

Cut elimination, substitution and normalisation

Roy Dyckhoff

Abstract We present a proof (of the main parts of which there is a formal version, checked with the Isabelle proof assistant) that, for a **G3**-style calculus covering all of intuitionistic zero-order logic, with an associated term calculus, and with a particular strongly normalising and confluent system of cut-reduction rules, every reduction step has, as its natural deduction translation, a sequence of zero or more reduction steps (detour reductions, permutation reductions or simplifications). This complements and (we believe) clarifies earlier work by (e.g.) Zucker and Pottinger on a question raised in 1971 by Kripke.

Keywords: Intuitionistic logic, minimal logic, sequent calculus, natural deduction, cut-elimination, substitution, normalisation

1 Introduction

It is well-known that, in intuitionistic logic, sequent calculus derivations (with or without *Cut*) are recipes for constructing natural deductions, and that, by the Curry-Howard correspondence, with care about variable discharge conventions, one can represent both the former and the latter using typed lambda terms. Natural deduction terms may, by various standard reductions, be reduced; but there are many sequent calculi S , reduction systems R for S and reduction strategies for R , including but not limited to those given by Gentzen. We present here, for a complete single-succedent sequent calculus **G3ip** (roughly that in [20] and [13]), a reduction system (of 32 rules) for cut-elimination, with the virtues that (a) it is strongly normalising; (b) it is confluent; (c) it is explicitly given using a term notation; and (d) it allows a homomorphism (as described below) from cut-elimination to normalisation.

Kripke asked [11] about the relation between cut-elimination and normalisation. Troelstra and van Dalen in [19] (p 565) comment that “The combinatorial relationship between normalization and cut-elimination has been investigated by Zucker (1974) and Pottinger (1977). Normalization corresponds to cut-elimination under a homomorphic mapping provided the basic cut-elimination steps are suitably chosen. Recently a still better correspondence has been achieved by Diller and Unterhalt.”

It is however not clear what the suitable choices of the “basic cut-elimination steps” should be. Diller has written (19 December 2012) that “I am very sorry that I cannot point at a publication of Unterhalt’s thesis or at a paper published by the two of us. After quarter of a century since Unterhalt’s PhD thesis (1986), I cannot even give a concise explanation of what the progress of Unterhalt’s work was in comparison to the work or methods of Zucker and Pottinger. I think that the central points are contained in [19]. Unterhalt mainly studies . . . semantics of E-logic (. . .), but he also establishes a transfer to cut-elimination and normalization.”

Zucker [25], using Gentzen’s cut-elimination steps and an innermost-first strategy, gave a partial answer, but had difficulties with disjunction, including a failure of strong normalisation. His calculus has explicit rules of *Weakening* and *Contraction* and context-splitting inference rules. Pottinger [15] gave a positive answer covering disjunction; but, as pointed out by Urban [21], the notion of normality for the sequent calculus proof terms does not coincide with cut-freedom, and this renders Pottinger’s answer “subtly” defective—in our view overtly defective, despite Pottinger’s claim that the difference is “trivial”. (Moreover, the closest system in [15] to a conventional sequent calculus is his H_L ; but, although it is complete for derivability of formulae, it does not admit *Contraction*; it does not derive, for example, the sequent $p \Rightarrow p \wedge p$. Nor does it admit *Weakening* or derive the sequent $p \wedge p \Rightarrow p \wedge p$. There is a section explaining what one might do if *Contraction* is added as a primitive rule, with no explanation of how the cut-reduction rules might change—one is reminded of Gentzen’s difficulties with *Contraction* and his avoidance thereof with his *Mix* rule.)

Von Plato [14] and (with Negri) [13] consider related issues, with a focus not on cut reduction and normalisation steps but on obtaining an isomorphism between sequent calculus and natural deduction, achieved by using generalised elimination rules in the latter. We see this as a solution to a different problem.

The purpose of this paper is thus to clarify matters in our preferred context, namely the sequent calculus **G3ip** (with context-sharing inference rules and without explicit rules of *Weakening* and *Contraction*) and a standard natural deduction calculus from [8, 16]. Such sequent calculi (widely studied in [13]) correspond better than others (such as those of [8, 25, 15]) to calculi used for root-first proof search, either as sequent calculi or (inverted) as tableau calculi. We also choose to use a term notation (with an appropriate binding mechanism) to allow the concise presentation of reduction rules: this has the extra virtue of simplifying automation and verification.

Urban’s solution also uses, for representing sequent derivations, a term notation, deriving from his work in classical logic [22], with names and co-names. His notation improves on Pottinger’s and Zucker’s, but his result (that there is such a homomorphism) applies only to the $(\supset, \wedge, \forall)$ -fragment of the logic.

Borisavljević [3] gives a detailed explanation of Zucker’s difficulties and proposes a solution (for intuitionistic predicate logic) using generalised elimination rules (from [14, 13]), concluding that “the problem in connections of conversions from the full systems (with \forall and \exists) δ and \mathcal{N} is the consequence of the different forms of elimination rules for \wedge, \supset and \forall on the one side, and \vee and \exists on the other side, in the system \mathcal{N} ” (where \mathcal{N} is Zucker’s natural deduction system). We prefer not to adopt a natural deduction system with generalised elimination rules but to use the original systems of Gentzen [8] and Prawitz [16].

Kikuchi [10] treats an aspect of the relationship between normalisation and cut-elimination, but only for implicational logic, and with a very different goal: the simulation of normalisation by cut-elimination, i.e. the lifting of reduction steps in natural deduction back to cut reduction steps (from which one can infer strong normalisation of natural deduction from the corresponding property of the sequent calculus)..

Thus, we present here a treatment of this issue for **all** of intuitionistic zero-order logic, using a standard natural deduction calculus, a standard sequent calculus and a standard notation for terms, and without Pottinger-style defects, allowing a clear understanding of exactly what reductions are required in the natural deduction calculus and in the sequent calculus for the following to hold: let L and L' be sequent derivations so that $L \rightsquigarrow^* L'$ by some sequence of cut-elimination steps; then $N \rightsquigarrow^* N'$, where $N \equiv \phi(L)$ and $N' \equiv \phi(L')$ are the natural deductions constructed from the recipes L and L' by means of the Gentzen-Prawitz translation ϕ . Like others, we consider this to give a *homomorphism* from one reduction system to another.

No claim is made about the converse; ensuring that a cut-reduction system can simulate beta-reduction is tricky. See [10], and also [4] for a solution involving a restricted sequent calculus, Herbelin’s **LJT**.

Our result is for **all** the connectives of intuitionistic zero-order logic, including disjunction. Given that there are examples in (for example) [21] illustrating the difficulty with disjunction, this may be surprising. The solution is given by the complex reduction rules such as ((6.18)) and ((6.20)). A computer-checked verification of the results is available [5], using Nominal Isabelle [23]. Work extending both the theory and the formal verification to first-order logic is not yet undertaken: no major difficulties are anticipated.

An extended abstract of an earlier version of this paper appeared as [6].

2 Technical Background

Where distinct meta-variables x, y , etc are used, they stand for distinct variables except where explicitly stated or indicated otherwise. We will use x and x' , for example, to indicate two variables which can but need not be distinct from each other. The symbol i will be understood as ranging over $\{1, 2\}$. *Atoms* are proposition variables p, q, \dots or \perp ; *formulae* A, B, C, D, E, \dots are built up from atoms using implication, conjunction and disjunction. *Contexts* Γ are (as usual) sets of expressions $x : A$ where each x is associated with at most one formula A .

2.1 Natural Deduction, in Logistic Style

We present the typing rules for typed lambda calculus in natural deduction format, but using Gentzen's logistic style.

$$\begin{array}{c}
\frac{\Gamma \Rightarrow N : \perp}{\Gamma \Rightarrow ef(N) : C} \perp E \qquad \frac{}{x : A, \Gamma \Rightarrow x : A} Ax \\
\frac{\Gamma \Rightarrow N : A \supset B \quad \Gamma \Rightarrow N' : A}{\Gamma \Rightarrow ap(N, N') : B} \supset E \qquad \frac{x : A, \Gamma \Rightarrow N : B}{\Gamma \Rightarrow \lambda x. N : A \supset B} \supset I \\
\frac{\Gamma \Rightarrow N : A_1 \wedge A_2}{\Gamma \Rightarrow pri(N) : A_i} \wedge E_i \qquad \frac{\Gamma \Rightarrow N_1 : A_1 \quad \Gamma \Rightarrow N_2 : A_2}{\Gamma \Rightarrow (N_1, N_2) : A_1 \wedge A_2} \wedge I \\
\frac{\Gamma \Rightarrow N : A \vee A' \quad x : A, \Gamma \Rightarrow N' : C \quad x' : A', \Gamma \Rightarrow N'' : C}{\Gamma \Rightarrow D(N, x.N', x'.N'') : C} \vee E \qquad \frac{\Gamma \Rightarrow N : A_i}{\Gamma \Rightarrow in_i(N) : A_1 \vee A_2} \vee I_i
\end{array}$$

We write $ap(N, N')$ just as NN' , or maybe as (NN') to avoid ambiguity; sometimes however we use the original form for emphasis. D is short for “decide”. When (for some context Γ , term N and formula A) one can infer that $\Gamma \Rightarrow N : A$, we also say that (in the context Γ) N has *type* A .

