Confluence of Cut-Elimination Procedures for the Intuitionistic Sequent Calculus

Kentaro Kikuchi

RIEC, Tohoku University
Katahira 2-1-1, Aoba-ku, Sendai 980-8577, Japan
kentaro@nue.riec.tohoku.ac.jp

Abstract. We prove confluence of two cut-elimination procedures for the implicational fragment of a standard intuitionistic sequent calculus. One of the cut-elimination procedures uses global proof transformations while the other consists of local ones. Both of them include permutation of cuts to simulate β-reduction in an isomorphic image of the λ-calculus. We establish the confluence properties through a conservativity result on the cut-elimination procedures.

Keywords: Sequent calculus, Cut-elimination, Confluence, λ-calculus, Explicit substitution.

1 Introduction

Gentzen's cut-elimination theorem [4] has long been a great influence on logic and theoretical computer science. Recent development of structural proof theory is revealing the computational aspect of cut-elimination procedures in the same sense that proof transformations in natural deduction play through the Curry-Howard correspondence [7]. In [8], the author identified a subset of proofs in a standard sequent calculus that correspond to simply typed λ-terms, and defined a reduction relation on those proofs that precisely corresponds to β-reduction of the simply typed λ-calculus. Since the reduction relation is simulated by a local-step cut-elimination procedure, the system of proof terms for the sequent calculus can be considered as a syntactical extension of the λ-calculus including reductions. It is worth noticing that the correspondence holds also for the type-free case, so the reduction system in [8] can simulate the type-free λ-calculus, which means that it is strong enough to represent all computations.

In this paper, we study confluence of a cut-elimination procedure based on the one introduced in [8]. Since the reduction system in [8] is not confluent, we modify one of the reduction rules to a more restricted form. The resulting system is still strong enough to simulate β-reduction in the isomorphic image of the λ-calculus. We also consider another cut-elimination procedure which includes global proof transformations in the style of [2]. The reduction system representing the cut-elimination procedure is similar to one considered in [3], which uses meta-operations like meta-substitution in the λ-calculus.

It is well-known that a local-step cut-elimination procedure has a similarity to explicit substitution calculi. Our proof method is essentially the one often

S.B. Cooper, B. Löwe, and A. Sorbi (Eds.): CiE 2007, LNCS 4497, pp. 398–407, 2007.

used in the field of explicit substitutions (see, e.g. [1]), called the interpretation method [6]. This method projects reduction steps with explicit substitutions onto those using meta-substitution, and reduces the confluence problem of an explicit substitution calculus to that of the original λ-calculus. To apply this method to the case of a cut-elimination procedure, we need to find an appropriate reduction using meta-operations. Although meta-operations are used in the reduction system for the global cut-elimination procedure mentioned above, it turns out that the system is not appropriate for a target calculus of the method because proving confluence of it has a delicate matter that is not present in the case of the usual λ-calculus. So we define another reduction relation on a certain class of proof terms, and first prove its confluence by the method of parallel reduction [10]. Confluence of the two cut-elimination procedures is inferred from confluence of this reduction by the interpretation method.

Danos et al. [2] proved confluence of their cut-elimination procedures with global proof transformations, depending on confluence of proof nets [5]. In this paper, we give a direct proof of confluence of a similar cut-elimination procedure, using proof terms and meta-operations on them. Our method works also for cut-elimination procedures consisting of local proof transformations and for underlying untyped calculi allowing non-terminating computations.

The paper is organized as follows. In Section 2 we introduce sequent calculus and cut-elimination procedures. In Section 3 we study a subcalculus and meta-operations from the reduction systems. In Section 4 we define another reduction relation and prove its confluence. In Section 5 we prove confluence of the cut-elimination procedures. In Section 6 we conclude by suggestions for future work.

To save space we omit details of proofs, but a full version with all details is available at `http://www.nue.riec.tohoku.ac.jp/user/kentaro/`.

2 Sequent Calculus and Cut-Elimination Procedures

In this section we introduce a term notation for proofs in a standard sequent calculus for intuitionistic implicational logic, following [8]. Our cut-elimination procedures are represented as reduction rules for those terms.

