The call-by-value $\lambda\mu\wedge\vee$-calculus

Karim Nour, Khelifa Saber

Computational Logic and Applications, CLA '05 DMTCS proc. **AF**, 2006, 97–108

Confluency property of the call-by-value $\lambda\mu^{\wedge\vee}$-calculus

Karim Nour[†] and Khelifa Saber[‡]

LAMA - Équipe de logique , Université de Savoie , F-73376 Le Bourget du Lac, France

In this paper, we introduce the $\lambda\mu^{\wedge\vee}$- call-by-value calculus and we give a proof of the Church-Rosser property of this system. This proof is an adaptation of that of Andou (2003) which uses an extended parallel reduction method and complete development.

Keywords: Call-by-value, Church-Rosser, Propositional classical logic, Parallel reduction, Complete development

1 Introduction

Gentzen (1955) introduced the natural deduction system to study the notion of proof. The full classical natural deduction system is well adapted for the human reasoning. By full we mean that all the connectives ($\rightarrow$, $\wedge$ and $\vee$) and $\perp$ (for the absurdity) are considered as primitive. As usual, the negation is defined by $\neg A = A \rightarrow \perp$. Considering this logic from the computer science of view is interesting because, by the Curry-Howard correspondence, formulas can be seen as types for the functional programming languages and correct programs can be extracted. The corresponding calculus is an extension of M. Parigot's $\lambda\mu$-calculus with product and coproduct, which is denoted by $\lambda\mu^{\wedge\vee}$-calculus.

De Groote (2001) introduced the typed $\lambda\mu^{\wedge\vee}$-calculus to code the classical natural deduction system, and showed that it enjoys the main important properties: the strong normalization, the confluence and the subformula property. This would guarantee that proof normalization may be interpreted as an evaluation process. As far as we know the typed $\lambda\mu^{\wedge\vee}$-calculus is the first extension of the simply typed λ-calculus which enjoys all the above properties. Ritter et al. (2000a) introduced an extension of the $\lambda\mu$-calculus that features disjunction as primitive (see also Ritter et al. (2000b)). But their system is rather different since they take as primitive a classical form of disjunction that amounts to $\neg A \rightarrow B$. Nevertheless, Ritter and Pym (2001) give another extension of the $\lambda\mu$-calculus with an intuitionistic disjunction. However, the reduction rules considered are not sufficient to guarantee that the normal forms satisfy the subformula property. The question of the strong normalization of the full logic has interested several authors, thus one finds in David and Nour (2003), Matthes (2005) and Nour and Saber (2005) different proofs of this result.

From a computer science point of view, the $\lambda\mu^{\wedge\vee}$-calculus may be seen as the kernel of a typed call-by-name functional language featuring product, coproduct and control operators. However we cannot apply

[†]knour@univ-savoie.fr
[‡]ksabe@univ-savoie.fr

an arbitrary reduction for implementation of programming languages, we have to fix a reduction strategy and usually it is the call-by-value strategy. Many programming langagues and control operations were developed through the studies of the call-by-value variant like `ML` and `Lisp` for λ-calculus, the calculus of exception handling $\lambda_{exn}^{\rightarrow}$ and $\mu\mathtt{PCF}_V$ for the $\lambda\mu$-calculus. Ong and Stewart (2001) showed that $\mu\mathtt{PCF}_V$ is sufficiently strong to express the various control constructs such as the `ML-style raise` and the first-class continuations `callcc`, `throw` and `abort`. In this sense, it seems to be important to study the call-by-value version of $\lambda\mu^{\wedge\vee}$-calculus.