2.2 Reductions of lambda terms

We use $[N/x]N'$ to indicate the result of *substituting* the term N for free occurrences of the variable x in the term N' . It is, as usual, capture-avoiding: bound variables are, if necessary, renamed to avoid capture. The order N, x, N' of the sub-expressions in this notation is deliberately chosen to match the order in which they appear in the premisses of the (admissible, by induction on the structure of N') typing rule for the operation:

$$\frac{\Gamma \Rightarrow N : C \quad \Gamma, x : C \Rightarrow N' : B}{\Gamma \Rightarrow [N/x]N' : B} \text{ Subs.}$$

Lemma 2.1. *Let N, N' and N'' be terms and let x and y be distinct variables, with x not free in N'' . Then*

$$[N''/y] [N'/x] N \equiv [[N''/y]N'/x] [N''/y] N.$$

Proof. See Barendregt [2, p. 27]; the proof extends without difficulty to cover all the connectives. \square

For reference, we give some standard reductions (“detour reductions” (2.1) to (2.3), “permutation reductions” (2.4) to (2.7), \perp -reductions (2.8) to (2.10) and a “simplification” (2.11)) of lambda terms:

$$(\lambda x.N)N' \rightsquigarrow [N'/x]N \quad (2.1)$$

$$pr_i((N_1, N_2)) \rightsquigarrow N_i \quad (2.2)$$

$$D(in_i(N), x_1.N_1, x_2.N_2) \rightsquigarrow [N/x_i]N_i \quad (2.3)$$

$$D(N, x_1.N_1, x_2.N_2)N' \rightsquigarrow D(N, x_1.N_1N', x_2.N_2N') \quad (2.4)$$

$$pr_i(D(N, x_1.N_1, x_2.N_2)) \rightsquigarrow D(N, x_1.pr_i(N_1), x_2.pr_i(N_2)) \quad (2.5)$$

$$D(D(N, x_1.N_1, x_2.N_2), y'.N', y''.N'') \rightsquigarrow D(N, x_1.D(N_1, y'.N', y''.N''), x_2.D(N_2, y'.N', y''.N'')) \quad (2.6)$$

$$ef(D(N, x_1.N_1, x_2.N_2)) \rightsquigarrow D(N, x_1.ef(N_1), x_2.ef(N_2)) \quad (2.7)$$

$$ef(N)N' \rightsquigarrow ef(N) \quad (2.8)$$

$$pr_i(ef(N)) \rightsquigarrow ef(N) \quad (2.9)$$

$$D(ef(N), x_1.N_1, x_2.N_2) \rightsquigarrow ef(N) \quad (2.10)$$

$$ef(ef(N)) \rightsquigarrow ef(N) \quad (2.11)$$

Other reductions might be considered; but, these suffice for our purposes. These include all those given in [15], with, in addition, those ((2.8), (2.9) and (2.10)) required by our avoidance of the restriction in [15] of $\perp E$ to an atomic conclusion C .

Various freshness conditions are required: in (2.4), neither x_i is free in N' ; in (2.6), neither x_i is free in N' or N'' .

Confluence and strong normalisation of this system is well-known; see [17] (where the restriction of $\perp E$ to atomic conclusions is inessential). A term-based strong normalisation proof can also be built on the basis of the techniques in [18, 9]; see for example [?].

2.3 Sequent Calculus

First, for clarity, we present a sequent calculus without proof terms. We choose, for reasons discussed elsewhere [24] by Vestergaard, a **G3i**-style calculus with principal formulae in the antecedent of the conclusion duplicated into the premisses. All two-premiss rules are context-sharing. Antecedents Γ are (temporarily) multisets of formulae:

$$\frac{}{\perp, \Gamma \Rightarrow C} L\perp \qquad \frac{}{A, \Gamma \Rightarrow A} Ax$$

$$\frac{A \supset B, \Gamma \Rightarrow A \quad B, A \supset B, \Gamma \Rightarrow C}{A \supset B, \Gamma \Rightarrow C} L\supset \qquad \frac{A, \Gamma \Rightarrow B}{\Gamma \Rightarrow A \supset B} R\supset$$

$$\frac{A_i, A_1 \wedge A_2, \Gamma \Rightarrow C}{A_1 \wedge A_2, \Gamma \Rightarrow C} L\wedge_i \qquad \frac{\Gamma \Rightarrow A \quad \Gamma \Rightarrow B}{\Gamma \Rightarrow A \wedge B} R\wedge$$

$$\frac{A_1, A_1 \vee A_2, \Gamma \Rightarrow C \quad A_2, A_1 \vee A_2, \Gamma \Rightarrow C}{A_1 \vee A_2, \Gamma \Rightarrow C} LV \qquad \frac{\Gamma \Rightarrow A_i}{\Gamma \Rightarrow A_1 \vee A_2} RV_i$$

$$\frac{\Gamma \Rightarrow A \quad A, \Gamma \Rightarrow C}{\Gamma \Rightarrow C} Cut$$

It is convenient to represent derivations by terms, in such a way that a (unique) derivation can be recovered from a term; these appear in the following typing rules. We use short names like X for constructors for typographic

reasons; note that A and C are used both as formula meta-variables and (with A for “apply” and C for “cut”) as such constructors. The variable Γ now ranges over “contexts”, i.e. assignments of formulae (aka “types”) to variables, as announced at the start of this Section. Terms are terms of a simple lambda calculus, with a variable binding mechanism, as in for example $C(L, x.L')$ where occurrences of x in L' are bound.

Now for the typing rules:

$$\begin{array}{c}
\frac{}{x : \perp, \Gamma \Rightarrow X(x) : C} L\perp \qquad \frac{}{x : A, \Gamma \Rightarrow x : A} Ax \\
\frac{z : A \supset B, \Gamma \Rightarrow L : A \quad y : B, z : A \supset B, \Gamma \Rightarrow L' : C}{z : A \supset B, \Gamma \Rightarrow A(z, L, y.L') : C} L\supset \qquad \frac{x : A, \Gamma \Rightarrow L : B}{\Gamma \Rightarrow \lambda x.L : A \supset B} R\supset \\
\frac{x : A_i, z : A_1 \wedge A_2, \Gamma \Rightarrow L : C}{z : A_1 \wedge A_2, \Gamma \Rightarrow E_i(z, x.L) : C} L\wedge_i \qquad \frac{\Gamma \Rightarrow L : A \quad \Gamma \Rightarrow L' : B}{\Gamma \Rightarrow (L, L') : A \wedge B} R\wedge \\
\frac{x : A_1, z : A_1 \vee A_2, \Gamma \Rightarrow L : C \quad x' : A_2, z : A_1 \vee A_2, \Gamma \Rightarrow L' : C}{z : A_1 \vee A_2, \Gamma \Rightarrow W(z, x.L, x'.L') : C} LV \qquad \frac{\Gamma \Rightarrow L : A_i}{\Gamma \Rightarrow in_i(L) : A_1 \vee A_2} RV_i \\
\frac{\Gamma \Rightarrow L : A \quad x : A, \Gamma \Rightarrow L' : C}{\Gamma \Rightarrow C(L, x.L') : C} Cut
\end{array}$$

Freshness constraints:

1. in $L\perp$, x is fresh for Γ ;
2. in Ax , x is fresh for Γ ;
3. in $L\supset$, z is fresh for Γ and y is fresh for $\Gamma, z : A \supset B$;
4. in $R\supset$, x is fresh for Γ ;
5. in $L\wedge_i$, z is fresh for Γ and x is fresh for $\Gamma, z : A_1 \wedge A_2$.
6. in LV , z is fresh for Γ , and both x and x' are fresh for $\Gamma, z : A_1 \vee A_2$;
7. in Cut , x is fresh for Γ .

Various easy consequences follow, e.g. that, in the rule $L\supset$, the variable y is not free in L .