First, the set of raw terms for sequent proofs is defined by the grammar: $t ::= x \mid \lambda x.t \mid \langle xt/x\rangle t \mid [t/x]t$ where x ranges over a denumerable set of variables. $\langle \_\_/\_\rangle\_$ and $[\_/\_]\_$ are function symbols like explicit substitutions and not meta-substitution ($[\_/\_]\_$ is called the cut-constructor). We use letters x, y, z, w for variables and t, s, r, u for terms. The notions of free and bound variables are defined as usual, with an additional clause that the variable x in $\langle ys/x\rangle t$ or $[s/x]t$ binds the free occurrences of x in t. The set of free variables of a term t is denoted by $FV(t)$. We often use the notation $\langle \underline{x}s/y\rangle t$ to denote $\langle xs/y\rangle t$ if $x \notin FV(s) \cup FV(t)$. The symbol $\equiv$ denotes syntactical equality modulo α-conversion; so for example, $\langle zr/x\rangle\langle \underline{x}s/y\rangle t \equiv \langle zr/w\rangle\langle \underline{w}s/y\rangle t$.

The term assignment for sequent proofs of intuitionistic implicational logic is given in Table 1. We define a context, ranged over by Γ, as a finite set of pairs $\{x_1 : A_1, \ldots, x_n : A_n\}$ where the variables are pairwise distinct. The context

Table 1. Sequent calculus and local cut-elimination

$$Ax \; \frac{}{\Gamma, x : A \vdash x : A} \qquad L\supset \; \frac{\Gamma \vdash s : A \quad \Gamma, y : B \vdash t : C}{\Gamma, x : A \supset B \vdash \langle xs/y\rangle t : C} \; y \notin \Gamma$$

$$R\supset \; \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \supset B} \; x \notin \Gamma \qquad Cut \; \frac{\Gamma \vdash s : A \quad \Gamma, x : A \vdash t : B}{\Gamma \vdash [s/x]t : B} \; x \notin \Gamma$$

$\langle \underline{x}s/y\rangle t$ is used for $\langle xs/y\rangle t$ when $x \notin FV(s) \cup FV(t)$. In that case we assume $x \notin \Gamma$ in the rule $L\supset$.

(1)	$[t/x]y \to y \quad (y \not\equiv x)$
(2)	$[t/x]x \to t$
(3)	$[s/x](\lambda y.t) \to \lambda y.[s/x]t$
(4)	$[r/z]\langle xs/y\rangle t \to \langle x([r/z]s)/y\rangle [r/z]t \quad (x \not\equiv z)$
(5)	$[r/x]\langle xs/y\rangle t \to [r/x]\langle \underline{x}([r/x]s)/y\rangle [r/x]t \quad \text{if } x \in FV(s) \cup FV(t)$
(6)	$[z/x]\langle \underline{x}s/y\rangle t \to \langle zs/y\rangle t$
(7′)	$[\langle xs/y\rangle t/z]\langle \underline{z}s'/w\rangle t' \to \langle xs/y\rangle [t/z]\langle \underline{z}s'/w\rangle t'$
(*Beta*)	$[\lambda z.r/x]\langle \underline{x}s/y\rangle t \to [[s/z]r/y]t$
($Perm_1$)	$[[r/x]\langle \underline{x}s/y\rangle t/z]\langle \underline{z}s'/w\rangle t' \to [r/x][\langle \underline{x}s/y\rangle t/z]\langle \underline{z}s'/w\rangle t'$
($Perm_2$)	$[u/w][\lambda z.r/x]\langle \underline{x}s/y\rangle t \to [[u/w](\lambda z.r)/x][u/w]\langle \underline{x}s/y\rangle t$

$\Gamma, x : A$ denotes the union $\Gamma \cup \{x : A\}$, and $x \notin \Gamma$ means that x does not appear in Γ. For precise representation of proofs by terms, we should specify formulas on binders, but we will omit them for brevity. If $x \notin FV(s) \cup FV(t)$ in the term $\langle xs/y\rangle t$, we assume $x \notin \Gamma$ in the rule $L\supset$, which means the formula $A \supset B$ is introduced without implicit contraction.

