Strong Normalisation of Cut-Elimination in Classical Logic

C. Urban G.M. Bierman

University of Cambridge Computer Laboratory
{cu200,gmb}@cl.cam.ac.uk

Abstract. In this paper a strongly normalising cut-elimination procedure is presented for classical logic. The procedure adapts the standard cut transformations, see for example [12]. In particular our cut-elimination procedure requires no special annotations on formulae. We design a term calculus for a variant of Kleene's sequent calculus G3 via the Curry-Howard correspondence and the cut-elimination steps are given as rewrite rules. In the strong normalisation proof we adapt the symmetric reducibility candidates developed by Barbanera and Berardi.

1 Introduction

Gentzen has shown in his seminal paper [10] that all cuts can be eliminated from proofs in LK and LJ. Since then many *Hauptsätze* (cut-elimination theorems) have appeared for various sequent calculus formulations. Most of them, including Gentzen's original, provide a cut-elimination procedure which is weakly normalising, i.e., they employ a particular reduction strategy (for example an inner-most reduction strategy or the elimination of the cut with the highest rank). Besides these weakly normalising methods a few strongly normalising cut-elimination procedures have been developed; for example in [4–7, 13, 14]. However, all those methods impose some form of restriction on the reduction rules to ensure strong normalisation. A common restriction is to not allow a cut-rule to pass over another cut-rule (exceptions are [6, 13]). However this limits, in the intuitionistic case, the correspondence between cut-elimination and beta-reduction [8, 14]. Therefore in this paper we develop a strongly normalising cut-elimination procedure adapting the standard cut-elimination steps for *logical cuts* and allowing *commuting cuts* to pass over other cuts. (A cut-rule is said to be a logical cut when both cut-formulae are introduced by axioms or logical inference rules; otherwise the cut is said to be a commuting cut.) Our method is closely related to the cut-elimination procedure developed for LK^{tq} [6, 15]. However we do not need their colour annotations.

The problem of non-termination of cut-elimination occurs in both intuitionistic logic and classical logic. One example of a non-terminating reduction sequence in intuitionistic logic is given in [20]; for classical logic [6] and [9] give the following example:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A\vdash A}\quad\dfrac{}{A\vdash A}}{A\vee A\vdash A,A}\,\vee_L}{A\vee A\vdash A}\,Contr_R \qquad \dfrac{\dfrac{\dfrac{}{A\vdash A}\quad\dfrac{}{A\vdash A}}{A,A\vdash A\wedge A}\,\wedge_R}{A\vdash A\wedge A}\,Contr_L}{A\vee A\vdash A\wedge A}\,Cut$$

where a commuting cut needs to be eliminated. There are two possible reductions: either the cut can be permuted upwards in the left proof branch or in the right proof branch. If one is not careful, applying these reductions in alternation can lead to arbitrary big normal forms and to non-termination. This is remedied in [6] by devising a specific protocol for cut-elimination, which depends on additional information ('colours') attached to every cut-formula. For this cut-elimination procedure strong normalisation and confluence has been proved; the colours are used to ingeniously map every LK^{tq}-proof to a corresponding proof-net in linear logic and every cut-elimination step to a series of reductions on proof-nets (strong normalisation for proof-nets has been proved in [11]).

We shall consider a sequent calculus formulation very similar to Kleene's G3 [16] and G3c of [18], where the structural rules are completely implicit in the form of the logical rules. Another feature of our work is that we shall annotate proofs with terms and term rewrite rules will describe the cut-elimination steps. In our approach no additional information is required to guide the cut-elimination process. The rest of the paper is organised as follows: §2 contains various notational conventions and definitions; §3 contains a detailed proof of strong normalisation for the rewrite system. The proof adapts the technique of symmetric reducibility candidates [1]; §4 concludes and gives suggestions for further work.

2 Terms, Judgements, Rewrite Rules and Substitution

The main idea behind the cut-elimination procedure presented in this paper is to transport one subderivation of a commuting cut to the place(s) where the cut-formula is introduced. Consider the following proof in G3c:

$$\dfrac{\pi_1\left\{\dfrac{\dfrac{}{A\vdash B\supset C,A^{\bullet}}\quad\dfrac{\dfrac{}{A,B\vdash C,A^{\bullet}}}{A\vdash B\supset C,A}\,\supset_R}{A\vee A\vdash B\supset C,A}\,\vee_L\right. \quad \left.\dfrac{\dfrac{\dfrac{}{A^{\star}\vdash D,A}\quad\dfrac{}{A^{\star}\vdash D,A}}{A\vdash D,A\wedge A}\,\wedge_R\quad\dfrac{\dfrac{}{A^{\star},E\vdash A}\quad\dfrac{}{A^{\star},E\vdash A}}{A,E\vdash A\wedge A}\,\wedge_R}{A,D\supset E\vdash A\wedge A}\,\supset_L\right\}\pi_2}{A\vee A,D\supset E\vdash B\supset C,A\wedge A}\,Cut$$

The cut-formula A is neither a main formula in the inference rule $\vee_L$, nor in $\supset_L$. Therefore the cut is a commuting cut. In π_1 the cut-formula is a main formula in the axioms marked with a bullet; in π_2, respectively, in the axioms marked with a star. Eliminating the cut in the proof above means to either transport the derivation π_2 to the places marked with a bullet and 'cut it against' the corresponding axioms, or to transport π_1 and 'cut it against' the axioms marked with a star. In both cases the derivation being transported is duplicated.

In the remainder of this section we shall annotate proofs, via the Curry-Howard correspondence, with terms and present a rewrite system for cut-elimination. The raw terms are defined in Figure 1 using *names* and *co-names* as

Raw Terms: $M, N ::=$	$\mathsf{Ax}(x, a)$	Axiom	
$\mid$	$\mathsf{Cut}(\langle a{:}B\rangle M, (x{:}B)N)$	Cut	
$\mid$	$\mathsf{And}_R(\langle a{:}B\rangle M, \langle b{:}C\rangle N, c)$	And-R	
$\mid$	$\mathsf{And}_L^i((x{:}B)M, y)$	And-L$_i$	$(i = 1, 2)$
$\mid$	$\mathsf{Or}_R^i(\langle a{:}B\rangle M, b)$	Or-R$_i$	$(i = 1, 2)$
$\mid$	$\mathsf{Or}_L((x{:}B)M, (y{:}C)N, z)$	Or-L	
$\mid$	$\mathsf{Imp}_R((x{:}B)\langle a{:}C\rangle M, b)$	Imp-R	
$\mid$	$\mathsf{Imp}_L(\langle a{:}B\rangle M, (x{:}C)N, y)$	Imp-L	

Fig. 1. The grammar for the raw terms where B and C are types; x, y, z are taken from a set of *names* and a, b, c from a set of *co-names*.

binders. Besides the terms, which are going to be used as annotations for proofs, there are two other syntactic categories which play an important rôle in the definition of substitution and in the strong normalisation proof. Let M and N be terms, then $(x{:}B)M$ and $\langle a{:}B\rangle N$ are called *named terms* and *co-named terms*, respectively. We use round brackets to signify that a name becomes bound in a term and angle brackets that a co-name becomes bound in a term. Analogous to the Church-style formation rules for the λ-calculus, all binders are explicitly typed (types are defined as normal). However in what follows we will omit these typings when they are clear from the context. Given a term M, its set of free names is written as $FN(M)$ and its set of free co-names is written as $FC(M)$ (similarly for named and co-named terms) – their routine definitions are omitted. We assume that the three types of terms are equal up to α-conversion and that a Barendregt-style naming convention holds for names and co-names (see 2.1.13 in [2]). Rewriting a name x to y in M is written as $M\{x \mapsto y\}$ (respectively $M\{a \mapsto b\}$ for co-names). The routine formalisation of the rewriting operation is omitted.

In the following we are only concerned with terms which can be well-typed by the inference system given in Figure 2. The typing judgements are of the form $\Gamma \triangleright M \triangleright \Delta$ where Γ is a set of name-type pairs and Δ is a set of co-name-type pairs. The reader will see that this system is the term system for a variant of Kleene's G3 formulation via the Curry-Howard correspondence. Our $\wedge_L$ and $\vee_R$ rules differ slightly from the G3 and G3c of [18]: they provide more convenience in the strong normalisation proof, but the original rules could be used as well (see Section 4). There are no primitive rules for contraction and weakening: they are completely implicit in the form of the logical rules. However, special care needs to be taken with implicit contractions. Consider the proof fragment:

$$\frac{x{:}B, \Gamma \triangleright M \triangleright \Delta, b{:}B \supset C, a{:}C}{\Gamma \triangleright \mathsf{Imp}_R((x)\langle a\rangle M, b) \triangleright \Delta, b{:}B \supset C}\ \supset_R \qquad (1)$$