3 Translation from Sequent Calculus into Natural Deduction

The following translation [8, 16, 25] is standard: details are only given for complete clarity.

$$\phi(X(x)) \equiv ef(x) \tag{3.1}$$

$$\phi(x) \equiv x \tag{3.2}$$

$$\phi(A(z, L, y.L')) \equiv [z\phi(L)/y] \phi(L') \tag{3.3}$$

$$\phi(\lambda x.L) \equiv \lambda x.\phi(L) \tag{3.4}$$

$$\phi(E_i(z, x.L)) \equiv [pr_i(z)/x] \phi(L) \tag{3.5}$$

$$\phi((L, L')) \equiv (\phi(L), \phi(L')) \tag{3.6}$$

$$\phi(W(z, x'.L', x''.L'')) \equiv D(z, x'.\phi(L'), x''.\phi(L'')) \tag{3.7}$$

$$\phi(in_i(L)) \equiv in_i(\phi(L)) \tag{3.8}$$

$$\phi(C(L, x.L')) \equiv [\phi(L)/x] \phi(L') \tag{3.9}$$

4 Translation from Natural Deduction to Sequent Calculus

The following translation, due to Gentzen [8], is only used here as evidence that the translation ϕ given above is surjective.

$$\psi(ef(x)) \equiv X(x) \quad (4.1)$$

$$\psi(x) \equiv x \quad (4.2)$$

$$\psi(ap(N, N')) \equiv C(\psi(N), x.A(x, \psi(N'), y.y)) \quad (4.3)$$

$$\psi(\lambda x.N) \equiv \lambda x.\psi(N) \quad (4.4)$$

$$\psi(pr_i(N)) \equiv C(\psi(N), x.E_i(x, y.y)) \quad (4.5)$$

$$\psi((N, N')) \equiv (\psi(N), \psi(N')) \quad (4.6)$$

$$\psi(D(N, x'.N', x''.N'')) \equiv C(\psi(N), z.W(z, x'.\psi(N'), x''.\psi(N'')) \quad (4.7)$$

$$\psi(in_i(N)) \equiv in_i(\psi(N)) \quad (4.8)$$

It is routine to observe that the composite of the two translations translates every natural deduction term to itself; thus ϕ is surjective (indeed, a retraction) and ψ is injective. In other words, for every N , $\phi(\psi(N)) = N$. The argument is, exploiting freshness of variables, by induction and case analysis:

1. $\phi(\psi(X(x))) \equiv \phi(ef(x)) \equiv X(x)$
2. $\phi(\psi(x)) \equiv \phi(x) \equiv x$
3. $\phi(\psi(ap(N, N')))$
 $\equiv \phi(C(\psi(N), x.A(x, \psi(N'), y.y)))$
 $\equiv [\phi(\psi(N))/x]\phi(A(x, \psi(N'), y.y))$
 $\equiv [N/x]\phi(A(x, \psi(N'), y.y))$
 $\equiv [N/x][ap(x, \phi(\psi(N')))/y]y$
 $\equiv [N/x]ap(x, \phi(\psi(N')))$
 $\equiv [N/x]ap(x, N')$
 $\equiv ap(N, N')$
4. $\phi(\psi(\lambda x.N)) \equiv \phi(\lambda x.\psi(N)) \equiv \lambda x.\phi(\psi(N)) \equiv \lambda x.N$
5. $\phi(\psi(pr_i(N)))$
 $\equiv \phi(C(\psi(N), x.E_i(x, y.y)))$
 $\equiv [\phi(\psi(N))/x]\phi(E_i(x, y.y))$
 $\equiv [\phi(\psi(N))/x][pr_i(x)/y]y$
 $\equiv [\phi(\psi(N))/x]pr_i(x)$
 $\equiv [N/x]pr_i(x)$
 $\equiv pr_i(N)$
6. $\phi(\psi((N, N'))) \equiv \phi((\psi(N), \psi(N'))) \equiv (\phi(\psi(N)), \phi(\psi(N'))) \equiv (N, N')$
7. $\phi(\psi(D(N, x'.N', x''.N'')))$
 $\equiv \phi(C(\psi(N), z.W(z, x'.\psi(N'), x''.\psi(N''))))$
 $\equiv [\phi(\psi(N))/z]\phi(W(z, x'.\psi(N'), x''.\psi(N'')))$
 $\equiv [N/z]\phi(W(z, x'.\psi(N'), x''.\psi(N'')))$
 $\equiv [N/z]D(z, x'.\phi(\psi(N')), x''.\phi(\psi(N'')))$
 $\equiv D(N, x'.N', x''.N'')$
8. $\phi(\psi(in_i(N))) \equiv \phi(in_i(\psi(N))) \equiv in_i(\phi(\psi(N))) \equiv N$

5 Substitution

We now present 32 lemmas about substitution; their only interest is that they are exactly what is required to show that cut-reduction steps translate to sequences of reduction steps. It is however of interest to see exactly what properties of substitution are required for the main result to hold.

Lemma 5.1. *Let N be a term and x and y be distinct variables. Then*

$$[N/x]y \equiv y.$$

Proof. By definition of substitution. □

Lemma 5.2. *Let N be a term and x be a variable. Then*

$$[N/x]x \equiv N.$$

Proof. By definition of substitution. □

Lemma 5.3. *Let N be terms and x and y be distinct variables, with y not free in N . Then*

$$[N/x](\lambda y.N') \equiv \lambda y.[N/x]N'.$$

Proof. By definition of substitution. □

Lemma 5.4. *Let N, N' and N'' be terms and x, y and z be distinct variables, with x not free in N and z not free in N or N' . Then*

$$[N/x][yN'/z]N'' \equiv [y[N/x]N'/z][N/x]N''.$$

Proof. By Lemma 2.1, since z is not free in N , the LHS $\equiv [[N/x](yN')/z][N/x]N''$ and, since $x \neq y$, $[N/x](yN') \equiv y[N/x]N'$. The result now follows. □

Lemma 5.5. *Let N' and N'' be terms and w, x and z be distinct variables, with z not free in N' . Then*

$$[w/x][xN'/z]N'' \equiv [w[w/x]N'/z][w/x]N''.$$

Proof. By Lemma 2.1, since $z \neq w$, the LHS $\equiv [[w/x](xN')/z][w/x]N''$ and, by definition of substitution, $[w/x](xN') \equiv w[w/x]N'$. The result now follows. (The freshness hypothesis about z is not used.) □

Lemma 5.6. *Let N, N^*, N' and N'' be terms and w, x, y and z be distinct variables, with x not free in N or N^* , y not free in N, N^* or N' and z not free in N, N' or N'' . Then*

$$[[wN/z]N^*/x][xN'/y]N'' \equiv [wN/z][N^*/x][xN'/y]N''.$$

Proof. Letting $M \equiv [xN'/y]N''$, observe that the RHS is just $[wN/z][N^*/x]M$; by Lemma 2.1, since x is not free in wN , this $\equiv [[wN/z]N^*/x][wN/z]M$. But, $z \neq x$ and z not free in N or N'' imply that $[wN/z]M \equiv M$. By symmetry of \equiv , the result now follows. (The freshness hypotheses about x and y and that about z w.r.t. N are not used.) □

Lemma 5.7. *Let N, N' and N'' be terms and w, x and y be distinct variables, with w not free in N' or N'' , x not free in N and y not free in N or N' . Then*

$$[\lambda w.N/x][xN'/y]N'' \rightsquigarrow^* [[\lambda w.N/x]N'/w][N/y][\lambda w.N/x]N''.$$

Proof. Let $M' \equiv [\lambda w.N/x]N'$ and $M'' \equiv [\lambda w.N/x]N''$. By definition of substitution, $M' \equiv (\lambda w.N)M'$. By Lemma 2.1, since y is not free in $\lambda w.N$, the LHS (of the present lemma) $\equiv [[\lambda w.N/x](xN')/y]M''$, which (by definition of substitution) $\equiv [(\lambda w.N)M'/y]M''$. This reduces (in 0 or more steps) by Rule 2.1 to $[[M'/w]N/y]M''$; there will be 0 steps if, for example, y is not free in N'' , but in general there may be several such steps. But, by Lemma 2.1, since y is not free in M' (because not free in N or N'), we have $[M'/w][N/y]M'' \equiv [[M'/w]N/y]M''$; by symmetry of \equiv the result follows. (The freshness hypotheses about w and x are not used.) \square