The reduction rules in Table 1 define a cut-elimination procedure consisting of local proof transformations. The reduction relation $\to_{\mathrm{cut}}$ is defined by the contextual closures of these reduction rules. We use $\xrightarrow{+}_{\mathrm{cut}}$ for its transitive closure, and $\xrightarrow{*}_{\mathrm{cut}}$ for its reflexive transitive closure. These kinds of notations are also used for the notions of other reductions in this paper.

The reduction system without the rule (*Beta*) is denoted by x. This subcalculus plays an important role in this paper and is studied in detail in Section 3.

The reduction rules (1) through (5) correspond to cut-elimination steps that permute a cut upwards through its right subproof. The rules (6) and (7′) correspond to steps permuting a cut upwards through its left subproof. The rule (*Beta*) corresponds to the key-case which breaks a cut on an implication into two cuts on its subformulas. The rules ($Perm_1$) and ($Perm_2$) permute two cuts

Table 2. Global cut-elimination

$(Beta)$	$[\lambda z.r/x]\langle \underline{x}s/y\rangle t$	$\to$	$[[s/z]r/y]t$	
$(left)$	$[u/x]\langle \underline{x}s/y\rangle t$	$\to$	$\langle \pmb{\{}u\pmb{\}}s/y\rangle t$	if u is not of the form $\lambda z.r$
$(right)$	$[u/x]r$	$\to$	$\{u/x\}r$	if r is not of the form $\langle \underline{x}s/y\rangle t$

where $\{\_/\_\}\_$ and $\langle\pmb{\{}\_\pmb{\}}\_/\_\rangle\_$ are the meta-operations defined as follows:

$$
\begin{aligned}
\{u/x\}y &=_{def} y \qquad (y \not\equiv x)\\
\{u/x\}x &=_{def} u\\
\{u/x\}(\lambda y.t) &=_{def} \lambda y.\{u/x\}t\\
\{u/x\}\langle zs/y\rangle t &=_{def} \langle z(\{u/x\}s)/y\rangle\{u/x\}t \qquad (z \not\equiv x)\\
\{u/x\}\langle xs/y\rangle t &=_{def} [u/x]\langle \underline{x}(\{u/x\}s)/y\rangle\{u/x\}t\\
\{u/x\}[s/y]t &=_{def} [\{u/x\}s/y]\{u/x\}t
\end{aligned}
$$

$$
\begin{aligned}
\langle\pmb{\{}z\pmb{\}}s/y\rangle t &=_{def} \langle zs/y\rangle t\\
\langle\pmb{\{}\lambda z.r\pmb{\}}s/y\rangle t &=_{def} [\lambda z.r/x]\langle \underline{x}s/y\rangle t\\
\langle\pmb{\{}\langle zs'/w\rangle r\pmb{\}}s/y\rangle t &=_{def} \langle zs'/w\rangle\langle\pmb{\{}r\pmb{\}}s/y\rangle t\\
\langle\pmb{\{}[s'/w]r\pmb{\}}s/y\rangle t &=_{def} [s'/w]\langle\pmb{\{}r\pmb{\}}s/y\rangle t
\end{aligned}
$$

with some restrictions. In ($Perm_1$), the left rule over the lower cut is another cut, and the right rules over both cuts must be $L\supset$ that introduces the cut-formula without implicit contraction. In ($Perm_2$), the right rule over the lower cut is another cut, which must construct a proof corresponding to a redex of the rule ($Beta$).

The original cut-elimination procedure in [8] uses the following rule (7) instead of (7′):

$$(7) \quad [\langle xs/y\rangle t/z]r \to \langle xs/y\rangle[t/z]r$$

This rule makes the cut-elimination procedure non-confluent (e.g., the critical pair $w \leftarrow [\langle xs/y\rangle t/z]w \to \langle xs/y\rangle[t/z]w$ is not joinable). For a confluent cut-elimination procedure, it is therefore necessary to restrict reductions. The rule (7′) restricts the rule (7) so that the right rule over the cut must be $L\supset$ that introduces the cut-formula without implicit contraction. As shown in [8], this cut-elimination procedure is still strong enough to simulate β-reduction in the isomorphic image of the λ-calculus.