The typing rule introduces the co-name-type pair $b : B \supset C$ in the conclusion. However it is allowed that this pair can already be present in the premise. On the other hand, the name-type pair $x : B$ and the co-name-type pair $a : C$ in the

$$x\!:\!B,\Gamma \triangleright \mathsf{Ax}(x,a) \triangleright \Delta, a\!:\!B$$

$$\frac{x\!:\!B_i,\Gamma \triangleright M \triangleright \Delta}{y\!:\!B_1\wedge B_2,\Gamma \triangleright \mathsf{And}^i_L(\langle x\rangle M,y) \triangleright \Delta}\ \wedge_{L_i} \qquad \frac{\Gamma \triangleright M \triangleright \Delta, a\!:\!B \quad \Gamma \triangleright N \triangleright \Delta, b\!:\!C}{\Gamma \triangleright \mathsf{And}_R(\langle a\rangle M,\langle b\rangle N,c) \triangleright \Delta, c\!:\!B\wedge C}\ \wedge_R$$

$$\frac{x\!:\!B,\Gamma \triangleright M \triangleright \Delta \quad y\!:\!C,\Gamma \triangleright N \triangleright \Delta}{z\!:\!B\vee C,\Gamma \triangleright \mathsf{Or}_L(\langle x\rangle M,\langle y\rangle N,z) \triangleright \Delta}\ \vee_L \qquad \frac{\Gamma \triangleright M \triangleright \Delta, a\!:\!B_i}{\Gamma \triangleright \mathsf{Or}^i_R(\langle a\rangle M,b) \triangleright \Delta, b\!:\!B_1\vee B_2}\ \vee_{R_i}$$

$$\frac{\Gamma \triangleright M \triangleright \Delta, a\!:\!B \quad x\!:\!C,\Gamma \triangleright N \triangleright \Delta}{y\!:\!B\supset C,\Gamma \triangleright \mathsf{Imp}_L(\langle a\rangle M,\langle x\rangle N,y) \triangleright \Delta}\ \supset_L \qquad \frac{x\!:\!B,\Gamma \triangleright M \triangleright \Delta, a\!:\!C}{\Gamma \triangleright \mathsf{Imp}_R(\langle x\rangle\langle a\rangle M,b) \triangleright \Delta, b\!:\!B\supset C}\ \supset_R$$

$$\frac{\Gamma_1 \triangleright M \triangleright \Delta_1, a\!:\!B \quad x\!:\!B,\Gamma_2 \triangleright N \triangleright \Delta_2}{\Gamma_1,\Gamma_2 \triangleright \mathsf{Cut}(\langle a\rangle M,\langle x\rangle N) \triangleright \Delta_1,\Delta_2}\ Cut$$

Fig. 2. The typing rules for the propositional fragment.

premise are not allowed to be in the conclusion: they become bound in the term. The following definition corresponds to the traditional notion of what the main formula of a inference rule is.

Definition 1.
A term M *introduces* the name z or co-name c if M is of the form:

for z: $\mathsf{Ax}(z,c)$	for c: $\mathsf{Ax}(z,c)$
$\mathsf{And}^i_L(\langle x\rangle S,z)$	$\mathsf{And}_R(\langle a\rangle S,\langle b\rangle T,c)$
$\mathsf{Or}_L(\langle x\rangle S,\langle y\rangle T,z)$	$\mathsf{Or}^i_R(\langle a\rangle S,c)$
$\mathsf{Imp}_L(\langle a\rangle S,\langle x\rangle T,z)$	$\mathsf{Imp}_R(\langle x\rangle\langle a\rangle S,c)$

Recall our example from the beginning of this section where a commuting cut can be permuted in two different directions. Therefore the rewrite system for our cut-elimination procedure is defined using two, symmetric forms of substitution, which are written as $P[x := \langle a\rangle Q]$ and $S[b := \langle y\rangle T]$. These substitutions are used when the inference rules directly above the cut do not introduce the cut-formula. In these cases the cuts can permute, or 'jump' directly to the place(s) where the cut-formula is introduced (i.e., is a main formula). Whenever a substitution 'hits' a term where the cut-formula is introduced the substitution 'expands' to a cut. Two examples are as follows:

$$\mathsf{And}_R(\langle a\rangle M,\langle b\rangle N,c)[c := \langle x\rangle P] \stackrel{\text{def}}{=} \mathsf{Cut}(\langle c\rangle \mathsf{And}_R(\langle a\rangle M,\langle b\rangle N,c),\langle x\rangle P)$$
$$\mathsf{Ax}(x,a)[x := \langle b\rangle Q] \stackrel{\text{def}}{=} \mathsf{Cut}(\langle b\rangle Q,\langle x\rangle \mathsf{Ax}(x,a))$$

In the first term the formula labelled with c is the main formula and in the second the formula labelled with x is a main formula. So in both cases the substitution expands to a cut. In the other cases where the name or co-name that is substituted is not a label for the main formula, then the substitution is pushed into the subterms or vanishes in case of the axioms. Two examples are as follows (assume the substitution $[\sigma]$ is *not* of the form $[z := \ldots]$ or $[a := \ldots]$):

$$\mathsf{Or}_L((x)M,(y)N,z)[\sigma] \stackrel{\text{def}}{=} \mathsf{Or}_L((x)\ M[\sigma],(y)\ N[\sigma],z)$$
$$\mathsf{Ax}(z,a)[\sigma] \stackrel{\text{def}}{=} \mathsf{Ax}(z,a)$$

However, special care needs to be taken for axioms, because they have two main formulae. For technical reasons in the strong normalisation proof we need the following property:

$$M[x := \langle a\rangle P][b := (y)Q] \equiv M[b := (y)Q][x := \langle a\rangle P] \quad (2)$$

if $b \notin FC(\langle a\rangle P)$ and $x \notin FN((y)Q)$. The naïve definition outlined above does not satisfy this property: in case M is of the form $\mathsf{Ax}(x,b)$ we get two different terms:

$$\mathsf{Ax}(x,b)[x := \langle a\rangle P][b := (y)Q] \stackrel{\text{def}}{=} \mathsf{Cut}(\langle a\rangle P,(x)\mathsf{Cut}(\langle b\rangle \mathsf{Ax}(x,b),(y)Q))$$
$$\mathsf{Ax}(x,b)[b := (y)Q][x := \langle a\rangle P] \stackrel{\text{def}}{=} \mathsf{Cut}(\langle b\rangle \mathsf{Cut}(\langle a\rangle P,(x)\mathsf{Ax}(x,b)),(y)Q)$$

Furthermore the nested cuts with an axiom as an immediate subterm could be a source for non-termination as noted in [6]. Therefore we use a more subtle definition of substitution and introduce two special clauses to handle the problematic example above.

Definition 2. Substitution

$$\begin{aligned}
\mathsf{Cut}(\langle a\rangle \mathsf{Ax}(x,a),(y)M)[x := \langle b\rangle P] &\stackrel{\text{def}}{=} \mathsf{Cut}(\langle b\rangle P,(x)M\{y\mapsto x\})\\
\mathsf{Cut}(\langle a\rangle M,(x)\mathsf{Ax}(x,b))[b := (y)P] &\stackrel{\text{def}}{=} \mathsf{Cut}(\langle b\rangle M\{a\mapsto b\},(y)P)\\
M[c := (y)P] &\stackrel{\text{def}}{=} \mathsf{Cut}(\langle c\rangle M,(y)P) \qquad \text{if } M \text{ introduces } c\\
M[y := \langle c\rangle P] &\stackrel{\text{def}}{=} \mathsf{Cut}(\langle c\rangle P,(y)M) \qquad \text{if } M \text{ introduces } y
\end{aligned}$$

otherwise

$$\begin{aligned}
\mathsf{Ax}(x,a)[\sigma] &\stackrel{\text{def}}{=} \mathsf{Ax}(x,a)\\
\mathsf{Cut}(\langle a\rangle M,(x)N)[\sigma] &\stackrel{\text{def}}{=} \mathsf{Cut}(\langle a\rangle\ M[\sigma],(x)\ N[\sigma])\\
\mathsf{And}_R(\langle a\rangle M,\langle b\rangle N,c)[\sigma] &\stackrel{\text{def}}{=} \mathsf{And}_R(\langle a\rangle\ M[\sigma],\langle b\rangle\ N[\sigma],c)\\
\mathsf{And}^i_L((x)M,y)[\sigma] &\stackrel{\text{def}}{=} \mathsf{And}^i_L((x)\ M[\sigma],y)\\
\mathsf{Or}^i_R(\langle a\rangle M,b)[\sigma] &\stackrel{\text{def}}{=} \mathsf{Or}^i_R(\langle a\rangle\ M[\sigma],b)\\
\mathsf{Or}_L((x)M,(y)N,z)[\sigma] &\stackrel{\text{def}}{=} \mathsf{Or}_L((x)\ M[\sigma],(y)\ N[\sigma],z)\\
\mathsf{Imp}_R((x)\langle a\rangle M,b)[\sigma] &\stackrel{\text{def}}{=} \mathsf{Imp}_R((x)\langle a\rangle\ M[\sigma],b)\\
\mathsf{Imp}_L(\langle a\rangle M,(x)N,y)[\sigma] &\stackrel{\text{def}}{=} \mathsf{Imp}_L(\langle a\rangle\ M[\sigma],(x)\ N[\sigma],y)
\end{aligned}$$

Recall that we assumed a Barendregt-style naming condition for (co-)names. A substitution $M[a := (x{:}B)N]$ is said to be *well-formed*, iff $\mathsf{Cut}(\langle a{:}B\rangle M,(x{:}B)N)$ is well-typed. In the following we shall consider only well-formed substitutions.

