogonal, and hence confluent.

In the restricted systems gs and ars there is an overlap between the right and left propagation sub-systems, i.e. there is a critical pair between any rule in $\{a, b, c, d, e\}$ and any rule any in $\{\pi, \phi\}$. This is shown in the following table, where column headers represent the different cases concerning the first premiss of the cut, while row headers represent the different cases for the second one (marking inside parentheses the status of the cut-type).

	Axiom	$R\supset$	$L0\supset$	$L\supset\supset$
Axiom (Principal)	a	a	$a\pi$	$a\phi$
Axiom (Non-Principal)	b	b	$b\pi$	$b\phi$
$R\supset$	c	c	$c\pi$	$c\phi$
$L0\supset$ (Non-Principal, Non-Auxiliary)	d	d	$d\pi$	$d\phi$
$L\supset\supset$ (Non-Principal)	e	e	$e\pi$	$e\phi$
$L0\supset$ (Non-Principal, Auxiliary)	g	f	π	ϕ
$L0\supset$ (Principal)	E	C	π	ϕ
$L\supset\supset$ (Principal)	F or G	D	π	ϕ

This overlap is well-known in sequent calculus, and corresponds to the choice of whether to push a cut into the proof of its left premiss or into the proof of its right premiss. The former corresponds to a *call-by-value* strategy and the latter corresponds to a *call-by-name* strategy.

Since the overlap only concerns cut reduction rules of Kind_1 and Kind_2, we shall only study those kinds of rules and leave the rules of Kind_3 as they are in system cers or in system cears since both are possible.

Call-by-name One way to make the system orthogonal is to give preference to rules *a-b-c-d-e* over rules π-ϕ, thus restricted to the case when N is an x-covalue Q, i.e. is of the form $x(y, w.N)$ or $x(u.v.M, w.N)$. We show the resulting reduction rules of Kind_1 and Kind_2 in Table 8.

Kind_1		
$\mathtt{cut}(M, x.x)$	$\longrightarrow_a$	M
$\mathtt{cut}(M, x.y)$	$\longrightarrow_b$	y
$\mathtt{cut}(M, x.\lambda y.N)$	$\longrightarrow_c$	$\lambda y.\mathtt{cut}(M, x.N)$
$\mathtt{cut}(M, x.y(z, w.N))$	$\longrightarrow_d$	$y(z, w.\mathtt{cut}(\mathtt{inv}(w, y.M), x.N))$
$\mathtt{cut}(M, x.y(u.v.N', w.N))$	$\longrightarrow_e$	$y(u.v.\mathtt{cut}(\mathtt{dec}(u, v, y.M), x.N'), w.\mathtt{cut}(\mathtt{inv}(w, y.M), x.N))$
$\mathtt{cut}(\lambda z.M, x.y(x, w.N))$	$\longrightarrow_f$	$y(u.v.\mathtt{cut}(u, z.\mathtt{dec}(u, v, y.M)), w.\mathtt{cut}(\mathtt{inv}(w, y.\lambda z.M), x.N))$
$\mathtt{cut}(z, x.y(x, w.N))$	$\longrightarrow_g$	$y(z, w.\mathtt{cut}(z, x.N))$
Kind_2		
$\mathtt{cut}(y(z, w.M), x.Q)$	$\longrightarrow_\pi$	$y(z, w.\mathtt{cut}(M, x.\mathtt{inv}(w, y.Q)))$
$\mathtt{cut}(y(u.v.M', w.M), x.Q)$	$\longrightarrow_\phi$	$y(u.v.M', w.\mathtt{cut}(M, x.\mathtt{inv}(w, y.Q)))$

Table 8. Cut Elimination Rules in system cecbn (Kind_1 and Kind_2)

Notice that in order to reduce a term like $\mathtt{cut}(M, x.y(x, w.N))$, there is no choice other than left-propagation (rules π and ϕ) until a similar redex is found in which M is a value, and then only rules f or g can be applied.

	Axiom	$R\supset$	$L0\supset$	$L\supset\supset$
Axiom (Principal)	a	a	a	a
Axiom (Non-Principal)	b	b	b	b
$R\supset$	c	c	c	c
$L0\supset$ (Non-Principal, Non-Auxiliary)	d	d	d	d
$L\supset\supset$ (Non-Principal)	e	e	e	e
$L0\supset$ (Non-Principal, Auxiliary)	g	f	π	ϕ
$L0\supset$ (Principal)	E	C	π	ϕ
$L\supset\supset$ (Principal)	F (G)	D	π	ϕ

Call-by-value Alternatively, preference might be given to rules π and ϕ, which we can formalise as restricting rules *a-b-c-d-e* to the case when M is a value V (variable or abstraction). We show the resulting reduction rules of Kind_1 and Kind_2 in Table 9.