Table 2 presents another cut-elimination procedure which includes global proof transformations. The cut-elimination procedure is implemented by reduction rules that use meta-operations $\{\_/\_\}\_$ and $\langle\pmb{\{}\_\pmb{\}}\_/\_\rangle\_$, analogously to proof transformations in natural deduction. The operation $\langle\pmb{\{}\_\pmb{\}}\_/\_\rangle\_$ corresponds to the cut-elimination process where the right rule over the cut is $L\supset$ introducing the cut-formula without implicit contraction, and the cut is permuted upwards through its left subproof. Note that the conditions of (*left*) and (*right*) make the cut-elimination procedure first permute a cut upwards through its right subproof and then through its left subproof. The reduction relation generated by the rules (*Beta*), (*left*) and (*right*) is denoted by $\rightarrow_{\text{gcut}}$.

The following lemma is immediate from the definition of $\{\_/\_\}\_$.

Lemma 1. *If $x \notin FV(t)$ then $\{u/x\}t \equiv t$.*

Proof. By induction on the structure of t. □

3 The Subcalculus x and Meta-operations

In this section we study properties of the subcalculus $\mathtt{x}$ which is the reduction system in Table 1 without the rule (*Beta*). In the typed case, it corresponds to the cut-elimination steps except the key-case, i.e., the case where both left and right rules over the cut rule introduce the cut-formula. We show that the subcalculus $\mathtt{x}$ is strongly normalizing and confluent, and investigate its relation to the meta-operations in Table 2.

First we give a technical definition to prove strong normalization of the subcalculus $\mathtt{x}$.

Definition 1. *A term $[s/x]t$ is called an* application term *if t is one of the forms: $[u/w]\langle \underline{x}s'/y\rangle t'$, $\langle \underline{x}s'/y\rangle t'$ and $[\langle \underline{x}s'/y\rangle t'/z]\langle \underline{z}s''/w\rangle t''$, where x occurs only once in t.*

Lemma 2. *If $[s/x]t$ is an application term and $t \rightarrow_{\mathtt{x}} t'$, then $[s/x]t'$ is also an application term.*

Proof. It suffices to check each case. □

Proposition 1. *The subcalculus $\mathtt{x}$ is strongly normalizing.*

Proof. The proof is by interpretation. We define a function h as follows:

$$\begin{array}{ll} h(x) & =_{def} 1 \\ h(\lambda x.t) & =_{def} h(t)+1 \\ h(\langle xs/y\rangle t) & =_{def} h(s)+h(t)+1 \\ h([s/x]t) & =_{def} \begin{cases} (h(s)+1)^2 \times h(t) & \text{if } [s/x]t \text{ is an application term} \\ (h(s)+1)^{2\times h(t)} & \text{otherwise} \end{cases} \end{array}$$

and observe that if $t \rightarrow_{\mathtt{x}} t'$ then $h(t) > h(t')$. If $t \equiv [s/x]r$ is an application term and $r \rightarrow_{\mathtt{x}} r'$, then we use Lemma 2. □

Proposition 2. *The subcalculus $\mathtt{x}$ is confluent.*

Proof. By Newman's Lemma, it suffices to check the local confluence. There are two critical pairs caused by the rules $(7')$ and $(Perm_1)$, and by $(Perm_1)$ and $(Perm_1)$, both of which are joinable. □

As a result, we can define the unique $\mathtt{x}$-normal form of each term.

Definition 2. *The unique* $\mathtt{x}$*-normal form of a term* t *is denoted by* $\mathtt{x}(t)$*.*

A term in which every cut-constructor forms a redex of the rule (*Beta*) is called a *Beta-term*. The relation between *Beta*-terms and $\mathtt{x}$-normal forms is as follows.

Proposition 3. t *is a Beta-term if and only if* t *is in* $\mathtt{x}$*-normal form.*

Proof. The only if part is by induction on the structure of *Beta*-terms. We prove the if part by induction on the structure of t. Suppose that t is in $\mathtt{x}$-normal form. Then by the induction hypothesis, all subterms of t are *Beta*-terms. Now, if t is not a *Beta*-term then t is of the form $[u/x]r(\not\equiv [\lambda z.r'/x]\langle \underline{x}s/y\rangle t')$ where u, r are *Beta*-terms. In this case, t is an $\mathtt{x}$-redex, which is a contradiction. □

The next lemma shows that the subcalculus $\mathtt{x}$ correctly simulates the meta-operations on *Beta*-terms.