Lemma 5.8. *Let N, N' and N'' be terms and x be a variable, not free in N . Then*

$$[N/x](N', N'') \equiv ([N/x]N', [N/x]N'').$$

Proof. By definition of substitution. \square

Lemma 5.9. *Let N and N' be terms and x, y and z be distinct variables, with x not free in N and z not free in N . Then*

$$[N/x][pr_i(y)/z]N' \equiv [pr_i(y)/z][N/x]N'.$$

Proof. By Lemma 2.1, since z is not free in N and, from $x \neq y$, $[N/x]pr_i(y) \equiv pr_i(y)$. \square

Lemma 5.10. *Let N' be a term and w, x and z be distinct variables. Then*

$$[w/x][pr_i(x)/z]N' \equiv [pr_i(w)/z][w/x]N'.$$

Proof. By Lemma 2.1, since $z \neq w$ and $[w/x]pr_i(x) \equiv pr_i(w)$. \square

Lemma 5.11. *Let N' and N'' be terms and w, x, y and z be distinct variables, with x not free in N'' , y not free in N' and z not free in N'' . Then*

$$[[pr_i(w)/y]N''/x][pr_j(x)/z]N' \equiv [pr_i(w)/y][N''/x][pr_j(x)/z]N'.$$

Proof. By Lemma 2.1, since $x \neq w$, we obtain, with $N''' \equiv [pr_j(x)/z]N'$,

$$[pr_i(w)/y][N''/x]N''' \equiv [[pr_i(w)/y]N''/x][pr_i(w)/y]N'''.$$

But, $y \neq x$ and y is not free in N' , so, y is not free in N''' . By definition of substitution, we obtain

$$[[pr_i(w)/y]N''/x][pr_i(w)/y]N''' \equiv [[pr_i(w)/y]N''/x]N'''.$$

The result now follows by symmetry and transitivity of \equiv . \square

Lemma 5.12. *Let N, N' and N'' be terms and w, x, y and z be distinct variables, with x not free in N , y not free in N or N' and z not free in N' or N'' . Then*

$$[[pr_i(w)/z]N/x][xN'/y]N'' \equiv [pr_i(w)/z][N/x][xN'/y]N''.$$

Proof. By Lemma 2.1 and symmetry of \equiv , since w and x are distinct and since z is not free in $[xN'/y]N''$. \square

Lemma 5.13. *Let N, N' and N'' be terms and w, x, y and z be distinct variables, with x not free in N or N'' , y not free in N or N' and z not free in N or N'' . Then*

$$[[wN/y]N''/x] [pr_i(x)/z]N' \equiv [wN/y] [N''/x] [pr_i(x)/z]N'.$$

Proof. Let $M \equiv [pr_i(x)/z]N'$; by symmetry of \equiv , we have to show that

$$[wN/y] [N''/x]M \equiv [[wN/y]N''/x]M.$$

Since x is not free in wN and y is not free in M , this follows by Lemma 2.1. \square

Lemma 5.14. *Let N_1, N_2 and N' be terms and x and z be distinct variables, with x not free in N_1 or N_2 and z not free in N_1 or N_2 . Then*

$$[(N_1, N_2)/x][pr_i(x)/z]N' \rightsquigarrow^* [N_i/z][(N_1, N_2)/x]N'.$$

Proof. By Lemma 2.1, since z is not free in (N_1, N_2) , we obtain

$$[(N_1, N_2)/x][pr_i(x)/z]N' \equiv [pr_i((N_1, N_2)/x)/z][(N_1, N_2)/x]N'$$

which, by definition of substitution for x in x , is just

$$[(N_1, N_2)/x][pr_i(x)/z]N' \equiv [pr_i((N_1, N_2))/z][(N_1, N_2)/x]N'$$

But also

$$[pr_i((N_1, N_2))/z][(N_1, N_2)/x]N' \rightsquigarrow^* [N_i/z][(N_1, N_2)/x]N'$$

whence the result. \square

Lemma 5.15. *Let N and N' be terms and x be a variable, with x not free in N . Then*

$$[N/x][in_i(N') \equiv in_i([N/x]N').$$

Proof. By definition of substitution. \square

Lemma 5.16. *Let N, N' and N'' be terms and x, y, z' and z'' be distinct variables, with x not free in N , z' not free in N and z'' not free in N . Then*

$$[N/x]D(y, z'.N', z''.N'') \equiv D(y, z'.[N/x]N', z''.[N/x]N'').$$

Proof. By definition of substitution. \square

Lemma 5.17. *Let N' and N'' be terms and w, x, z' and z'' be distinct variables. Then*

$$[w/x]D(x, z'.N', z''.N'') \equiv D(w, z'.[w/x]N', z''.[w/x]N'').$$

Proof. By definition of substitution. \square

Lemma 5.18. *Let N_1, N_2, N' and N'' be terms and w, w_1, w_2, x, x' and x'' be distinct variables, with w_1 not free in N_2 or N' or N'' , w_2 not free in N_1 or N' or N'' , x not free in N_1 or N_2 , x' not free in N_1 or N_2 or N'' and x'' not free in N_1 or N_2 or N' . Then, with $M \equiv D(w, w_1.N_1, w_2.N_2)$,*

$$\begin{aligned} & [D(w, w_1.N_1, w_2.N_2)/x] D(x, x'.N', x''.N'') \\ & \quad \rightsquigarrow^* \\ & D(w, w_1.[N_1/x]D(x, x'.[M/x]N', x''.[M/x]N''), w_2.[N_2/x]D(x, x'.[M/x]N', x''.[M/x]N'')). \end{aligned}$$

Proof. Note that neither x nor x' nor x'' is free in M . By definition of substitution,

$$[M/x] D(x, x'.N', x''.N'') \equiv D(M, x'.[M/x]N', x''.[M/x]N'').$$

By permutation reduction rule 2.6, the RHS of this reduces to

$$D(w, w_1.D(N_1, x'.[M/x]N', x''.[M/x]N''), w_2.D(N_2, x'.[M/x]N', x''.[M/x]N'')).$$

But, since x is not free in M , this is \equiv to

$$D(w, w_1.[N_1/x]D(x, x'.[M/x]N', x''.[M/x]N''), w_2.[N_2/x]D(x, x'.[M/x]N', x''.[M/x]N'')).$$

as required. \square

Lemma 5.19. *Let N_1, N_2 and N be terms and x, y_1 and y_2 be distinct variables, with x not free in N , y_1 not free in N and y_2 not free in N . Then*

$$[in_i(N)/x]D(x, y_1.N_1, y_2.N_2) \rightsquigarrow^* [N/y_i][in_i(N)/x]N_i.$$

Proof. The LHS is \equiv to $D(in_i(N), y_1.[in_i(N)/x]N_1, y_2.[in_i(N)/x]N_2)$, by definition of substitution; this reduces by 2.3 to the RHS. \square

Lemma 5.20. *Let N_1, N_2, N' and N'' be terms and x, y, z, w_1 and w_2 be distinct variables, with x not free in N_1 or N_2 , y not free in N_1, N_2 or N' , w_1 not free in N' or N'' and w_2 not free in N' or N'' . Then (writing $M \equiv D(z, w_1.N_1, w_2.N_2)$ for brevity)*

$$\begin{aligned} & [M/x] [xN'/y]N'' \\ & \quad \rightsquigarrow^* \\ & [D(z, w_1.[N_1/x](x[M/x]N'), w_2.[N_2/x](x[M/x]N'))/y] [M/x] N'' . \end{aligned} \tag{5.1}$$

Proof. By Lemma 2.1, and since y is free in M ,

$$[M/x] [xN'/y]N'' \equiv [M[M/x]N'/y] [M/x] N'';$$

By reduction rule 2.4, $M[M/x]N' \rightsquigarrow^* D(z, w_1.N_1[M/x]N', w_2.N_2[M/x]N')$, whence

$$[M[M/x]N'/y] [M/x] N'' \rightsquigarrow^* [D(z, w_1.N_1[M/x]N', w_2.N_2[M/x]N')/y] [M/x] N''.$$

Since x is not free in M , and thus not free in $[M/x]N'$, this is \equiv to

$$[D(z, w_1.[N_1/x](x[M/x]N'), w_2.[N_2/x](x[M/x]N'))/y] [M/x] N'' ,$$

as required. \square

Lemma 5.21. *Let N_1, N_2, N and N'' be terms and x, w and z be distinct variables, distinct from w_1 and w_2 , with x not free in N or N'' and z not free in N, N_1 or N_2 . Then*

$$[[wN/z]N''/x]D(x, w_1.N_1, w_2.N_2) \equiv [wN/z][N''/x]D(x, w_1.N_1, w_2.N_2).$$