A naïve translation of the traditional, logical cut-elimination rules into our term calculus is, for example, as follows ($\wedge_1$ case):

$$\mathsf{Cut}(\langle c\rangle \mathsf{And}_R(\langle a\rangle M,\langle b\rangle N,c),(y)\mathsf{And}^1_L((x)P,y)) \longrightarrow \mathsf{Cut}(\langle a\rangle M,(x)P)$$

However, there is a problem with this reduction rule. In our sequent calculus, the structural rules are implicit (see the discussion of proof (1)). This makes the calculus smaller, and more importantly it provides a very convenient way to define substitution (no explicit contractions are required when a term is duplicated). Unfortunately, we have to pay a price for this in the logical cut-elimination rules. Consider the following instance of the redex above:

$$\dfrac{\dfrac{\Gamma_1 \triangleright M \triangleright \Delta_1, c\!:\!B\wedge C, a\!:\!B \quad \Gamma_1 \triangleright N \triangleright \Delta_1, b\!:\!C}{\Gamma_1 \triangleright \mathsf{And}_R(\langle a\rangle M, \langle b\rangle N, c) \triangleright \Delta_1, c\!:\!B\wedge C}\wedge_R \quad \dfrac{x\!:\!B, \Gamma_2 \triangleright P \triangleright \Delta_2}{y\!:\!B\wedge C, \Gamma_2 \triangleright \mathsf{And}^1_L(\langle x\rangle P, y) \triangleright \Delta_2}\wedge_{L_1}}{\Gamma_1, \Gamma_2 \triangleright \mathsf{Cut}(\langle c\rangle \mathsf{And}_R(\langle a\rangle M, \langle b\rangle N, c), (y)\mathsf{And}^1_L(\langle x\rangle P, y)) \triangleright \Delta_1, \Delta_2}\,Cut$$

where $c\!:\!B\wedge C \in FC(M)$. The naïve reduction rule given above would (incorrectly!) reduce this proof to the following:

$$\dfrac{\Gamma_1 \triangleright M \triangleright \Delta_1, c\!:\!B\wedge C, a\!:\!B \quad x\!:\!B, \Gamma_2 \triangleright P \triangleright \Delta_2}{\Gamma_1, \Gamma_2 \triangleright \mathsf{Cut}(\langle a\rangle M, (x)P) \triangleright \Delta_1, \Delta_2, c\!:\!B\wedge C}\,Cut$$

Unfortunately c has now become free! In order to obtain a subject reduction property for the rewrite system we have to include in every logical reduction step extra substitutions (the main formula of the conclusion could potentially be in every subterm). These substitutions ensure that no bound (co-)name becomes free. In effect the logical reduction rules look slightly complicated, but that is the price we have to pay for the convenience of not having explicit structural rules. The cut-elimination procedure is defined (in its entirety) as follows:

Definition 3. Cut-Elimination

Logical Cuts ($i = 1, 2$)

1. $\mathsf{Cut}(\langle b\rangle \mathsf{And}_R(\langle a_1\rangle M_1, \langle a_2\rangle M_2, b), (y)\mathsf{And}^i_L((x)N, y))$
   $\longrightarrow \mathsf{Cut}(\langle a_i\rangle M_i[b := (y)\mathsf{And}^i_L((x)N, y)], (x)N[y := \langle b\rangle \mathsf{And}_R(\langle a_1\rangle M_1, \langle a_2\rangle M_2, b)])$
2. $\mathsf{Cut}(\langle b\rangle \mathsf{Or}^i_R(\langle a\rangle M, b), (y)\mathsf{Or}_L((x_1)N_1, (x_2)N_2, y))$
   $\longrightarrow \mathsf{Cut}(\langle a\rangle M[b := (y)\mathsf{Or}_L((x_1)N_1, (x_2)N_2, y)], (x_i)N_i[y := \langle b\rangle \mathsf{Or}^i_R(\langle a\rangle M, b)])$
3. $\mathsf{Cut}(\langle b\rangle \mathsf{Imp}_R((x)\langle a\rangle M, b), (z)\mathsf{Imp}_L(\langle c\rangle N, (y)P, z))$
   $\longrightarrow \mathsf{Cut}(\langle a\rangle \mathsf{Cut}(\langle c\rangle N[z := \langle b\rangle S], (x)M[b := (z)T]), (y)P[z := \langle b\rangle S])$ or
   $\longrightarrow \mathsf{Cut}(\langle c\rangle N[z := \langle b\rangle S], (x)\mathsf{Cut}(\langle a\rangle M[b := (z)T], (y)P[z := \langle b\rangle S]))$
   where $S \equiv \mathsf{Imp}_R((x)\langle a\rangle M, b)$ and $T \equiv \mathsf{Imp}_L(\langle c\rangle N, (y)P, z)$
4. $\mathsf{Cut}(\langle a\rangle M, (x)\mathsf{Ax}(x, b)) \longrightarrow M\{a \mapsto b\}$ if M introduces a
5. $\mathsf{Cut}(\langle a\rangle \mathsf{Ax}(y, a), (x)M) \longrightarrow M\{x \mapsto y\}$ if M introduces x

Commuting Steps (otherwise)

6. $\mathsf{Cut}(\langle a\rangle M, (x)N) \longrightarrow M[a := (x)N]$ if M does not introduce a or
   $\longrightarrow N[x := \langle a\rangle M]$ if N does not introduce x

There are a few subtleties in the reduction rule for the third case. Firstly, there are two ways to reduce a cut-rule having an implication as the cut-formula. Therefore we have included two reductions for this case. Secondly, special care needs to be taken that there is no clash between bound and free (co-)names. In the first reduction rule we need to ensure that a is not a free co-name in N; in the second rule that x is not free in P. This can always be achieved by renaming a and x appropriately (they are binders in $\mathsf{Imp}_R((x)\langle a\rangle M, b)$). We assume that the renaming is done implicitly in the cut-elimination procedure.

The main difference between our rules and the cut-elimination procedure defined for LK^{tq} is the inclusion of non-determinism. Recall our example from the beginning of this section where a commuting cut can move in two directions. Let $\mathsf{Cut}(\langle a\rangle M, (x)N)$ be the term annotation for this commuting cut where M and N are the corresponding term annotations for proofs π_1 and π_2, respectively. According to our last rule, this term can reduce to either $M[a := (x)N]$ or $N[x := \langle a\rangle M]$. The choice to which term it reduces is not specified (similarly for the reduction of the logical cut in the third case). In contrast, in LK^{tq} this choice is completely determined by the colour annotation. In general the colour annotation reduces the number of normal forms (cut-free proofs) reachable from a proof containing cuts (see §4 for an example). For the substitution we have the following lemmas:

Lemma 1.
(i) $M[x := \langle a\rangle \mathsf{Ax}(y,a)] \longrightarrow^{+} M\{x \mapsto y\}$ *or* $M[x := \langle a\rangle \mathsf{Ax}(y,a)] \equiv M$
(ii) $M[a := (x)\mathsf{Ax}(x,b)] \longrightarrow^{+} M\{a \mapsto b\}$ *or* $M[a := (x)\mathsf{Ax}(x,b)] \equiv M$

Proof. Routine induction on the structure of M.

Lemma 2. *For any arbitrary substitution* $[\sigma]$
if $M \longrightarrow M'$*, then* $M[\sigma] \longrightarrow M'[\sigma]$ *or* $M[\sigma] \equiv M'[\sigma]$

Proof. Induction on the structure of M. One interesting case is where $M[\sigma] \equiv M'[\sigma]$; it is as follows:

Case $M \equiv \mathsf{Cut}(\langle a\rangle \mathsf{Ax}(y,a), (x)P)$**:** Let P introduce x, then $M \longrightarrow M'$ with $M' \equiv P\{x \mapsto y\}$. Let $[\sigma]$ be $[y := \langle c\rangle Q]$. We have:

$$M[\sigma] \equiv \mathsf{Cut}(\langle a\rangle \mathsf{Ax}(y,a), (x)P)[y := \langle c\rangle Q] \stackrel{\text{def}}{=} \mathsf{Cut}(\langle c\rangle Q, (y)P\{x \mapsto y\})$$
$$M'[\sigma] \equiv P\{x \mapsto y\}[y := \langle c\rangle Q] \stackrel{\text{def}}{=} \mathsf{Cut}(\langle c\rangle Q, (y)P\{x \mapsto y\})$$

3 Proof of Strong Normalisation

We give in this section a detailed proof of strong normalisation for the reduction system developed in the previous section. To save space only details for the $\wedge$-fragment are presented, but some pointers are given at the end of this section for the other connectives. The proof uses the notion of symmetric reducibility candidates from [1]. The proof proceeds as follows:

1. Define the sets of candidates over types using a fixed point construction.
2. Prove that candidates are closed under reduction.
3. Show that a named or co-named term in a candidate implies strong normalisation for the corresponding term.
4. Prove that all terms are strongly normalising.

The set SN denotes the set of strongly normalising terms. The candidates are defined only for named and co-named terms. We say that $\langle B\rangle$ is the *type* of co-named terms of the form $\langle a{:}B\rangle M$; similarly (B) is the type of named terms of the form $(x{:}B)M$. We define:

1. $CT_{\langle B\rangle}$ is the set of co-named terms of type $\langle B\rangle$,
2. $NT_{(B)}$ is the set of named terms of type (B).