Lemma 3. *Let* u, t, s *be Beta-terms. Then*

1. $[u/x]t \stackrel{*}{\to}_{\mathtt{x}} \{u/x\}t$,
2. $[u/x]\langle \underline{x}s/y\rangle t \stackrel{*}{\to}_{\mathtt{x}} \langle \{\!\{u\}\!\}s/y\rangle t$*. Moreover,* $\langle \{\!\{u\}\!\}s/y\rangle t$ *is a Beta-term, hence* $\mathtt{x}([u/x]\langle \underline{x}s/y\rangle t) \equiv \langle \{\!\{u\}\!\}s/y\rangle t$*.*

Proof.

1. By induction on the structure of t.
2. By induction on the structure of u. □

Next we show that $\to_{\mathrm{gcut}}$ is sufficient to reach $\mathtt{x}$-normal forms.

Lemma 4. *Let* u, s, t *be Beta-terms. Then*

1. $[u/x]\langle \underline{x}s/y\rangle t \stackrel{*}{\to}_{\mathrm{gcut}} \mathtt{x}([u/x]\langle \underline{x}s/y\rangle t)$,
2. $\{u/x\}t \stackrel{*}{\to}_{\mathrm{gcut}} \mathtt{x}(\{u/x\}t)$.

Proof.

1. If $u \equiv \lambda z.r$ then $[u/x]\langle \underline{x}s/y\rangle t \equiv \mathtt{x}([u/x]\langle \underline{x}s/y\rangle t)$. If u is not of the form $\lambda z.r$, then $[u/x]\langle \underline{x}s/y\rangle t' \to_{\mathit{left}} \langle \{\!\{u\}\!\}s/y\rangle t' \equiv \mathtt{x}([u/x]\langle \underline{x}s/y\rangle t')$ by Lemma 3 (2).
2. By induction on the structure of t. □

Lemma 5. $t \stackrel{*}{\to}_{\mathrm{gcut}} \mathtt{x}(t)$*.*

Proof. By induction on the structure of t. □

The following lemmas are essential to the parallel reduction method in the next section. Note that $\{u/x\}\langle \{\!\{t\}\!\}s/y\rangle t' \equiv \langle \{\!\{\{u/x\}t\}\!\}s/y\rangle t'$ instead of Lemma 7 does not hold in general; for example, $\{z/x\}\langle \{\!\{x\}\!\}w/y\rangle w' \equiv [z/x]\langle xw/y\rangle w' \not\equiv \langle zw/y\rangle w' \equiv \langle \{\!\{\{z/x\}x\}\!\}w/y\rangle w'$. This makes it difficult to apply a direct parallel reduction method to $\to_{\mathrm{gcut}}$. So we consider the meta-operation $\{\_/\_\}\_$ followed by $\mathtt{x}$-reductions to $\mathtt{x}$-normal forms (i.e., $\mathtt{x}(\{\_/\_\}\_)$), and in the next section we define another reduction relation that matches such operation.

Lemma 6. $\langle \{\!\{\langle \{\!\{u\}\!\}s/y\rangle t\}\!\}s'/y'\rangle t' \equiv \langle \{\!\{u\}\!\}s/y\rangle \langle \{\!\{t\}\!\}s'/y'\rangle t'$*.*

Proof. By induction on the structure of u. □

Lemma 7. *Let u, t, s, t' be Beta-terms. Then*
$\mathtt{x}(\{u/x\}\langle\{\!\{t\}\!\}s/y\rangle t') \equiv \langle\{\!\{\mathtt{x}(\{u/x\}t)\}\!\}\mathtt{x}(\{u/x\}s)/y\rangle\mathtt{x}(\{u/x\}t')$.
In particular, if $x \notin FV(s) \cup FV(t')$ then
$\mathtt{x}(\{u/x\}\langle\{\!\{t\}\!\}s/y\rangle t') \equiv \langle\{\!\{\mathtt{x}(\{u/x\}t)\}\!\}s/y\rangle t'$.