Proof. Essentially the same as 5.6. In other words, letting $M \equiv D(x, w_1.N_1, w_2.N_2)$, observe that the RHS is just $[wN/z][N''/x]M$; by Lemma 2.1, since x is not free in wN , this $\equiv [[wN/z]N''/x][wN/z]M$. But, $z \neq x$ and z not free in N_1 or N_2 imply that $[wN/z]M \equiv M$. By symmetry of \equiv , the result now follows. \square

Lemma 5.22. *Let N_1, N_2 and N be terms and x, y and z be distinct variables, distinct from w_1 and w_2 , with x not free in N_1 or N_2 and y not free in N_1 or N_2 and w_1 not free in N and w_2 not free in N . Then*

$$\begin{aligned} & [D(z, w_1.N_1, w_2.N_2)/x] [(pr_i(x)/y) N] \\ & \quad \rightsquigarrow^* \\ & [D(z, w_1.pr_i(N_1), w_2.pr_i(N_2))/y] [D(z, w_1.N_1, w_2.N_2)/x] N. \end{aligned} \quad (5.2)$$

Proof. By Lemma 2.1 and the conditions on y , and with $N' \equiv [D(z, w_1.N_1, w_2.N_2)/x]N$, we have

$$[D(z, w_1.N_1, w_2.N_2)/x] [(pr_i(x)/y) N] \equiv [pr_i(D(z, w_1.N_1, w_2.N_2))/y]N'.$$

By the reduction 2.5, this \equiv

$$[D(z, w_1.pr_i(N_1), w_2.pr_i(N_2))/y]N'$$

as required. \square

Lemma 5.23. *Let N, N_1, N_2 be terms and let x, y and z be distinct variables, distinct from w_1 and w_2 , with x not free in N and y not free in N_1 or N_2 . Then*

$$[[pr_i(z)/y]N/x] D(x, w_1.N_1, w_2.N_2) \equiv [pr_i(z)/y] [N/x] D(x, w_1.N_1, w_2.N_2).$$

Proof. Again, essentially the same as Lemma 5.6. Let $M \equiv D(x, w_1.N_1, w_2.N_2)$. By Lemma 2.1, the RHS \equiv

$$[[pr_i(z)/y]N/x][pr_i(z)/y]M.$$

But, since y is not free in M , this is \equiv the LHS. \square

Lemma 5.24. *Let N be a term and x and y be distinct variables, with x not free in N . Then*

$$[N/x](ef(y)) \equiv ef(y).$$

Proof. By definition of substitution. \square

Lemma 5.25. *Let w and x be distinct variables. Then*

$$[w/x](ef(x)) \equiv ef(w).$$

Proof. By definition of substitution. □

Lemma 5.26. *Let w and x be distinct variables. Then*

$$[ef(w)/x](ef(x)) \rightsquigarrow^* ef(w).$$

Proof. By definition of substitution and Rule 2.11. □

Lemma 5.27. *Let N and N' be terms and x, y and z be distinct variables, with y not free in N . Then*

$$[ef(z)/x][xN/y]N' \rightsquigarrow^* [ef(z)/y][ef(z)/x]N'.$$

Proof. By Lemma 2.1, the LHS $\equiv [ef(z)[ef(z)/x]N/y][ef(z)/x]N'$. It now suffices to show that

$$ef(z)[ef(z)/x]N \rightsquigarrow^* ef(z);$$

this follows by Rule 2.8. □

Lemma 5.28. *Let N and N' be terms and w, x and z be distinct variables, with x not free in N or N' and z not free in N . Then*

$$[[wN/z]N'/x] ef(x) \equiv [wN/z] [N'/x] ef(x).$$

Proof. Since x is not free in wN , by Lemma 2.1 the RHS is $[[wN/z]N'/x] [wN/z] ef(x)$. Since $z \neq x$, and so is not free in $ef(x)$, this is \equiv to the LHS. □

Lemma 5.29. *Let N be a term and x, y and z be distinct variables. Then*

$$[ef(z)/x] [pr_i(x)/y] N \rightsquigarrow^* [ef(z)/y] [ef(z)/x] N.$$

Proof. By Lemma 2.1, the LHS $\equiv [pr_i(ef(z))/y] [ef(z)/x] N$. By Rule 2.9, this reduces to

$$[ef(z)/y] [ef(z)/x] N.$$

□

Lemma 5.30. *Let N be a term and w, x and z be distinct variables, with x not free in N . Then*

$$[[pr_i(w)/z]N/x]ef(x) \equiv [pr_i(w)/z][N/x]ef(x)$$

Proof. Since $x \neq w$ and z is not free in $ef(x)$, by Lemma 2.1 the RHS simplifies to the LHS. □

Lemma 5.31. *Let N' and N'' be terms and x, z, y' and y'' be distinct variables. Then*

$$[ef(z)/x]D(x, y'.N', y''.N'') \rightsquigarrow^* ef(z)$$

Proof. The LHS is just $D(ef(z), y'.[ef(z)/x]N', y''.[ef(z)/x]N'')$; by Rule 2.10 this reduces to the RHS. □

Lemma 5.32. *Let N and N' be terms and x, w, y and y' be distinct variables. Then*

$$[D(w, y.N, y'.N')/x]ef(x) \rightsquigarrow^* D(w, y.[N/x]ef(x), y'.[N'/x]ef(x)).$$

Proof. The LHS is, by definition of substitution, $ef(D(w, y.N, y'.N'))$; this reduces by Rule 2.7 to $D(w, y.ef(N), y'.ef(N'))$, which, by definition of substitution, is just the RHS. \square

6 Cut reduction

Consider the following rules for reducing *Cut*:

$$C(L, x.y) \rightsquigarrow y \tag{6.1}$$

$$C(L, x.x) \rightsquigarrow L \tag{6.2}$$

$$C(L, x.\lambda y.L') \rightsquigarrow \lambda y.C(L, x.L') \tag{6.3}$$

$$C(L, x.A(y, L', z.L'')) \rightsquigarrow A(y, C(L, x.L'), z.C(L, x.L'')) \tag{6.4}$$

$$C(w, x.A(x, L', z.L'')) \rightsquigarrow A(w, C(w, x.L'), z.C(w, x.L'')) \tag{6.5}$$

$$C(A(w, L, z.L^*), x.A(x, L', y.L'')) \rightsquigarrow A(w, L, z.C(L^*, x.A(x, L', y.L''))) \tag{6.6}$$

$$C(\lambda w.L, x.A(x, L', y.L'')) \rightsquigarrow C(C(\lambda w.L, x.L'), w.C(L, y.C(\lambda w.L, x.L''))) \tag{6.7}$$

By Lemmata 5.1, 5.2, 5.3, 5.4, 5.5, 5.6 and 5.7 respectively, these rules are *semantically sound*, i.e. the translations of these rules into the language of natural deduction are either identities or (for the last rule) 0 or more reductions. The various hypotheses about distinctness and freeness in those lemmata come direct from the conditions implicit in the notation of these rules.

We now add further rules, for dealing with conjunction (and the interaction between conjunction and implication):

$$C(L, x.(L', L'')) \rightsquigarrow (C(L, x.L'), C(L, x.L'')) \tag{6.8}$$

$$C(L, x.E_i(y, z.L')) \rightsquigarrow E_i(y, z.C(L, x.L')) \tag{6.9}$$

$$C(w, x.E_i(x, z.L')) \rightsquigarrow E_i(w, z.C(w, x.L')) \tag{6.10}$$

$$C(E_i(w, y.L''), x.E_j(x, z.L')) \rightsquigarrow E_i(w, y.C(L'', x.E_j(x, z.L'))) \tag{6.11}$$

$$C(E_i(w, z.L), x.A(x, L', y.L'')) \rightsquigarrow E_i(w, z.C(L, x.A(x, L', y.L''))) \tag{6.12}$$

$$C(A(w, L, y.L''), x.E_i(x, z.L')) \rightsquigarrow A(w, L, y.C(L'', x.E_i(x, z.L'))) \tag{6.13}$$

$$C((L_1, L_2), x.E_i(x, z.L')) \rightsquigarrow C(L_i, z.C((L_1, L_2), x.L')) \tag{6.14}$$