In the following we define for every type $\langle B\rangle$ and (B) the candidates, written as $[\![\langle B\rangle]\!]$ and $[\![(B)]\!]$; they are subsets of $CT_{\langle B\rangle}$ and $NT_{(B)}$, respectively. The definition of the candidates uses set operators for which we define the types as follows (where the set of all subsets of a given set S will be denoted as $\mathcal{P}(S)$):

$$\begin{array}{l}
\textsc{andright}_{\langle B\wedge C\rangle} : \mathcal{P}(CT_{\langle B\rangle}) \times \mathcal{P}(CT_{\langle C\rangle}) \times \mathcal{P}(NT_{(B\wedge C)}) \to \mathcal{P}(CT_{\langle B\wedge C\rangle})\\
\textsc{andleft}^{i}_{(B_1\wedge B_2)} : \mathcal{P}(NT_{(B_i)}) \times \mathcal{P}(CT_{\langle B_1\wedge B_2\rangle}) \to \mathcal{P}(NT_{(B_1\wedge B_2)})\\
\textsc{binding}_{(B)} : \mathcal{P}(CT_{\langle B\rangle}) \to \mathcal{P}(NT_{(B)})\\
\textsc{binding}_{\langle B\rangle} : \mathcal{P}(NT_{(B)}) \to \mathcal{P}(CT_{\langle B\rangle})\\
\textsc{neg}_{(B)} : \mathcal{P}(CT_{\langle B\rangle}) \to \mathcal{P}(NT_{(B)})\\
\textsc{neg}_{(B)} : \mathcal{P}(CT_{\langle B\rangle}) \to \mathcal{P}(NT_{(B)})
\end{array}$$

The operators are indexed on types. When defining the set operators we use the following two sets of named and co-named axioms:

$$\begin{array}{l}
\textsc{axioms}_{(B)} \stackrel{\text{def}}{=} \{(x{:}B)\mathsf{Ax}(y,b) \mid \text{for all } \mathsf{Ax}(y,b)\} \subseteq NT_{(B)}\\
\textsc{axioms}_{\langle B\rangle} \stackrel{\text{def}}{=} \{\langle a{:}B\rangle\mathsf{Ax}(y,b) \mid \text{for all } \mathsf{Ax}(y,b)\} \subseteq CT_{\langle B\rangle}
\end{array}$$

The set operators $\textsc{andright}$, $\textsc{andleft}^{i}$ and $\textsc{binding}$ are defined as follows:

$$\begin{array}{rl}
\textsc{andright}_{\langle B\wedge C\rangle}(X,Y,Z) \stackrel{\text{def}}{=} & \{\langle c{:}B\wedge C\rangle\mathsf{And}_R(\langle a{:}B\rangle M, \langle b{:}C\rangle N, c) \mid\\
& \forall\, (x{:}B\wedge C)P \in Z.\ \langle a\rangle\ M[c := (x)P] \in X \text{ and } \langle b\rangle\ N[c := (x)P] \in Y\}\\
\textsc{andleft}^{i}_{(B_1\wedge B_2)}(X,Y) \stackrel{\text{def}}{=} & \{(y{:}B_1\wedge B_2)\mathsf{And}^{i}_{L}((x{:}B_i)M, y) \mid\\
& \forall\, \langle a{:}B_1\wedge B_2\rangle P \in Y.\ (x)\ M[y := \langle a\rangle P] \in X\}\\
\textsc{binding}_{(B)}(X) \stackrel{\text{def}}{=} & \{(x{:}B)M \mid \forall \langle a{:}B\rangle P \in X.\ M[x := \langle a{:}B\rangle P] \in SN\}\\
\textsc{binding}_{\langle B\rangle}(Y) \stackrel{\text{def}}{=} & \{\langle a{:}B\rangle M \mid \forall (x{:}B)P \in Y.\ M[a := (x{:}B)P] \in SN\}
\end{array}$$

The set operator $\textsc{neg}$ and the candidates $[\![(B)]\!]$ and $[\![\langle B\rangle]\!]$ are defined simultaneously over types:

$$\begin{array}{rll}
\textsc{neg}_{\langle B\rangle}(X) \stackrel{\text{def}}{=} & \textsc{axioms}_{\langle B\rangle} \cup\ \textsc{binding}_{\langle B\rangle}(X) & \langle B\rangle \text{ atomic}\\
\stackrel{\text{def}}{=} & \textsc{axioms}_{\langle C\wedge D\rangle}\ \cup\ \textsc{binding}_{\langle C\wedge D\rangle}(X) \cup & \langle B\rangle \equiv \langle C\wedge D\rangle\\
& \textsc{andright}_{\langle C\wedge D\rangle}([\![\langle C\rangle]\!], [\![\langle D\rangle]\!], X) &\\
\textsc{neg}_{(B)}(Y) \stackrel{\text{def}}{=} & \textsc{axioms}_{(B)} \cup \textsc{binding}_{(B)}(Y) & (B) \text{ atomic}\\
\stackrel{\text{def}}{=} & \textsc{axioms}_{(C\wedge D)}\ \cup\ \textsc{binding}_{(C\wedge D)}(Y) \cup & (B) \equiv (C\wedge D)\\
& \textsc{andleft}^{1}_{(C\wedge D)}([\![(C)]\!], Y)\ \cup \textsc{andleft}^{2}_{(C\wedge D)}([\![(D)]\!], Y) &
\end{array}$$

For the definition of the candidates we use fixed points of an increasing set operator. A set operator *op* is said to be:

$$\begin{array}{ll}
\text{increasing, iff} & S \subseteq S' \ \Rightarrow\ op(S) \subseteq op(S'), \text{ and}\\
\text{decreasing, iff} & S \subseteq S' \ \Rightarrow\ op(S) \supseteq op(S').
\end{array}$$

The candidates are defined as follows:

$$[\![(B)]\!] \stackrel{\text{def}}{=} X_0 \text{ and } [\![\langle B\rangle]\!] \stackrel{\text{def}}{=} \textsc{neg}_{\langle B\rangle}([\![(B)]\!])$$

where X_0 is the least fixed point of the operator $\textsc{neg}_{(B)}\circ\textsc{neg}_{\langle B\rangle}$.[1] We have that $\textsc{binding}_{\langle B\rangle}$ and $\textsc{andright}_{\langle C\wedge D\rangle}$ (i.e., $X \mapsto \textsc{andright}_{\langle C\wedge D\rangle}([\![\langle C\rangle]\!], [\![\langle D\rangle]\!], X)$) are decreasing operators. But then $\textsc{neg}_{\langle B\rangle}$ must be a decreasing operator (similarly $\textsc{neg}_{(B)}$ must be decreasing). If both $\textsc{neg}_{\langle B\rangle}$ and $\textsc{neg}_{(B)}$ are decreasing, then the operator $\textsc{neg}_{(B)}\circ\textsc{neg}_{\langle B\rangle}$ is increasing and the least fixed point X_0 exists according to Tarski's fixed point theorem. For the candidates we have:

$$[\![(B)]\!] = \textsc{neg}_{(B)}([\![\langle B\rangle]\!]) \text{ and } [\![\langle B\rangle]\!] = \textsc{neg}_{\langle B\rangle}([\![(B)]\!]).$$

Since $\textsc{neg}$ is closed under $\textsc{axioms}$ we also have have:

$$\textsc{axioms}_{(B)} \subseteq [\![(B)]\!] \text{ and } \textsc{axioms}_{\langle B\rangle} \subseteq [\![\langle B\rangle]\!]. \tag{3}$$

Lemma 3.

(i) If $\langle a{:}B\rangle M \in [\![\langle B\rangle]\!]$ *and* $M\longrightarrow M'$ *then* $\langle a{:}B\rangle M' \in [\![\langle B\rangle]\!]$.
(ii) If $(x{:}B)M \in [\![(B)]\!]$ *and* $M\longrightarrow M'$ *then* $(x{:}B)M' \in [\![(B)]\!]$.

Proof. We prove both cases simultaneously by induction on $\langle B\rangle$ and (B).

Case $\langle B\rangle$ atomic: For (i) we have $[\![\langle B\rangle]\!] = \textsc{neg}_{\langle B\rangle}([\![(B)]\!])$; therefore $\langle a{:}B\rangle M \in \textsc{axioms}_{\langle B\rangle} \cup \textsc{binding}_{\langle B\rangle}([\![(B)]\!])$. M cannot be an axiom (because axioms do not reduce), therefore $\langle a{:}B\rangle M \in \textsc{binding}_{\langle B\rangle}([\![(B)]\!]) \stackrel{\text{def}}{=} \{\langle a{:}B\rangle S \mid \forall (x{:}B)T \in [\![(B)]\!].S[a := (x{:}B)T] \in SN\}$. For $\langle a{:}B\rangle M$ we have $M[a := (x{:}B)P] \in SN$ for all $(x{:}B)P \in [\![(B)]\!]$ and since $M\longrightarrow M'$ we know by Lemma 2 that either $M[a := (x)P]\longrightarrow M'[a := (x)P]$ or $M[a := (x)P] \equiv M'[a := (x)P]$. In both cases we have $M'[a := (x{:}B)P] \in SN$ for all $(x{:}B)P \in [\![(B)]\!]$. This implies that $\langle a{:}B\rangle M' \in \textsc{binding}_{\langle B\rangle}([\![(B)]\!])$ and hence $\langle a{:}B\rangle M' \in \textsc{neg}_{\langle B\rangle}([\![(B)]\!])$. Therefore $\langle a{:}B\rangle M' \in [\![\langle B\rangle]\!]$. Similarly for (ii).