Proof. By induction on the structure of t. □

Lemma 8. *Let u, s, t be Beta-terms with $y \notin FV(u)$. Then*
$\mathtt{x}(\{u/x\}\mathtt{x}(\{s/y\}t)) \equiv \mathtt{x}(\{\mathtt{x}(\{u/x\}s)/y\}\mathtt{x}(\{u/x\}t))$.

Proof. By induction on the structure of t. □

4 Confluence of β-Reduction

In this section we introduce another reduction relation on *Beta*-terms and show that it is confluent by the parallel reduction method [10]. Confluence of the two cut-elimination procedures is proved using projections onto this reduction.

The reduction relation $\to_\beta$ on *Beta*-terms is defined by the contextual closure of the rule:

$$(\beta) \quad [\lambda z.r/x]\langle \underline{x}s/y\rangle t \to \mathtt{x}(\{\mathtt{x}(\{s/z\}r)/y\}t)$$

This reduction relation is indeed an extension of β-reduction on pure terms (i.e., the isomorphic image of λ-terms) in [8].

Proposition 4. *Let t, t' be Beta-terms.*

1. *If $t \to_\beta t'$ then $t \xrightarrow{+}_{\mathrm{cut}} t'$.*
2. *If $t \to_\beta t'$ then $t \xrightarrow{+}_{\mathrm{gcut}} t'$.*

Proof. By induction on the reduction relation $\to_\beta$. We treat the case where the reduction is at the root. Then

$$\begin{aligned}
[\lambda z.r/x]\langle \underline{x}s/y\rangle t_0 &\to_{Beta} [[s/z]r/y]t_0 \\
&\xrightarrow{*}_{\mathtt{x}} \mathtt{x}([\mathtt{x}([s/z]r)/y]t_0) && (*)\\
&\equiv \mathtt{x}([\mathtt{x}(\{s/z\}r)/y]t_0) && \text{(by Lemma 3 (1))}\\
&\equiv \mathtt{x}(\{\mathtt{x}(\{s/z\}r)/y\}t_0) && \text{(by Lemma 3 (1))}
\end{aligned}$$

where the step $(*)$ can also be established with $\xrightarrow{*}_{\mathrm{gcut}}$ by Lemma 5. □

The parallel reduction $\Rightarrow$ for $\to_\beta$ is defined by the rules in Table 3.

Lemma 9. *For every Beta-term t, $t \Rightarrow t$.*

Proof. By induction on the structure of t. □

Table 3. Parallel reduction

$$\frac{}{x \Rightarrow x}\ (pr_1) \qquad \frac{t \Rightarrow t'}{\lambda x.t \Rightarrow \lambda x.t'}\ (pr_2) \qquad \frac{s \Rightarrow s' \quad t \Rightarrow t'}{\langle xs/y\rangle t \Rightarrow \langle xs'/y\rangle t'}\ (pr_3)$$

$$\frac{r \Rightarrow r' \quad s \Rightarrow s' \quad t \Rightarrow t'}{[\lambda z.r/x]\langle \underline{x}s/y\rangle t \Rightarrow [\lambda z.r'/x]\langle \underline{x}s'/y\rangle t'}\ (pr_4)$$

$$\frac{r \Rightarrow r' \quad s \Rightarrow s' \quad t \Rightarrow t'}{[\lambda z.r/x]\langle \underline{x}s/y\rangle t \Rightarrow \mathtt{x}(\{\mathtt{x}(\{s'/z\}r')/y\}t')}\ (pr_5)$$

Lemma 10

1. *If* $t \rightarrow_\beta t'$ *then* $t \Rightarrow t'$*.*
2. *If* $t \Rightarrow t'$ *then* $t \xrightarrow{*}_\beta t'$*.*
3. *If* $u \Rightarrow u'$*,* $s \Rightarrow s'$ *and* $t \Rightarrow t'$ *then* $\langle\{\!\{u\}\!\}s/y\rangle t \Rightarrow \langle\{\!\{u'\}\!\}s'/y\rangle t'$*.*
4. *If* $u \Rightarrow u'$ *and* $t \Rightarrow t'$ *then* $\mathtt{x}(\{u/x\}t) \Rightarrow \mathtt{x}(\{u'/x\}t')$*.*

Proof.