Their semantic soundness follows from Lemmata 5.8, 5.9, 5.10, 5.11, 5.12, 5.13 and 5.14 respectively. Now we add the rules for disjunction (and its interactions with conjunction and implication):

$$C(L, x.in_i(L')) \rightsquigarrow in_i(C(L, x.L')) \quad (6.15)$$

$$C(L, x.W(y, z'.L', z''.L'')) \rightsquigarrow W(y, z'.C(L, x.L'), z''.C(L, x.L'')) \quad (6.16)$$

$$C(w, x.W(x, z'.L', z''.L'')) \rightsquigarrow W(w, z'.C(w, x.L'), z''.C(w, x.L'')) \quad (6.17)$$

$$\begin{aligned} C(W(w, w_1.L_1, w_2.L_2), x.W(x, x'.L', x''.L'')) \rightsquigarrow & W(w, w_1.C(L_1, x.W(x, x'.C(W(w, w_1.L_1, w_2.L_2), x.L'), \\ & x''.C(W(w, w_1.L_1, w_2.L_2), x.L''))), \\ & w_2.C(L_2, x.W(x, x'.C(W(w, w_1.L_1, w_2.L_2), x.L'), \\ & x''.C(W(w, w_1.L_1, w_2.L_2), x.L'')))) \end{aligned} \quad (6.18)$$

$$C(in_i(L), x.W(x, y_1.L_1, y_2.L_2)) \rightsquigarrow C(L, y_i.C(in_i(L), x.L_i)) \quad (6.19)$$

$$\begin{aligned} C(W(z, w_1.L_1, w_2.L_2), x.A(x, L', y.L'')) \rightsquigarrow & C(W(z, w_1.C(L_1, x.A(x, C(W(z, w_1.L_1, w_2.L_2), x.L'), y.y)), \\ & w_2.C(L_2, x.A(x, C(W(z, w_1.L_1, w_2.L_2), x.L'), y.y))), \\ & y.C(W(z, w_1.L_1, w_2.L_2), x.L'')) \end{aligned} \quad (6.20)$$

$$C(A(w, L, z.L''), x.W(x, w_1.L_1, w_2.L_2)) \rightsquigarrow A(w, L, z.C(L'', x.W(x, w_1.L_1, w_2.L_2))) \quad (6.21)$$

$$\begin{aligned} C(W(z, w_1.L_1, w_2.L_2), x.E_i(x, y.L)) \rightsquigarrow & C(W(z, w_1.C(L_1, x.E_i(x, y.y)), w_2.C(L_2, x.E_i(x, y.y))), \\ & y.C(W(z, w_1.L_1, w_2.L_2), x.L)) \end{aligned} \quad (6.22)$$

$$C(E_i(z, y.L), x.W(x, w_1.L_1, w_2.L_2)) \rightsquigarrow E_i(z, y.C(L, x.W(x, w_1.L_1, w_2.L_2))) \quad (6.23)$$

Their semantic soundness follows from Lemmata 5.15, 5.16, 5.17, 5.18, 5.19, 5.20, 5.21, 5.22 and 5.23 respectively.

Finally the rules for absurdity (and its interactions with the other logical constants):

$$C(L, x.X(y)) \rightsquigarrow X(y) \quad (6.24)$$

$$C(w, x.X(x)) \rightsquigarrow X(w) \quad (6.25)$$

$$C(X(w), x.X(x)) \rightsquigarrow X(w) \quad (6.26)$$

$$C(X(z), x.A(x, L, y.L')) \rightsquigarrow C(X(z), y.C(X(z), x.L')) \quad (6.27)$$

$$C(A(w, L, z.L'), x.X(x)) \rightsquigarrow A(w, L, z.C(L', x.X(x))) \quad (6.28)$$

$$C(X(z), x.E_i(x, y.L)) \rightsquigarrow C(X(z), y.C(X(z), x.L)) \quad (6.29)$$

$$C(E_i(w, z.L), x.X(x)) \rightsquigarrow E_i(w, z.C(L, x.X(x))) \quad (6.30)$$

$$C(X(z), x.W(x, y.L, y'.L')) \rightsquigarrow X(z) \quad (6.31)$$

$$C(W(w, y.L, y'.L'), x.X(x)) \rightsquigarrow W(w, y.C(L, x.X(x)), y'.C(L', x.X(x))) \quad (6.32)$$

Their semantic soundness follows from Lemmata 5.24, 5.25, 5.26, 5.27, 5.28, 5.29, 5.30, 5.31 and 5.32 respectively.

7 Completeness

Where the cut formula is non-principal in the second premiss, we use one of the reductions (6.1), (6.3), (6.4), (6.8), (6.9), (6.15), (6.16) and (6.24), according to the form of the last step of the second premiss.

Otherwise, we consider the cases where the cut formula is principal in the second premiss, presented in tabular form (where the first column indicates the last step of the first premiss and the top row indicates the last step of the second premiss). Note that some pairs, e.g. $R \supset / L \wedge$, can never arise.

	Ax	$L\supset$	$L\wedge$	$L\vee$	$L\perp$
Ax	6.2	6.5	6.10	6.17	6.25
$L\supset$	↓	6.6	6.13	6.21	6.28
$L\wedge$	↓	6.12	6.11	6.23	6.30
$L\vee$	↓	6.20	6.22	6.18	6.32
$L\perp$	↓	6.27	6.29	6.31	6.26
$R\supset$	↓	6.7	—	—	—
$R\wedge$	↓	—	6.14	—	—
$R\vee$	↓	—	—	6.19	—

It follows that every cut (except possibly where one of the premisses ends with a cut) can be matched to the LHS of one of the 32 reduction rules.

8 Counterexamples

Now that all the reduction rules have been presented, we are able to present some counterexamples (suggested by Graham-Lengrand). An alternative reduction for the antecedent of (6.18) is just to permute the cut into the first premiss without any adjustments. Thus, with $R \equiv W(x, x'.L', x''.L'')$,

$$C(W(w, w_1.L_1, w_2.L_2), x.R)$$

would be transformed to

$$W(w, w_1.C(L_1, x.R), w_2.C(L_2, x.R)).$$

Consider, with $w : A_1 \vee A_2$ and $L_i \equiv z_i : B_1 \vee B_2$, the term (of type $B_1 \vee B_2$)

$$L \equiv C(W(w, w_1.z_1, w_2.z_2), x.W(x, x_1.x, x_2.x)),$$

which reduces by this rule to

$$L' \equiv W(w, w_1.C(z_1, x.W(x, x_1.x, x_2.x)), w_2.C(z_2, x.W(x, x_1.x, x_2.x))).$$

The natural deduction image $\phi(L)$ of L is

$$[D(w, w_1.z_1, w_2.z_2)/x]D(x, x_1.x, x_2.x)$$

i.e.

$$N \equiv D(D(w, w_1.z_1, w_2.z_2), x_1.D(w, w_1.z_1, w_2.z_2), x_2.D(w, w_1.z_1, w_2.z_2))$$

The only reduction applicable to N is the permutative reduction (2.6), which reduces it to the normal term

$$N_1 \equiv D(w, w_1.D(z_1, x_1.S, x_2.S), w_2.D(z_2, x_1.S, x_2.S))$$

where

$$S \equiv D(w, w_1.z_1, w_2.z_2).$$

But, the natural deduction image $\phi(L')$ of L' is

$$D(w, w_1.[z_1/x]D(x, x_1.x, x_2.x)), w_2.[z_2/x]D(x, x_1.x, x_2.x))$$

i.e.

$$N' \equiv D(w, w_1.D(z_1, x_1.z_1, x_2.z_1), w_2.D(z_2, x_1.z_2, x_2.z_2)).$$

But N' is neither N nor N_1 ; so it is not the case that

$$N \rightsquigarrow^* N'.$$

It is also not the case that $N' \rightsquigarrow^* N_1$; since N_1 is normal (and the reduction system is confluent), we conclude that it is not even the case that $N = N'$ (in the equational theory generated by \rightsquigarrow^*).

Let us consider adding “immediate simplification” to the reduction system for NJ, i.e. adding the reduction of an $\vee E$ step, when a minor premiss does not use the extra assumption, to the derivation given by the minor premiss. Such immediate simplifications destroy confluence, so we prefer to avoid them; in their presence, however, it is now the case that $N_1 \rightsquigarrow^* N'$ (and hence $N \rightsquigarrow^* N'$), as follows:

$$N_1 \equiv D(w, w_1.D(z_1, x_1.S, x_2.S), w_2.D(z_2, x_1.S, x_2.S))$$

\rightsquigarrow^*

$$N' \equiv D(w, w_1.D(z_1, x_1.z_1, x_2.z_1), w_2.D(z_2, x_1.z_2, x_2.z_2))$$

by using immediate simplifications $S \rightsquigarrow z_i$ on the four different copies of S .

But we can block these immediate simplifications. A temporary abbreviation will be useful: we let xM abbreviate $A(x, M, u.u)$.