Case $\langle B\rangle \equiv \langle C\wedge D\rangle$: $\langle a{:}C\wedge D\rangle M$ is element of $[\![\langle C\wedge D\rangle]\!] = \textsc{neg}_{\langle C\wedge D\rangle}([\![(C\wedge D)]\!]) \stackrel{\text{def}}{=}$

$$\textsc{axioms}_{\langle C\wedge D\rangle} \cup \textsc{binding}_{\langle C\wedge D\rangle}([\![(C\wedge D)]\!]) \cup \textsc{andright}_{\langle C\wedge D\rangle}([\![\langle C\rangle]\!], [\![\langle D\rangle]\!], [\![(D\wedge C)]\!]).$$

$\langle a{:}C\wedge D\rangle M \notin \textsc{axioms}_{\langle C\wedge D\rangle}$, because axioms do not reduce. Therefore we have that $\langle a{:}C\wedge D\rangle M \in \textsc{andright}_{\langle C\wedge D\rangle}([\![\langle C\rangle]\!], [\![\langle D\rangle]\!], [\![(C\wedge D)]\!])$ or that $\langle a{:}C\wedge D\rangle M \in \textsc{binding}_{\langle C\wedge D\rangle}([\![(C\wedge D)]\!])$. In the second case we reason as in the atomic case. In the first case we know that $\langle a\rangle M$ is of the form $\langle c{:}C\wedge D\rangle \mathsf{And}_R(\langle d\rangle S, \langle e\rangle T, c)$ and $\langle a\rangle M' \equiv \langle c{:}C\wedge D\rangle \mathsf{And}_R(\langle d\rangle S', \langle e\rangle T', c)$ where either $S\longrightarrow S'$ and $T \equiv T'$ or $S \equiv S'$ and $T\longrightarrow T'$. Assume the former case (the other case being similar). We have that $\langle d{:}C\rangle S[c := (x)P] \in [\![\langle C\rangle]\!]$ for all $(x{:}C\wedge D)P \in [\![(C\wedge D)]\!]$. Since $S\longrightarrow S'$ we know by Lemma 2 that either $S[c := (x)P] \equiv S'[c := (x)P]$ or $S[c := (x)P]\longrightarrow S'[c := (x)P]$. In both

[1] In all rigour we also have to assume that the candidates are closed under α-conversion.

cases (in the second by IH) we can infer that $\langle d\rangle S'[c := (x)P] \in [\![(C)]\!]$ for all $(x{:}C\wedge D)P \in [\![(C\wedge D)]\!]$. Therefore we know that $\langle a{:}C\wedge D\rangle M'$ must be in $\text{ANDRIGHT}_{\langle C\wedge D\rangle}([\![\langle C\rangle]\!], [\![\langle D\rangle]\!], [\![(C\wedge D)]\!])$ and we can conclude that $\langle a{:}C\wedge D\rangle M' \in [\![\langle C\wedge D\rangle]\!]$. Similarly for (ii).

Lemma 4.

(i) If $\langle a{:}B\rangle M \in [\![\langle B\rangle]\!]$, then $M \in SN$.
(ii) If $(x{:}B)M \in [\![(B)]\!]$, then $M \in SN$.

Proof. Simultaneous induction on the types $\langle B\rangle$ and (B).

Case $\langle B\rangle$ atomic: Since $[\![\langle B\rangle]\!] = \text{NEG}_{\langle B\rangle}([\![(B)]\!])$ we have $\langle a{:}B\rangle M \in \text{AXIOMS}_{\langle B\rangle}$ or $\langle a{:}B\rangle M \in \text{BINDING}_{\langle B\rangle}([\![(B)]\!])$. In the first case M is an axiom and therefore strongly normalising. In the second case we know that $M[a := (x{:}B)P] \in SN$ for all $(x{:}B)P \in [\![(B)]\!]$. By (3) we have $(x{:}B)\mathsf{Ax}(x, a) \in [\![(B)]\!]$ and therefore $M[a := (x)\mathsf{Ax}(x, a)] \in SN$. Furthermore we know by Lemma 2 that either $M[a := (x)\mathsf{Ax}(x, a)] \equiv M$ or $M[a := (x)\mathsf{Ax}(x, a)] \longrightarrow^{+} M$. Therefore $M \in SN$. Similarly for (ii).

Case $\langle B\rangle \equiv \langle C\wedge D\rangle$: By $[\![\langle C\wedge D\rangle]\!] = \text{NEG}_{\langle C\wedge D\rangle}([\![(C\wedge D)]\!])$ we have that:

$$\langle a{:}C\wedge D\rangle M \in \text{AXIOMS}_{\langle C\wedge D\rangle} \cup \text{BINDING}_{\langle C\wedge D\rangle}([\![(C\wedge D)]\!]) \cup \text{ANDRIGHT}_{\langle C\wedge D\rangle}([\![\langle C\rangle]\!], [\![\langle D\rangle]\!], [\![(C\wedge D)]\!])$$

If $\langle a{:}C\wedge D\rangle M$ is element of the first two sets we reason as in the atomic case. Left to show is that $M \in SN$ if $\langle a\rangle M \in \text{ANDRIGHT}_{\langle C\wedge D\rangle}([\![\langle C\rangle]\!], [\![\langle D\rangle]\!], [\![(C\wedge D)]\!])$. In this case $\langle a\rangle M$ is of the form $\langle c\rangle \mathsf{And}_R(\langle d\rangle S, \langle e\rangle T, c)$ where $\langle d\rangle S[c := (x)P] \in [\![\langle C\rangle]\!]$ and $\langle e\rangle T[c := (x)P] \in [\![\langle D\rangle]\!]$ for all $(x : C\wedge D)P \in [\![(C\wedge D)]\!]$. By (3) we know that $(x{:}C\wedge D)\mathsf{Ax}(x, c) \in [\![(C\wedge D)]\!]$ and we have $\langle d\rangle S[c := (x)\mathsf{Ax}(x, c)] \in [\![\langle C\rangle]\!]$ and $\langle e\rangle T[c := (x)\mathsf{Ax}(x, c)] \in [\![\langle D\rangle]\!]$. By IH we can infer that $S[c := (x)\mathsf{Ax}(x, c)] \in SN$ and $T[c := (x)\mathsf{Ax}(x, c)] \in SN$. From Lemma 1 we can infer that $S[c := (x)\mathsf{Ax}(x, c)] \equiv S$ or $S[c := (x)\mathsf{Ax}(x, c)] \longrightarrow^{+} S$. In both cases we know that $S \in SN$ (similarly $T \in SN$). But then $\mathsf{And}_R(\langle d\rangle S, \langle e\rangle T, c)$ must be strongly normalising too. Similarly for (ii).

Lemma 5. *If $M, N \in SN$ and $\langle a{:}B\rangle M \in [\![\langle B\rangle]\!]$, $(x{:}B)N \in [\![(B)]\!]$ then $\mathsf{Cut}(\langle a{:}B\rangle M, (x{:}B)N) \in SN$.*

Proof. We assign to each term of the form $\mathsf{Cut}(\langle a{:}B\rangle M, (x{:}B)N)$ a lexicographically ordered induction value of the form $(\delta, l(M), l(N))$ where δ is the degree of the cut-formula B; $l(M)$ and $l(N)$ are the lengths of the maximal reduction sequences starting from M and N, respectively. By assumption both $l(M)$ and $l(N)$ are finite. We prove that all terms to which $\mathsf{Cut}(\langle a\rangle M, (x)N)$ reduces are strongly normalising.

Inner Reduction: $\mathsf{Cut}(\langle a\rangle M, (x)N) \longrightarrow \mathsf{Cut}(\langle a\rangle M', (x)N')$ where either $M \equiv M'$ and $N \longrightarrow N'$ or $M \longrightarrow M'$ and $N \equiv N'$. Assume the later case (the other case being similar). We have to prove that $\mathsf{Cut}(\langle a\rangle M', (x)N) \in SN$.

From $\langle a{:}B\rangle M \in [\![\langle B\rangle]\!]$ we can infer by Lemmas 3 and 4 that $\langle a{:}B\rangle M' \in [\![\langle B\rangle]\!]$ and $M' \in SN$. We know that the degree of the cut-formula is in both terms equal, but $l(M') < l(M)$. Therefore we can apply the IH and infer that $\mathsf{Cut}(\langle a\rangle M', (x)N) \in SN$.

Commuting Reduction: $\mathsf{Cut}(\langle a\rangle M, (x)N) \longrightarrow M[a := (x)N]$. By assumption we have $\langle a{:}B\rangle M \in [\![\langle B\rangle]\!] = \textsc{neg}_{\langle B\rangle}([\![(B)]\!])$. We know that the commuting reduction is only applicable if M does not introduce a; therefore we have that $\langle a{:}C{\wedge}D\rangle M \notin \textsc{andright}_{\langle C\wedge D\rangle}([\![\langle C\rangle]\!], [\![\langle D\rangle]\!], [\![(C{\wedge}D)]\!])$ (where $B \equiv C{\wedge}D$). That means that $\langle a{:}B\rangle M \in \textsc{axioms}_{\langle B\rangle}$ or $\langle a{:}B\rangle M \in \textsc{binding}_{\langle B\rangle}([\![(B)]\!])$. In the first case we have $\mathsf{Cut}(\langle a\rangle M, (x)N) \longrightarrow M[a := (x)N] \equiv M$ (because M is an axiom and does not introduce a); M is strongly normalising by assumption.

In the second case we have that $M[a := (y{:}B)P] \in SN$ for all $(y{:}B)P \in [\![(B)]\!]$. Set $(y{:}B)P$ to $(x{:}B)N$ which is in $[\![(B)]\!]$ by assumption. Symmetric case is similar.