1. By induction on the reduction relation $\rightarrow_\beta$.
2. By induction on the definition of $t \Rightarrow t'$.
3. By induction on the definition of $u \Rightarrow u'$.
4. By induction on the definition of $t \Rightarrow t'$. □

Definition 3. *For each Beta-term* t*, the term* $t^\star$ *is defined inductively as follows:*

1. $x^\star =_{def} x$*,*
2. $(\lambda x.t)^\star =_{def} \lambda x.t^\star$*,*
3. $(\langle xs/y\rangle t)^\star =_{def} \langle xs^\star/y\rangle t^\star$*,*
4. $([\lambda z.r/x]\langle \underline{x}s/y\rangle t)^\star =_{def} \mathtt{x}(\{\mathtt{x}(\{s^\star/z\}r^\star)/y\}t^\star)$*.*

Lemma 11. *If* $t \Rightarrow t'$ *then* $t' \Rightarrow t^\star$*.*

Proof. By induction on the definition of $t \Rightarrow t'$. □

Lemma 12. *If* $t \Rightarrow t_1$ *and* $t \Rightarrow t_2$ *then there is* t' *such that* $t_1 \Rightarrow t'$ *and* $t_2 \Rightarrow t'$*.*

Proof. By Lemma 11. □

Theorem 1. *The reduction relation* $\rightarrow_\beta$ *is confluent.*

Proof. By Lemmas 10 and 12. □

Lemma 13. *Let* u, t *be Beta-terms.*

1. *If* $u \rightarrow_\beta u'$ *then* $\mathtt{x}(\{u/x\}t) \xrightarrow{*}_\beta \mathtt{x}(\{u'/x\}t)$*.*
2. *If* $t \rightarrow_\beta t'$ *then* $\mathtt{x}(\{u/x\}t) \xrightarrow{*}_\beta \mathtt{x}(\{u/x\}t')$*.*

Proof. These are derived from Lemmas 9 and 10. □

5 Confluence of Cut-Elimination Procedures

In this section we complete the proofs of confluence of the cut-elimination procedures. We also establish a conservativity result among the cut-elimination procedures and β-reduction on *Beta*-terms.

Lemma 14

1. $\mathtt{x}(\langle \{\!\!\{u\}\!\!\} s/y\rangle t) \equiv \langle \{\!\!\{\mathtt{x}(u)\}\!\!\} \mathtt{x}(s)/y\rangle \mathtt{x}(t)$,
2. $\mathtt{x}(\{u/x\}t) \equiv \mathtt{x}(\{\mathtt{x}(u)/x\}\mathtt{x}(t))$.

Proof.

1. By induction on the structure of u.
2. By induction on the structure of t. □

The next two lemmas show that the cut-elimination procedures project onto β-reduction on *Beta*-terms.

Lemma 15. *If* $t \to_{\mathrm{gcut}} t'$ *then* $\mathtt{x}(t) \xrightarrow{*}_{\beta} \mathtt{x}(t')$.

Proof. By induction on the reduction relation $\to_{\mathrm{gcut}}$. □

Lemma 16. *If* $t \to_{\mathrm{cut}} t'$ *then* $\mathtt{x}(t) \xrightarrow{*}_{\beta} \mathtt{x}(t')$.

Proof. If $t \to_{\mathtt{x}} t'$ then $\mathtt{x}(t) \equiv \mathtt{x}(t')$. So it suffices to show that if $t \to_{Beta} t'$ then $\mathtt{x}(t) \xrightarrow{*}_{\beta} \mathtt{x}(t')$. This is proved in a similar way to Lemma 15. □

Now we have a conservativity result among the reductions on *Beta*-terms.

Theorem 2. *For any Beta-terms* t, t'*, the following are equivalent.*

1. $t \xrightarrow{*}_{\mathrm{gcut}} t'$
2. $t \xrightarrow{*}_{\mathrm{cut}} t'$
3. $t \xrightarrow{*}_{\beta} t'$

Proof. By Lemmas 15 and 16, and Proposition 4. □

We are now ready to show that the reduction relations $\to_{\mathrm{gcut}}$ and $\to_{\mathrm{cut}}$ are confluent, using confluence of $\to_{\beta}$ on *Beta*-terms (Theorem 1). The results also hold in the typed case, so that confluence of the cut-elimination procedures follows.