Kind_1		
$\mathtt{cut}(V, x.x)$	$\longrightarrow_a$	V
$\mathtt{cut}(V, x.y)$	$\longrightarrow_b$	y
$\mathtt{cut}(V, x.\lambda y.N)$	$\longrightarrow_c$	$\lambda y.\mathtt{cut}(V, x.N)$
$\mathtt{cut}(V, x.y(z, w.N))$	$\longrightarrow_d$	$y(z, w.\mathtt{cut}(\mathtt{inv}(w, y.V), x.N))$
$\mathtt{cut}(V, x.y(u.v.N', w.N))$	$\longrightarrow_e$	$y(u.v.\mathtt{cut}(\mathtt{dec}(u, v, y.V), x.N'), w.\mathtt{cut}(\mathtt{inv}(w, y.V), x.N))$
$\mathtt{cut}(\lambda z.M, x.y(x, w.N))$	$\longrightarrow_f$	$y(u.v.\mathtt{cut}(u, z.\mathtt{dec}(u, v, y.M)), w.\mathtt{cut}(\mathtt{inv}(w, y.\lambda z.M), x.N))$
$\mathtt{cut}(z, x.y(x, w.N))$	$\longrightarrow_g$	$y(z, w.\mathtt{cut}(z, x.N))$
Kind_2		
$\mathtt{cut}(y(z, w.M), x.N)$	$\longrightarrow_\pi$	$y(z, w.\mathtt{cut}(M, x.\mathtt{inv}(w, y.N)))$
$\mathtt{cut}(y(u.v.M', w.M), x.N)$	$\longrightarrow_\phi$	$y(u.v.M', w.\mathtt{cut}(M, x.\mathtt{inv}(w, y.N)))$

Table 9. Cut Elimination Rules in system cecbv (Kind_1 and Kind_2)

This choice is particularly coherent because the two rules of right-propagation f and g only apply to cuts whose first argument is a value. This suggests that **G4ip** has an inherent *call-by-value* flavour, echoing the idea that it is somehow based on the call-by-value sequent calculus **LJQ**. Indeed, completeness of **LJQ** gives a short proof of the completeness of **G4ip** [DL06].

	Axiom	$R\supset$	$L0\supset$	$L\supset\supset$
Axiom (Principal)	a	a	π	ϕ
Axiom (Non-Principal)	b	b	π	ϕ
$R\supset$	c	c	π	ϕ
$L0\supset$ (Non-Principal, Non-Auxiliary)	d	d	π	ϕ
$L\supset\supset$ (Non-Principal)	e	e	π	ϕ
$L0\supset$ (Non-Principal, Auxiliary)	g	f	π	ϕ
$L0\supset$ (Principal)	E	C	π	ϕ
$L\supset\supset$ (Principal)	F (G)	D	π	ϕ

We finish this section by stating the following property of the orthogonal systems presented here.

Theorem 5. *Reduction systems* cbn *and* cbv *are confluent, hence normal forms are unique.*

Proof: Systems cbn and cbv can be seen as particular *orthogonal CRS*, so they enjoy confluence (see [vOvR94] for details). □

7 Another proof of strong normalization

We present here a second proof of strong normalization.

Lemma 2. *If* $N \in SN_{\mathsf{gs}}$*, then* $\mathtt{inv}(x, y.N) \in SN_{\mathsf{gs}}$.

Proof. By induction on $\langle N, |N| \rangle$ w.r.t the lexicographic order $\langle \rightarrow, > \rangle$.

Lemma 3. *If* $N \in SN_{\mathsf{gs}}$*, then* $\mathtt{of}(N, x) \in SN_{\mathsf{gs}}$.

Proof. By induction on $\langle N, |N| \rangle$ w.r.t the lexicographic order $\langle \rightarrow, > \rangle$.

Lemma 4. *Suppose* $\Gamma, z{:}A \vdash N : E$ *and* $N \in SN_{\mathsf{gs}}$*. Then*

1. *If* $A = (C \supset D) \supset B$*, and* x, y *are fresh, then* $\mathtt{dec}(x, y, z.N) \in SN_{\mathsf{gs}}$*;*
2. *Let* $\Gamma \vdash M : A$ *with* $M \in SN_{\mathsf{gs}}$*; then* $\mathtt{cut}(M, z.N) \in SN_{\mathsf{gs}}$.

Proof. By simultaneous induction on tuples $\langle (\{\!\{\Gamma, A, E\}\!\}, N, M \rangle$ w.r.t. the lexicographic order $\langle >_{multiset}, \rightarrow, \rightarrow \rangle$.