Case Logical Reduction I: $\mathsf{Cut}(\langle a\rangle \mathsf{Ax}(y, a), (x)N) \longrightarrow N\{x \mapsto y\}$. By assumption we know that $N \in SN$. This implies that $N\{x \mapsto y\} \in SN$. Symmetric case is similar.

Case Logical Reduction II: $\mathsf{Cut}(\langle c\rangle \mathsf{And}_R(\langle a\rangle\, S, \langle b\rangle\, T, c), (y)\mathsf{And}^1_L((x)\,U, y))$, where $B \equiv C{\wedge}D$. For more clarity we set $\langle c\rangle M \equiv \langle c{:}C{\wedge}D\rangle \mathsf{And}_R(\langle a\rangle S, \langle b\rangle T, c)$ and $(y)N \equiv (y{:}C{\wedge}D)\mathsf{And}^1_L((x)U, y)$.

$$\begin{array}{l}\mathsf{Cut}(\langle c\rangle \mathsf{And}_R(\langle a\rangle S, \langle b\rangle T, c), (y)\mathsf{And}^1_L((x)U, y))\\ \longrightarrow \mathsf{Cut}(\langle a\rangle S[c := (y)N], (x)U[y := \langle c\rangle M]).\end{array}$$

By assumption we know that $\langle c{:}C{\wedge}D\rangle M \in [\![\langle C{\wedge}D\rangle]\!]$ and $(y{:}C{\wedge}D)N \in [\![(C{\wedge}D)]\!]$. We have to show that $\mathsf{Cut}(\langle a{:}C\rangle S[c := (y)N], (x{:}C)U[y := \langle c\rangle M]) \in SN$. Since $\langle c\rangle M \in [\![\langle C{\wedge}D\rangle]\!] = \textsc{neg}_{\langle C\wedge D\rangle}([\![(C{\wedge}D)]\!])$ and $\langle c\rangle M \notin \textsc{axioms}_{\langle C\wedge D\rangle}$ we know that:

$\langle c{:}C{\wedge}D\rangle M \in \textsc{binding}_{\langle C\wedge D\rangle}([\![(C{\wedge}D)]\!])$ or
$\langle c{:}C{\wedge}D\rangle M \in \textsc{andright}_{\langle C\wedge D\rangle}([\![\langle C\rangle]\!], [\![\langle D\rangle]\!], [\![(C{\wedge}D)]\!])$.

Similarly

$(y{:}C{\wedge}D)N \in \textsc{binding}_{(C\wedge D)}([\![\langle C{\wedge}D\rangle]\!])$ or
$(y{:}C{\wedge}D)N \in \textsc{andleft}^1_{(C\wedge D)}([\![(C)]\!], [\![\langle C{\wedge}D\rangle]\!])$.

If $\langle c{:}C{\wedge}D\rangle M \in \textsc{binding}_{\langle C\wedge D\rangle}([\![(C{\wedge}D)]\!])$ we know that $M[c := (z)P] \in SN$ for all $(z{:}C{\wedge}D)P \in [\![(C{\wedge}D)]\!]$. By assumption $(y{:}C{\wedge}D)N \in [\![(C{\wedge}D)]\!]$ and therefore $M[c := (y)N] \equiv \mathsf{Cut}(\langle c\rangle M, (y)N) \in SN$. But then we also have that its reduct $\mathsf{Cut}(\langle a\rangle S[c := (y)N], (x)U[y := \langle c\rangle M]) \in SN$. Similarly for the case $(y{:}C{\wedge}D)N \in \textsc{binding}_{(C\wedge D)}([\![\langle C{\wedge}D\rangle]\!])$. It is left to show strong normalisation in the case where $\langle c{:}C{\wedge}D\rangle M \in \textsc{andright}_{\langle C\wedge D\rangle}([\![\langle C\rangle]\!], [\![\langle D\rangle]\!], [\![(C{\wedge}D)]\!])$ and $(y{:}C{\wedge}D)N \in \textsc{andleft}^1_{(C\wedge D)}([\![(C)]\!], [\![\langle C{\wedge}D\rangle]\!])$. We have $\langle a\rangle\, S[c := (y)P] \in [\![\langle C\rangle]\!]$ and $(x)\, U[y := \langle c\rangle Q] \in [\![(C)]\!]$ for all terms $(y{:}C{\wedge}D)P \in [\![(C{\wedge}D)]\!]$ and $\langle c{:}C{\wedge}D\rangle Q \in [\![\langle C{\wedge}D\rangle]\!]$. By assumption we know that $\langle c{:}C{\wedge}D\rangle M \in [\![\langle C{\wedge}D\rangle]\!]$ and $(y{:}C{\wedge}D)N \in [\![(C{\wedge}D)]\!]$; set $\langle c\rangle M$ for $\langle c\rangle Q$ and $(y)N$ for $(y)P$ respectively. Therefore we know that $\langle a\rangle\, S[c := (y)N] \in [\![\langle C\rangle]\!]$ and $(x)\, U[y := \langle c\rangle M] \in [\![(C)]\!]$. Furthermore, by Lemma 4 we have $S[c := (y)N] \in SN$ and $U[y :=$

$\langle c\rangle M] \in SN$. Because the degree of the cut-formula decreased we can apply the IH and infer that

$$\mathsf{Cut}(\langle a\rangle S[c := (y)N], (x)U[y := \langle c\rangle M]) \in SN.$$

We have shown that all immediate reducts of $\mathsf{Cut}(\langle a\rangle M, (x)N)$ are strongly normalising. Consequently $\mathsf{Cut}(\langle a\rangle M, (x)N)$ must be strongly normalising.

It is left to show that all well-typed terms are strongly normalising. To do so, we shall consider a special class of simultaneous substitutions, which are called safe. The principal property of safe substitutions $[\sigma_1]$ and $[\sigma_2]$ is that they can be commuted, i.e. $M[\sigma_1][\sigma_2] \equiv M[\sigma_2][\sigma_1]$.

Let $\hat{\sigma}$ be a set of substitutions of the form $[x := \langle a\rangle P]$ and $[b := (y)Q]$. Let us call the set of the x's and b's the domain of $\hat{\sigma}$ (written as $dom(\hat{\sigma})$); the set of named terms $(y)Q$ and co-named terms $\langle a\rangle P$ is called the co-domain of $\hat{\sigma}$ (written as $codom(\hat{\sigma})$). A *safe simultaneous substitution* (sss) is a set of substitutions where no variable clash between the domain and co-domain occurs (this can always be achieved by appropriate α-conversions, however, we omit a precise definition). The next lemma shows that a specific type of simultaneous substitutions is safe.

Lemma 6. *Let $\hat{\sigma}$ be of the form:*

$$\left\{ \bigcup_{i=0,\ldots,n} [x_i := \langle c\rangle \mathsf{Ax}(x_i, c)] \right\} \cup \left\{ \bigcup_{j=0,\ldots,m} [a_j := (y)\mathsf{Ax}(y, a_j)] \right\}$$

where the x_i's and a_i's are distinct names and co-names, respectively. Substitution $\hat{\sigma}$ is a sss.

Proof. Induction on the length of $\hat{\sigma}$.

Lemma 7. *For every term M (not necessarily strongly normalising) and for every sss $\hat{\sigma}$, such that $FN(M) \cup FC(M) \subseteq dom(\hat{\sigma})$ (i.e., $\hat{\sigma}$ is a closing substitution[2]) and for every $(x{:}B)P \in codom(\hat{\sigma})$ $(x{:}B)P \in [\![(B)]\!]$ and every $\langle a{:}C\rangle Q \in codom(\hat{\sigma})$ $\langle a{:}C\rangle Q \in [\![\langle C\rangle]\!]$, we have $M\hat{\sigma} \in SN$.*

Proof. We proceed by induction over the structure of M. We write $\hat{\sigma}, [\sigma]$ for the set $\hat{\sigma} \cup [\sigma]$ where $[\sigma] \notin \hat{\sigma}$.

Case $\mathsf{Ax}(x, a)$: We have to prove that: $\mathsf{Ax}(x, a)\ \hat{\sigma}, [x := \langle b\rangle P], [a := (y)Q] \in SN$. By definition of substitution $\mathsf{Ax}(x, a)\ \hat{\sigma}, [x := \langle b\rangle P], [a := (y)Q] \equiv \mathsf{Cut}(\langle b\rangle P, (y)Q)$. By assumption $\langle b{:}B\rangle P \in [\![\langle B\rangle]\!]$ and $(y{:}B)Q \in [\![(B)]\!]$. By Lemma 4 we know that $P \in SN$ and $Q \in SN$. Therefore we can apply Lemma 5 and can infer that $\mathsf{Cut}(\langle b\rangle P, (y)Q) \in SN$. Therefore $\mathsf{Ax}(x, a)\hat{\sigma}, [x := \langle b\rangle P], [a := (y)Q] \in SN$.

[2] All free names and co-names of M are amongst the domain of $\hat{\sigma}$.