Recall that

$$L \equiv C(W(w, w_1.z_1, w_2.z_2), x.W(x, x_1.x, x_2.x)) : B_1 \vee B_2.$$

Each term $z_i : B_1 \vee B_2$ in L is replaced by $y_i w_i$ where $y_i : A_i \supset (B_1 \vee B_2)$; unlike z_i , this depends on w_i . The term $R \equiv W(x, x_1.x, x_2.x)$ in L is replaced by

$$T \equiv W(x, x_1.A(u_1, x_1, z.zx), x_2.A(u_2, x_2, z.zx)),$$

where $u_i : B_i \supset ((B_1 \vee B_2) \supset E)$. The fresh bound variable z herein is of type $(B_1 \vee B_2) \supset E$; so zx is of type E . Note that $A(u_i, x_i, z.zx)$ depends on x_i .

So the new term $L_0 : E$ whose cut, in the context

$$w : A_1 \vee A_2, y_1 : A_1 \supset (B_1 \vee B_2), y_2 : A_2 \supset (B_1 \vee B_2), u_1 : B_1 \supset ((B_1 \vee B_2) \supset E), u_2 : B_2 \supset ((B_1 \vee B_2) \supset E)$$

we reduce by our questionable reduction, is

$$L_0 \equiv C(W(w, w_1.y_1 w_1, w_2.y_2 w_2), x.T) : E$$

which reduces to

$$L'_0 \equiv W(w, w_1.C(y_1 w_1, x.T), w_2.C(y_2 w_2, x.T)) : E.$$

We now consider their natural deduction images $N_0 \equiv \phi(L_0)$ and $N'_0 \equiv \phi(L'_0)$. Carrying out the substitutions, we find that

$$N_0 \equiv [D(w, w_1.y_1 w_1, w_2.y_2 w_2)/x]D(x, x_1.u_1 x_1 x, x_2.u_2 x_2 x)$$

i.e.

$$N_0 \equiv D(D(w, w_1.y_1 w_1, w_2.y_2 w_2), x_1.u_1 x_1 D(w, w_1.y_1 w_1, w_2.y_2 w_2), x_2.u_2 x_2 D(w, w_1.y_1 w_1, w_2.y_2 w_2))$$

and similarly

$$N'_0 \equiv D(w, w_1.[y_1 w_1/x]D(x, x_1.u_1 x_1 x, x_2.u_2 x_2 x), w_2.[y_2 w_2/x]D(x, x_1.u_1 x_1 x, x_2.u_2 x_2 x))$$

i.e.

$$N'_0 \equiv D(w, w_1.D(y_1 w_1, x_1.u_1 x_1(y_1 w_1), x_2.u_2 x_2(y_1 w_1)), w_2.D(y_2 w_2, x_1.u_1 x_1(y_2 w_2), x_2.u_2 x_2(y_2 w_2))).$$

The immediate simplifications are now blocked: there is no sequence of reductions from N_0 to N'_0 . All we can do is apply the permutative conversion (2.6) and we obtain (with some renaming to avoid confusion of variables) the normal term

$$D(w, w_1.D(y_1 w_1, x_1.u_1 x_1 D(w, w'_1.y_1 w'_1, w'_2.y_2 w'_2), x_2.u_2 x_2 D(w, w'_1.y_1 w'_1, w'_2.y_2 w'_2)), w_2.\dots).$$

(But, since we no longer have confluence, the normality of this term is unhelpful.)

9 Strong Normalisation of Cut Reduction

Weak normalisation (in the typed system) is shown by observing that the rank of the cut is always reduced, where the *rank* of a cut is the triple comprising the cut formula, the height of the first premiss and the height of the second premiss, with ranks lexicographically ordered and formulae ordered by the “is a subformula of” relation.

Strong normalisation (in the typed system) is shown by a lexicographic path order argument (as in [1]). First one removes the variable binders, so we are dealing with a system of first-order terms (i.e. terms with no binders). (An infinite reduction sequence with binders would translate to an infinite reduction sequence without binders.) We then order the constructors as follows: every cut constructor C exceeds every other constructor (such as W), and cut constructors C are ordered according to the size of the cut formula (which is not made explicit in the term; this could be done, but less readably). The details are tedious, but routine, since one merely has to check, for each rule, that $LHS > RHS$; they have been checked using a Prolog program [7] that implements the LPO method.

The rules (for LPO ordering $>$ of terms s, t, \dots) are as follows (where \triangleright indicates the relation between a term and each of its immediate subterms, \triangleleft is the converse, and $>^{lex}$ is the lexicographic extension of $>$ to tuples, with associated rule $>^{lex}$):

$$\frac{\exists u \triangleleft s. u \geq t}{s > t} >_i \quad \frac{s \gg t \quad \forall u \triangleleft t. s > u}{s > t} >_{ii}$$

$$\frac{f > g}{f(s_1, \dots, s_m) \gg g(t_1, \dots, t_n)} \gg_i \frac{(s_1, \dots, s_n) >^{lex} (t_1, \dots, t_n)}{f(s_1, \dots, s_n) \gg f(t_1, \dots, t_n)} \gg_{ii}$$

An illustrative example (6.8) with the constructor p for ‘pair’, is thus:

$$\frac{\frac{C > p}{C(L, p(L', L'')) \gg p(C(L, L'), C(L, L''))} \gg_i \frac{\frac{\frac{L' \geq L'}{p(L', L'') > L'} >_i \quad \frac{(L, p(L', L'')) >^{lex} (L, L')}{C(L, p(L', L'')) \gg C(L, L')} >^{lex} \quad \dots \quad \dots}{C(L, p(L', L'')) > C(L, L')} \gg_{ii} \quad \dots \quad \dots}{C(L, p(L', L'')) > p(C(L, L'), C(L, L''))} >_{ii}$$

and neither space nor their intrinsic interest permits the inclusion here of the 31 other cases. (The missing bits indicated here by \dots are similar, concluding that $C(L, p(L', L'')) > L$, that $C(L, p(L', L'')) > L'$ and that $C(L, p(L', L'')) > C(L, L'')$ respectively.)

10 Subject Reduction

As always, one has to show that if a term L has (in the context Γ) type A , and $L \rightsquigarrow L'$, then also L' has type A . We could consider the 32 separate cases; but we only show two in detail.

One of these is routine. Using D for $A \supset B$ for brevity and omitting the context Γ , Rule 6.7 transforms

$$\frac{\frac{\dots}{w : A \Rightarrow L : B} \Rightarrow \lambda w.L : D}{R \supset} \quad \frac{\frac{\dots}{x : D \Rightarrow L' : A} \quad \frac{\dots}{x : D, y : B \Rightarrow L'' : E}}{x : A \supset B \Rightarrow A(x, L', y.L'') : E} L \supset}{\Rightarrow C(\lambda w.L, x.A(x, L', y.L'')) : E} \text{Cut}$$

into (using Wk for some weakening steps)

$$\frac{\frac{\dots}{\Rightarrow \lambda w.L : D} \quad \frac{\dots}{x : D \Rightarrow L' : A}}{\Rightarrow C(\lambda w.L, x.L') : A} \text{Cut} \quad \frac{\frac{\dots}{w : A \Rightarrow L : B} \quad \frac{\frac{\dots}{y : B, w : A \Rightarrow \lambda w.L : D} Wk \quad \frac{\dots}{x : D, y : B, w : A \Rightarrow L'' : E} Wk}}{y : B, w : A \Rightarrow C(\lambda w.L, x.L'') : E} \text{Cut}}{w : A \Rightarrow C(L, y.C(\lambda w.L, x.L'')) : E} \text{Cut}}{\Rightarrow C(C(\lambda w.L, x.L'), w.C(L, y.C(\lambda w.L, x.L'')) : E)} \text{Cut}$$

The other example, however, is more complicated. We recall Rule 6.18:

$$C(W(w, w_1.L_1, w_2.L_2), x.W(x, x'.L', x''.L'')) \rightsquigarrow W(w, w_1.C(L_1, x.W(x, x'.C(W(w, w_1.L_1, w_2.L_2), x.L'), x''.C(W(w, w_1.L_1, w_2.L_2), x.L''))), w_2.C(L_2, x.W(x, x'.C(W(w, w_1.L_1, w_2.L_2), x.L'), x''.C(W(w, w_1.L_1, w_2.L_2), x.L'')))))$$

We need some abbreviations. Let

$$\begin{aligned} M &\equiv W(w, w_1.L_1, w_2.L_2), \\ M^* &\equiv W(x, x'.C(M, x.L'), x''.C(M, x.L'')), \\ M_i &\equiv C(L_i, x.M^*). \end{aligned}$$

So we can simplify the rule to

$$C(W(w, w_1.L_1, w_2.L_2), x.W(x, x'.L', x''.L'')) \rightsquigarrow W(w, w_1.C(L_1, x.M^*), w_2.C(L_2, x.M^*)).$$