Theorem 6. *If* $\Gamma \vdash M : A$*, then* $M \in SN_{\mathsf{gs}}$.

Proof. By induction on the structure of M using Lemmas 2, 3 and 4.

Theorem 7. *If* $\Gamma \vdash M : A$*, then* $M \in SN_{\mathsf{rs}}$.

Proof. This is evident since every rs-reduction step can be simulated by a non-empty sequence in gs.

Theorem 8. *If* $\Gamma \vdash M : A$*, then* $M \in SN_{\mathsf{ars}}$.

Proof. One can do the same proof as in Theorem 6 by remarking that rule G also decreases the measure of sequents.

8 Conclusion

This paper defines various proof-term calculi for the depth-bounded intuitionistic sequent calculus of Hudelmaier. Using standard techniques of rewriting, we prove subject-reduction and strong normalisation for all of them, so *Cut*-admissibility turns out to be a corollary. The cbn and cbv systems presented in this paper are also orthogonal, which guarantees confluence (and uniqueness of normal forms).

Some relations between G4ip and other calculi for intuitionistic logic are studied in [DL06]. Our approach also suggests how to obtain a term calculus for G4ip but (as in λ-calculus) with *implicit*, rather than explicit, operators to model cut-elimination. This would bring our calculus closer to that of Matthes [Mat02], and with a strong normalising cut-elimination procedure. As mentioned in the introduction, defining a denotational semantics for our calculi as well as investigating the connexions with the simply-typed λ-calculus would reveal more properties of the proofs in G4ip. This is left for further investigations.

References

[BN98] F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.

[DL06] R. Dyckhoff and S. Lengrand. **LJQ**, a strongly focused calculus for intuitionistic logic, 2006. Submitted. Available at `http://www.pps.jussieu.fr/~lengrand/Work/Papers.html`.

[DM79] N. Dershowitz and Z. Manna. Proving termination with multiset orderings. *Communications of the ACM*, 22(8):465–476, 1979.

[DN00] R. Dyckhoff and S. Negri. Admissibility of structural rules for contraction-free systems of intuitionistic logic. *The Journal of Symbolic Logic*, 65(4):1499–1518, 2000.

[Dyc92] R. Dyckhoff. Contraction-free sequent calculi for intuitionistic logic. *The Journal of Symbolic Logic*, 57(3):795–807, 1992.

[Hud89] J. Hudelmaier. *Bounds for Cut Elimination in Intuitionistic Logic.* PhD thesis, Universität Tübingen, 1989.

[Hud92] J. Hudelmaier. Bounds on cut-elimination in intuitionistic propositional logic. *Archive for Mathematical Logic*, 31:331–354, 1992.

[KL80] S. Kamin and J.-J. Lévy. Attempts for generalizing the recursive path orderings. Handwritten paper, University of Illinois, 1980.

[LSS91] P. Lincoln, A. Scedrov, and N. Shankar. Linearizing intuitionistic implication. In *Proc. of the Sixth Annual IEEE Symposium on Logic in Computer Science*, pages 51–62, Amsterdam, The Netherlands, 1991.

[Mat02] R. Matthes. Contraction-aware lambda-calculus, 2002. Seminar at Oberwolfach.

[O'D77] M. J. O'Donnell. *Computing in Systems Described by Equations*, volume 58 of *Lecture Notes in Computer Science*. Springer-Verlag, 1977.

[ORK05] J. Otten, T. Raths, and C. Kreitz. The ILTP Library: Benchmarking automated theorem provers for intuitionistic logic. In B. Beckert, editor, *International Conference TABLEAUX-2005*, volume 3702 of *Lecture Notes in Artificial Intelligence*, pages 333–337. Springer Verlag, 2005.

[Pit92] A. M. Pitts. On an interpretation of second order quantification in first-order intuitionistic propositional logic. *Journal of Symbolic Logic*, 57:33–52, 1992.

[TS00] A. S. Troelstra and H. Schwichtenberg. *Basic Proof Theory.* Cambridge University Press, 2000.

[Ves99] R. Vestergaard. Revisiting Kreisel: A computational anomaly in the Troelstra-Schwichtenberg **G3i** system, March 1999. Available at `http://www.cee.hw.ac.uk/~jrvest/`.

[Vor70] N. N. Vorob'ev. A new algorithm for derivability in the constructive propositional calculus. *American Mathematical Society Translations*, 94(2):37–71, 1970.

[vOvR94] V. van Oostrom and F. van Raamsdonk. Weak orthogonality implies confluence: the higher-order case. In A. Nerode and Y. Matiyasevich, editors, *Proceedings of the 3rd International Symposium on Logical Foundations of Computer Science*, volume 813 of *Lecture Notes in Computer Science*, pages 379–392. Springer-Verlag, July 1994.