Case $\mathsf{And}_R(\langle a\rangle M, \langle b\rangle N, c)$**:** We prove that $\mathsf{And}_R(\langle a\rangle M, \langle b\rangle N, c)\ \hat{\sigma}, [c := (z)R] \in SN$ where $(z{:}B\wedge C)R$ is an arbitrary named term in $[\![(B\wedge C)]\!]$. We can infer that $\mathsf{And}_R(\langle a\rangle M, \langle b\rangle N, c)\ \hat{\sigma}, [c{:=}(z)R] \equiv \mathsf{Cut}(\langle c\rangle \mathsf{And}_R(\langle a\rangle M\hat{\sigma}, \langle b\rangle N\hat{\sigma}, c), (z)R)$. By IH we know that $M\ \hat{\sigma}, [c := (x)S], [a := (y)P] \in SN$ and $N\ \hat{\sigma}, [c := (x)S], [b := (v)Q] \in SN$ for arbitrary $(y{:}B)P \in [\![\langle B\rangle]\!]$, $(v{:}C)Q \in [\![\langle C\rangle]\!]$ and $(x{:}B\wedge C)S \in [\![(B\wedge C)]\!]$.

Making appropriate α-conversions we have $(M\hat{\sigma})[c := (x)S][a := (y)P] \in SN$ and $(N\hat{\sigma})[c := (x)S][b := (v)Q] \in SN$. By definition of BINDING we have $\langle a{:}B\rangle\ (M\hat{\sigma})[c := (x)S] \in [\![\langle B\rangle]\!]$ and $\langle b{:}C\rangle (N\hat{\sigma})[c := (x)S] \in [\![\langle C\rangle]\!]$. Because $(x{:}B\wedge C)S$ is an arbitrary named term in the candidate $[\![(B\wedge C)]\!]$ we have by definition of ANDRIGHT$_{\langle B\wedge C\rangle}$ that $\langle c{:}B\wedge C\rangle \mathsf{And}_R(\langle a\rangle M\hat{\sigma}, \langle b\rangle N\hat{\sigma}, c) \in [\![\langle B\wedge C\rangle]\!]$. Furthermore we know by Lemma 4 that $\mathsf{And}_R(\langle a\rangle M\hat{\sigma}, \langle b\rangle N\hat{\sigma}, c) \in SN$.

For $(z{:}B\wedge C)R \in [\![(B\wedge C)]\!]$ we have by Lemma 4 that $R \in SN$. We can apply Lemma 5 and have $\mathsf{Cut}(\langle c\rangle \mathsf{And}_R(\langle a\rangle M\hat{\sigma}, \langle b\rangle N\hat{\sigma}, c), (z)R) \in SN$ and therefore $\mathsf{And}_R(\langle a\rangle M, \langle b\rangle N, c)\ \hat{\sigma}, [c := (z)R] \in SN$.

Case $\mathsf{And}_L^i((x)M, y)$ $(i = 1, 2)$: We have to prove that $\mathsf{And}_L^i((x)M, y)\ \hat{\sigma}, [y := \langle c\rangle R] \in SN$ where $\langle c{:}B_1\wedge B_2\rangle R$ is an arbitrary co-named term in $[\![\langle B_1\wedge B_2\rangle]\!]$. We have $\mathsf{And}_L^i((x)M, y)\ \hat{\sigma}, [y := \langle c\rangle R] \equiv \mathsf{Cut}(\langle c\rangle R, (y)\mathsf{And}_L^i((x)M\hat{\sigma}, y))$ by definition of substitution. By IH we know that $M\ \hat{\sigma}, [y := \langle a\rangle S], [x := \langle b\rangle T] \in SN$ for arbitrary $\langle a{:}B_1\wedge B_2\rangle S \in [\![\langle B_1\wedge B_2\rangle]\!]$, and arbitrary $(b{:}B_i)T \in [\![\langle B_i\rangle]\!]$. Making appropriate α-conversions we have $(M\hat{\sigma})[y := \langle a\rangle S][x := \langle b\rangle T] \in SN$. By definition of BINDING we have $(x{:}B_i)\ (M\hat{\sigma})[y := \langle a\rangle S] \in [\![(B_i)]\!]$. Since $\langle a{:}B_1\wedge B_2\rangle S$ is an arbitrary co-named term in $[\![\langle B_1\wedge B_2\rangle]\!]$ we have by definition of ANDLEFT$^i_{(B_1\wedge B_2)}$ that $(y{:}B_1\wedge B_2)\mathsf{And}_L^i((x)M\hat{\sigma}, y) \in [\![(B_1\wedge B_2)]\!]$. By Lemma 4 we can infer that $\mathsf{And}_L^i((x)M\hat{\sigma}, y) \in SN$. For $(c{:}B_1\wedge B_2)R \in [\![(B_1\wedge B_2)]\!]$ we have by Lemma 4 that $R \in SN$. We can apply Lemma 5 and have $\mathsf{Cut}(\langle c\rangle R, (y)\mathsf{And}_L^i((x)M\hat{\sigma}, y)) \in SN$. Therefore $\mathsf{And}_L^i((x)M, y)\ \hat{\sigma}, [y := \langle c\rangle R] \in SN$.

Case $\mathsf{Cut}(\langle a\rangle M, (x)N)$**:**

Subcase I: M is an axiom (case N being an axiom is similar). We have to show that $\mathsf{Cut}(\langle a\rangle \mathsf{Ax}(x, a), (y)N)\ [x := \langle b\rangle S], \hat{\sigma} \in SN$. By definition of substitution $\mathsf{Cut}(\langle a\rangle \mathsf{Ax}(x, a), (y)N)\ [x := \langle b\rangle S], \hat{\sigma} \equiv \mathsf{Cut}(\langle b\rangle S, (x)\ N\{x{\mapsto}y\}\hat{\sigma})$. By assumption we know that $\langle b{:}B\rangle S \in [\![\langle B\rangle]\!]$; using Lemma 4 we know that $S \in SN$. By assumption we know that $N\ \hat{\sigma}, [x := \langle b\rangle S], [y := \langle b\rangle S] \in SN$ for arbitrary $\langle b{:}B\rangle S \in [\![\langle B\rangle]\!]$. Because $\hat{\sigma}, [x := \langle b\rangle S], [y := \langle b\rangle S]$ is a safe simultaneous substitution we have (making appropriate α-conversions) $N\ \hat{\sigma}, [x := \langle b\rangle S], [y := \langle b\rangle S] \equiv (N\{y \mapsto x\}\hat{\sigma})\ [x := \langle b\rangle S]$. By definition of BINDING we know that $(x{:}B)\ N\{y{\mapsto}x\}\hat{\sigma} \in [\![(B)]\!]$. By Lemma 4 we can infer that $N\{y \mapsto x\}\hat{\sigma} \in SN$. Then we can apply Lemma 5 and can show that $\mathsf{Cut}(\langle b\rangle S, (x)\ N\{y{\mapsto}x\}\hat{\sigma}) \in SN$. Therefore $\mathsf{Cut}(\langle a\rangle \mathsf{Ax}(x, a), (y)N)\ \hat{\sigma}, [x := \langle b\rangle S] \in SN$.

Subcase II: M and N are not axioms. We prove that $\mathsf{Cut}(\langle a\rangle M, (x)N)\ \hat{\sigma} \in SN$. By IH we know that $M\ \hat{\sigma}, [a := (y)S] \in SN$ and $N\ \hat{\sigma}, [x := \langle b\rangle T] \in SN$ for arbitrary $(y{:}B)S \in [\![(B)]\!]$ and $\langle b{:}B\rangle T \in [\![\langle B\rangle]\!]$. Making appropriate α-conversions we know that $(M\hat{\sigma})[a := (y)S] \in SN$ and $(N\hat{\sigma})[x := \langle b\rangle T] \in SN$.

By definition of BINDING we can infer that $\langle a{:}B\rangle\ M\hat{\sigma} \in [\![\langle B\rangle]\!]$ and $(x{:}B)\ N\hat{\sigma} \in [\![(B)]\!]$. By Lemma 4 we have that $M\hat{\sigma} \in SN$ and $N\hat{\sigma} \in SN$. Therefore we can apply Lemma 5 and infer $\mathsf{Cut}(\langle a\rangle\ M\hat{\sigma}, (x)\ N\hat{\sigma}) \equiv \mathsf{Cut}(\langle a\rangle M, (x)N)\ \hat{\sigma} \in SN$.

We can now prove our main theorem.

Theorem 1. *All well-typed terms are strongly normalising.*

Proof. We know by Lemma 7 that for arbitrary well-typed terms M and arbitrary safe simultaneous substitution $\hat{\sigma}$, we have $M\hat{\sigma} \in SN$. Let $\hat{\sigma}$ be the safe simultaneous substitution from Lemma 6. Using Lemma 1 we can infer that either $M\hat{\sigma} \longrightarrow^{+} M$ or $M\hat{\sigma} \equiv M$. From this we have $M \in SN$.