With $A \equiv A_1 \vee A_2$ and $B \equiv B' \vee B''$, the *LHS* of this type-checks, we suppose, as follows, in the presence of some context Γ left implicit:

$$\frac{\frac{\dots}{w : A, w_1 : A_1 \Rightarrow L_1 : B} \dots}{w : A \Rightarrow M : B} L \vee \quad \frac{\frac{\dots}{x : B, w : A, x' : B' \Rightarrow L' : E} \dots}{x : B, w : A \Rightarrow W(x, x'.L', x''.L'') : E} L \vee}{w : A \Rightarrow C(M, x.W(x, x'.L', x''.L'')) : E} \text{Cut}$$

It is transformed into a derivation too large to be conveniently displayed. We begin by considering the derivation of the first upper right premiss: $x : B, w : A, x' : B' \Rightarrow L' : E$; we rename, for some fresh variable x^* , the variable x to x^* , obtaining a derivation of $x^* : B, w : A, x' : B' \Rightarrow L^* : E$; this can now be weakened with $x : B$ to a derivation of $x : B, x^* : B, w : A, x' : B' \Rightarrow L^* : E$. The left premiss $w : A \Rightarrow M : B$ can, after a weakening, be cut with this, giving

$$\frac{\frac{\dots}{x : B, w : A, x' : B' \Rightarrow M : B} \quad \frac{\dots}{x : B, x^* : B, w : A, x' : B' \Rightarrow L^* : E}}{x : B, w : A, x' : B' \Rightarrow C(M, x^*.L^*) : E} \text{Cut}$$

and the same is done to obtain a derivation of $x : B, w : A, x'' : B'' \Rightarrow C(M, x^*.L''^*) : E$. Since $B \equiv B' \vee B''$, we use $L\vee$ to obtain a derivation of

$$x : B, w : A \Rightarrow W(x, x'.C(M, x^*.L^*), x''.C(M, x^*.L''^*)) : E,$$

i.e. of

$$x : B, w : A \Rightarrow M^* : E.$$

We now (for each $i = 1, 2$) perform, after some weakenings, another cut:

$$\frac{\frac{\dots}{x : B, w : A, w_i : A_i \Rightarrow L_i : B} \quad \frac{\dots}{x : B, w : A, w_i : A_i \Rightarrow M^* : E}}{x : B, w : A, w_i : A_i \Rightarrow C(L_i, x.M^*) : E} \text{Cut}$$

and conclude as follows:

$$\frac{\frac{\dots}{w : A, w_1 : A_1 \Rightarrow C(L_1, x.M^*) : E} \quad \frac{\dots}{w : A, w_2 : A_2 \Rightarrow C(L_2, x.M^*) : E}}{w : A \Rightarrow W(w, w_1.C(L_1, x.M^*), w_2.C(L_2, x.M^*)) : E} L\vee.$$

All the renamings are, by use of alpha-conversion, omitted in the earlier presentation (rule 6.18). Note that we have used the height-preserving admissible rule of *Weakening*.

11 Confluence of Cut Reduction

The system of cut-reduction rules (6.1–6.32) is a left-linear orthogonal pattern-rewrite system, without critical pairs; by the results of [12], confluence is immediate. That there are no critical pairs is simply the observation that every term matches at most one LHS from the set of rules (6.1–6.32), and any LHS from one of these rules fails to match any non-variable proper subterm of any of these rules.

12 Conclusion

Putting the various results together, we have the following:

Theorem 1 *The system of cut reduction rules (6.1–6.32) is complete (for reducing cuts), confluent, strongly normalising (on typed terms) and satisfies the subject reduction property; moreover, every cut reduction rule translates via ϕ to a sequence of zero or more reductions in the natural deduction setting.*

13 Acknowledgments

Thanks are due to Jan von Plato, Peter Chapman, Jacob Howe, Stéphane Graham-Lengrand and Christian Urban for helpful comments and (to the last of these) for a copy of [21] prior to its publication, albeit many years after its 2001 presentation in Rio. Chapman’s work [5] (incorporating also ideas by Urban) was invaluable in checking the correctness of all the lemmata about substitution. The work was motivated by requirements for some not yet published work (joint with James Caldwell) supported by EPSRC grant EP/F031114/1.

References

1. Franz Baader and Tobias Nipkow. *Term Rewriting and All That*, Cambridge University Press, 1998.
2. Henk Barendregt. *The Lambda Calculus*, revised ed, Elsevier Science, 1984.
3. Mirjana Borisavljević. *Extended natural-deduction images of conversions from the system of sequents*, J Logic Computation **14**, pp 769–799, 2004.
4. Roy Dyckhoff and Christian Urban. *Strong normalization of Herbelin’s explicit substitution calculus with substitution propagation*, J Logic Computation **13**, pp 689–706, 2003.
5. Roy Dyckhoff and Peter Chapman. Isabelle proof script, available from its first author, 2009.
6. Roy Dyckhoff. *Cut-elimination, substitution and normalisation*, Section A3. Logic and Computation, 14th Congress of Logic, Methodology and Philosophy of Science, Nancy, 2011.
7. Roy Dyckhoff. LPO checker in Prolog, available from the author, 2013.
8. Gerhard Gentzen. *Untersuchungen uber das logische Schließen*, Mathematische Zeitschrift **39**, pp 176–210 and 405–431, 1935.
9. Felix Joachimski and Ralph Matthes. *Short proofs of normalization for the simply-typed lambda-calculus, permutative conversions and Gödel’s T*, Archive for Mathematical Logic **42**, pp 59–87, 2003.
10. Kentaro Kikuchi. *On a local-step cut-elimination procedure for the intuitionistic sequent calculus*, Proc. of the 13th Int. Conf. on Logic for Programming Artificial Intelligence and Reasoning (LPAR’06), **4246** of LNCS, pp 120–134, Springer-Verlag, 2006.
11. Georg Kreisel. *A survey of proof theory II*, in J.E.Fenstad (ed.), *Proc. 2nd Scandinavian Logic Symposium*, North-Holland, pp 109–170, 1971.
12. Richard Mayr and Tobias Nipkow. *Higher-order rewrite systems and their confluence*, Theoretical Computer Science **192**, pp 3–29, 1998.
13. Sara Negri and Jan von Plato. *Structural Proof Theory*, Cambridge University Press, 2001.
14. Jan von Plato. *Natural deduction with general elimination rules*, Arch. Math. Logic **40**, pp 541–567, 2001.
15. Garrel Pottinger. *Normalisation as a homomorphic image of cut-elimination*, Annals of Mathematical Logic **12**, pp 323–357, North-Holland, 1988.
16. Dag Prawitz. *Natural Deduction: A Proof-Theoretical Study*, Almqvist & Wiksell, Stockholm, 1965.
17. Dag Prawitz. *Ideas and results in proof theory*, in J.E.Fenstad (ed.), *Proc. 2nd Scandinavian Logic Symposium*, North-Holland, pp 235–307, 1971.
18. Femke von Raamsdonk and Paula Severi. *On normalisation*, Centrum voor Wiskunde en Informatica, Amsterdam, CS-R9545, 1995.
19. Helmut Schwichtenberg and Stanley S. Wainer. *Proofs and Computations*, Perspectives in Logic series, Association for Symbolic Logic, Cambridge University Press, 2012.
20. Anne Troelstra and Dirk van Dalen. *Constructivism in Mathematics: An Introduction, Vol II*, Studies in Logic and the Foundations of Mathematics **123**, North-Holland, 2000.
21. Anne Troelstra and Helmut Schwichtenberg. *Basic Proof Theory*, 2nd ed, Cambridge, 2000.
22. Christian Urban. *Revisiting Zucker’s work on the correspondence between cut-elimination and normalisation*, in L.C. Pereira, E.H. Haeusler and V. de Paiva (eds.), *Advances in Natural Deduction. A Celebration of Dag Prawitz’s Work*, Springer, Dordrecht, 21 pp, 2014.
23. Christian Urban and Gavin Bierman. *Strong normalisation of cut-elimination in classical logic*, Fundamenta Informaticae **45**, pp 123–155, 2001.
24. Christian Urban. *Nominal techniques in Isabelle/HOL*, J Automatic Reasoning **40**, pp 327–356, 2008.
25. René Vestergaard. *Revisiting Kreisel: A computational anomaly in the Troelstra-Schwichtenberg G3i system*, unpublished, March 1999.
26. Jeff Zucker. *The correspondence between cut-elimination and normalisation*, Annals of Mathematical Logic **7**, pp 1–112, 1974. 2000.