Theorem 3

1. *The reduction relation* $\to_{\mathrm{gcut}}$ *is confluent.*
2. *The reduction relation* $\to_{\mathrm{cut}}$ *is confluent.*

Proof.

1. Suppose that $t \xrightarrow{*}_{\mathrm{gcut}} t_1$ and $t \xrightarrow{*}_{\mathrm{gcut}} t_2$. Then by Lemma 15, $\mathtt{x}(t) \xrightarrow{*}_{\beta} \mathtt{x}(t_i)$ $(i = 1, 2)$, so by confluence of $\to_{\beta}$, there is a *Beta*-term t' such that $\mathtt{x}(t_i) \xrightarrow{*}_{\beta} t'$ $(i = 1, 2)$. Since $\mathtt{x}(t_i) \xrightarrow{*}_{\mathrm{gcut}} t'$ by Theorem 2 and $t_i \xrightarrow{*}_{\mathrm{gcut}} \mathtt{x}(t_i)$ by Lemma 5, we have $t_i \xrightarrow{*}_{\mathrm{gcut}} t'$ $(i = 1, 2)$.
2. Similar, using Lemma 16 instead of Lemma 15. □

6 Conclusion

We have proved confluence of global and local cut-elimination procedures, using proof terms for a standard sequent calculus of intuitionistic logic. For the interpretation method to work, we have introduced β-reduction on *Beta*-terms, and proved its confluence by the method of parallel reduction. Then confluence of the two cut-elimination procedures has been obtained through projections onto the β-reduction. Additionally, we have established a conservativity result among the cut-elimination procedures and the β-reduction. Note that our proofs are also effective in the type-free case allowing non-terminating computations.

The problem on substitution lemmas (cf. the remark before Lemma 6) was also pointed out in [11, page 136] for the case of the classical sequent calculus. In future work, we will investigate the relation between their observations and ours, and develop proofs of confluence for some cut-elimination procedures in the classical sequent calculus.

Acknowledgements. This work was inspired by Stéphane Lengrand's conjectures on confluence of cut-elimination procedures that simulate β-reduction. The work was partially supported by the Japanese Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for Young Scientists (B) 17700003.

References

1. Bloo, R., Geuvers, H.: Explicit substitution: On the edge of strong normalization. Theoretical Computer Science 211, 375–395 (1999)
2. Danos, V., Joinet, J.-B., Schellinx, H.: A new deconstructive logic: Linear logic. The Journal of Symbolic Logic 62, 755–807 (1997)
3. Espírito Santo, J.: Revisiting the correspondence between cut elimination and normalisation. In: Welzl, E., Montanari, U., Rolim, J.D.P. (eds.) ICALP 2000. LNCS, vol. 1853, pp. 600–611. Springer, Heidelberg (2000)
4. Gentzen, G.: Untersuchungen über das logische Schliessen. Mathematische Zeitschrift, 39: pp. 176–210, pp. 405–431, English translation in [9 pp. 68–131] (1935)
5. Girard, J.-Y.: Linear logic. Theoretical Computer Science 50, 1–102 (1987)
6. Hardin, T.: Résultats de confluence pour les règles fortes de la logique combinatoire catégorique et liens avec les lambda-calculs. Thèse de doctorat, Université de Paris VII (1987)
7. Howard, W.A.: The formulae-as-types notion of construction. In: Seldin, J.P., Hindley, J.R. (eds.) To H. B. Curry: Essays on Combinatory Logic, Lambda-Calculus and Formalism, pp. 479–490. Academic Press, San Diego (1980)
8. Kikuchi, K.: On a local-step cut-elimination procedure for the intuitionistic sequent calculus. In: Hermann, M., Voronkov, A. (eds.) LPAR 2006. LNCS (LNAI), vol. 4246, pp. 120–134. Springer, Heidelberg (2006)
9. Szabo, M.E. (ed.): The Collected Papers of Gerhard Gentzen. North-Holland (1969)
10. Takahashi, M.: Parallel reductions in λ-calculus. Information and Computation 118, 120–127 (1995)
11. Urban, C., Bierman, G.M.: Strong normalisation of cut-elimination in classical logic. Fundamenta Informaticae 45, 123–155 (2001)