This theorem can be extended to the full classical logic. To save space we give only the definitions for the set operators with implicational type:

$$\begin{aligned}\textsc{impleft}_{(B\supset C)} &: \mathcal{P}(CT_{\langle B\rangle}) \times \mathcal{P}(NT_{(C)}) \times \mathcal{P}(CT_{\langle B\supset C\rangle}) \to \mathcal{P}(NT_{(B\supset C)})\\ \textsc{impright}_{\langle B\supset C\rangle} &: \mathcal{P}(NT_{(B)}) \times \mathcal{P}(CT_{\langle C\rangle}) \times \mathcal{P}(NT_{(B\supset C)}) \to \mathcal{P}(CT_{\langle B\supset C\rangle})\end{aligned}$$

$$\begin{aligned}\textsc{impleft}_{(B\supset C)}(X,Y,Z) &\stackrel{\text{def}}{=} \{(z{:}B\supset C)\mathsf{Imp}_L(\langle a{:}B\rangle M, (x{:}C)N, z) \mid \\ &\quad \forall\ \langle c{:}B\supset C\rangle P \in Z.\langle a\rangle\ M[z := \langle c\rangle P] \in X \text{ and } (x)\ N[z := \langle c\rangle P] \in Y\}\\ \textsc{impright}_{\langle B\supset C\rangle}(X,Y,Z) &\stackrel{\text{def}}{=} \{\langle b{:}B\supset C\rangle\mathsf{Imp}_R((x{:}B)\langle a{:}C\rangle M, b) \mid \\ &\quad \forall\ (z{:}B\supset C)P \in Z, \forall\ \langle c{:}B\rangle S \in X.\langle a\rangle\ M[z := \langle c\rangle P][x := \langle c\rangle S] \in Y \text{ and}\\ &\quad \forall\ (z{:}B\supset C)P \in Z, \forall\ (y{:}C)T \in Y.(x)\ M[z := \langle c\rangle P][a := (y)T] \in X\}\end{aligned}$$

$$\begin{aligned}\textsc{neg}_{\langle B\supset C\rangle}(X) &\stackrel{\text{def}}{=} \textsc{axioms}_{\langle B\supset C\rangle} \cup \textsc{binding}_{\langle B\supset C\rangle}(X) \cup \textsc{impright}_{\langle B\supset C\rangle}([\![(B)]\!], [\![\langle C\rangle]\!], X)\\ \textsc{neg}_{(B\supset C)}(X) &\stackrel{\text{def}}{=} \textsc{axioms}_{(B\supset C)} \cup \textsc{binding}_{(B\supset C)}(X) \cup \textsc{impleft}_{(B\supset C)}([\![\langle B\rangle]\!], [\![(C)]\!], X)\end{aligned}$$

The strong normalisation proof can be easily extended using the definitions above. The only difficulty arises in Lemma 5 for the cut-elimination reduction for the connective $\supset$. The reduct of such a cut contains two nested cuts. Although the degree of the cut-formula decreases for the outer cut, the IH is not immediately applicable. In order to apply the induction hypothesis for the outer cut one has to show for the inner cut that:

$$\begin{aligned}&\langle a\rangle\mathsf{Cut}(\langle c\rangle N[z := \langle b\rangle\mathsf{Imp}_R((x)\langle a\rangle M, b)], (x)M[b := (z)\mathsf{Imp}_L(\langle c\rangle N, (y)P, z)]) \in [\![\langle C\rangle]\!] \text{ and}\\ &(x)\mathsf{Cut}(\langle a\rangle M[b := (z)\mathsf{Imp}_L(\langle c\rangle N, (y)P, z)], (y)P[z := \langle b\rangle\mathsf{Imp}_R((x)\langle a\rangle M, b)]) \in [\![(B)]\!]\end{aligned}$$

In the first case (the other being similar) one has to show that:

$$\mathsf{Cut}(\langle c\rangle N[z := \langle b\rangle\mathsf{Imp}_R((x)\langle a\rangle M, b)], (x)M[b := (z)\mathsf{Imp}_L(\langle c\rangle N, (y)P, z)])[a := (v)T] \in SN.$$

To infer this it is essential to know that a is not a free name in N and P (requirement of the reduction rule which can always be achieved by renaming a appropriately).

4 Conclusion

In this paper we presented a reduction system for cut-elimination in classical logic. One feature of the reduction system is to permute a subderivation of a

$$\dfrac{\dfrac{\dfrac{}{A\lor A\vdash A\lor A}\quad\dfrac{}{A\lor A\vdash A\lor A}}{(A\lor A)\lor(A\lor A)\vdash A\lor A}\,\lor_L\qquad\dfrac{\dfrac{\dfrac{}{A\vdash A}\quad\dfrac{}{A\vdash A}}{A\lor A\vdash A}\,\lor_L\quad\dfrac{\dfrac{}{A\vdash A}\quad\dfrac{}{A\vdash A}}{A\vdash A\land A}\,\land_R}{A\lor A\vdash A\land A}\,Cut}{(A\lor A)\lor(A\lor A)\vdash A\land A}\,Cut$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A\vdash A}\quad\dfrac{}{A\vdash A}}{A\vdash A\land A}\,\land_R\quad\dfrac{\dfrac{}{A\vdash A}\quad\dfrac{}{A\vdash A}}{A\vdash A\land A}\,\land_R}{A\lor A\vdash A\land A}\,\lor_L\qquad\dfrac{\dfrac{\dfrac{}{A\vdash A}\quad\dfrac{}{A\vdash A}}{A\lor A\vdash A}\,\lor_L\quad\dfrac{\dfrac{}{A\vdash A}\quad\dfrac{}{A\vdash A}}{A\lor A\vdash A}\,\lor_L}{A\lor A\vdash A\land A}\,\land_R}{(A\lor A)\lor(A\lor A)\vdash A\land A}\,\lor_L$$

Fig. 3. A proof in G3c and a cut-free normalform which is not reachable by a cut-elimination procedure using colours as in LK^{tq}.

commuting cut directly to the place(s) where the cut-formula is a main formula. This is an idea taken from the work in LK^{tq} [6]. However we do not require their colour annotations on the cut-formulae (in fact no additional information is required at all). One consequence is that, in general, more normal forms can be reached from a given proof containing cuts (see Figure 3 for an example). Because of the fewer constraints on our reduction system strong normalisation cannot be proved by translating every reduction to a series of reductions in proof-nets as done for LK^{tq}. The use of a term calculus for sequent derivations allowed us to use directly proof techniques from the λ^{Sym}-calculus [1] to prove strong normalisation. This use of syntax to study proof structures is part of a on-going research project [3, 19].

The result presented in this paper can be extended to the first-order calculus and can be adapted to LK or free-style LK^{tq}. There are many directions for further work. For example what is the precise correspondence in the intuitionistic case between normalisation and our strongly normalising cut-elimination procedure? For classical logic the correspondence between our cut-elimination procedure and normalisation in, for example, Parigot's $\lambda\mu$ [17] is another interesting question. Some of these problems will be addressed in Urban's PhD-thesis.

Acknowledgements: We should like to thank Roy Dyckhoff and Martin Hyland for their help and encouragement. The work has greatly benefited from discussions with Harold Schellinx and Jean-Baptiste Joinet on LK^{tq}. Urban is supported by a scholarship from the DAAD. Bierman is supported by EPSRC Grant GR-M04716 and Gonville & Caius College, Cambridge.

References

1. F. Barbanera and S. Berardi. A Symmetric Lambda Calculus for "Classical" Program Extraction. In *Theoretical Aspects of Computer Software*, volume 789 of *LNCS*, pages 495–515. Springer Verlag, 1994.

2. H. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 1981.
3. G. M. Bierman. Some Lectures on Proof Theory. Notes from course given at PUC-Rio, November 1997.
4. E. T. Bittar. Strong Normalisation Proofs for Cut-Elimination in Gentzen's Sequent Calculi. In *Logic, Algebra and Computer Science*, volume 46 of *Banach-Center Publications*, pages 179–225, 1999.
5. E. A. Cichon, M. Rusinowitch, and S. Selhab. Cut-Elimination and Rewriting: Termination Proofs. Technical Report, 1996.
6. V. Danos, J.-B. Joinet, and H. Schellinx. A New Deconstructive Logic: Linear Logic. *Journal of Symbolic Logic*, 62(3):755–807, 1997.
7. A. G. Dragalin. *Mathematical Intuitionism: Introduction to Proof Theory*, volume 67 of *Translations of Mathematical Monographs*. American Mathematical Society, 1988.
8. R. Dyckhoff and L. Pinto. Cut-Elimination and a Permutation-Free Sequent Calculus for Intuitionistic Logic. *Studia Logica*, 60(1):107–118, 1998.
9. J. Gallier. Constructive Logics. Part I: A Tutorial on Proof Systems and Typed λ-calculi. *Theoretical Computer Science*, 110(2):249–239, 1993.
10. G. Gentzen. Untersuchungen über das logische Schließen I and II. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1935.
11. J.-Y. Girard. Linear Logic. *Theoretical Computer Science*, 50(1):1–102, 1987.
12. J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types*, volume 7 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1989.
13. E. H. Hauesler and L. C. Pereira. Gentzen's Second Consistency Proof and Strong Cut-Elimination. *Logique and Analyse*, 154:95–111, 1996.
14. H. Herbelin. A λ-calculus Structure Isomorphic to Sequent Calculus Structure. In *Computer Science Logic*, volume 933 of *LNCS*, pages 67–75. Springer Verlag, 1995.
15. J.-B. Joinet, H. Schellinx, and L. Tortora de Falco. SN and CR for Free-Style LK^{tq}: Linear Decorations and Simulation of Normalisation. Preprint No. 1067, Utrecht University, Department of Mathematics, 1998.
16. S. C. Kleene. *Introduction to Metamathematics*. North-Holland, 1952.
17. M. Parigot. $\lambda\mu$-calculus: An Algorithmic Interpretation of Classical Logic. In *Logic Programming and Automated Deduction*, volume 624 of *LNCS*, pages 190–201. Springer Verlag, 1992.
18. A. S. Troelstra and H. Schwichtenberg. *Basic Proof Theory*, volume 43 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1996.
19. C. Urban. First-Year Report: Computational Content of Classical Proofs. Technical Report, 1997.
20. J. Zucker. The Correspondence Between Cut-Elimination and Normalisation. *Annals of Mathematical Logic*, 7:1–112, 1974.