Among the important properties required in any abstract reduction system, there is the confluence which ensures the uniqueness of the normal form (if it exists). The notion of parallel reduction which is based on the method of Tait and Martin-Löf is a good tool to prove the confluence property for several reduction systems. The idea is very clear and intuitive: It consists in reducing a number of redexes existing in the term simultaneously. However, this method does not work for the $\lambda\mu^{\wedge\vee}$-calculus. In fact the diamond property which stipulates that: If $t \succ t'$ then $t' \succ t^*$ (where t^* is usually referred as the complete development of t) does not hold because more complicated situations appear, and that is due to the presence of the permutative reductions "$((u\,[x.v, y.w])\,\varepsilon) \triangleright (u\,[x.(v\,\varepsilon), y.(w\,\varepsilon)])$". Hence the proof of the confluence becomes hard and not at all trivial as it seems to be.

Consider the terms $t = (((u\,[x.v, y.w])\,[r.p, s.q])\,\varepsilon)$, $t_1 = ((u\,[x.(v\,[r.p, s.q]), y.(w\,[r.p, s.q])])\,\varepsilon)$, and $t_2 = ((u\,[x.v, y.w])\,[r.(p\,\varepsilon), s.(q\,\varepsilon)])$. We have: $t \succ t_1$ and $t \succ t_2$, if we want the diamond property to hold, t_1 and t_2 must be reduced to the same term t^* by one reduction step , however this is not possible. To make it possible we need another step of permutative reduction. We consider such a successive sequence of reductions as a one parallel reduction step, i.e, we follow the permutative reductions in the term step by step to a certain depth which allows to join and consider this sequence as a one reduction step. The notion of Prawitz' s *segment* yields the formulation of this new parallel reduction. Therefore the difficulties are overcome by extending this notion to our system (see Andou (1995), Andou (2003), Prawitz (1965) and Prawitz (1971)) and considering the extended structural reductions along this *segment* which allow us to define a complete development to obtain directly the common reductum, hence the Church-Rosser property. This is exactly what is done in Andou (2003); our proof is just a checking that this method is well adapted to provide the diamond property for the call-by-value $\lambda\mu^{\wedge\vee}$-calculus including the symmetrical rules. Thus $t_1 \succ t^*$ and $t_2 \succ t^*$, where $t^* = (u\,[x.(v\,[r.(p\,\varepsilon), s.(q\,\varepsilon)]), y.(w\,[r.(p\,\varepsilon), s.(q\,\varepsilon)])])$.

The paper is organized as follows. Section 2 is an introduction to the typed system, the relative cut-elimination procedure of $\lambda\mu^{\wedge\vee}$-calculus and the call-by-value $\lambda\mu^{\wedge\vee}$-calculus. In section 3, we define the parallel reduction related to the notion of *segment-tree*, thus we give the key lemma from which the diamond-property is directly deduced. Section 4 is devoted to the proof of the key lemma. We conclude with some future work.

2 Notations and definitions

Definition 2.1 *We use notations inspired by Andou (2003).*

1. *Let $\mathcal{X}$ and $\mathcal{A}$ be two disjoint alphabets for distinguishing the λ-variables and μ-variables respectively. We code deductions by using a set of terms $\mathcal{T}$ which extends the λ-terms and is given by the following grammar (which gives terms at the untyped level):*

$$\mathcal{T} ::= \mathcal{X} \mid \lambda\mathcal{X}.\mathcal{T} \mid (\mathcal{T}\ \ \mathcal{E}) \mid \langle\mathcal{T},\mathcal{T}\rangle \mid \omega_1\mathcal{T} \mid \omega_2\mathcal{T} \mid \mu\mathcal{A}.\mathcal{T} \mid (\mathcal{A}\ \ \mathcal{T})$$
$$\mathcal{E} ::= \mathcal{T} \mid \pi_1 \mid \ \pi_2 \mid [\mathcal{X}.\mathcal{T}, \mathcal{X}.\mathcal{T}]$$

An element of the set $\mathcal{E}$ is said to be an $\mathcal{E}$-term. Application between two $\mathcal{E}$-terms u and ε is denoted by $(u\ \varepsilon)$.

2. *The meaning of the new constructors is given by the typing rules below where Γ (resp. Δ) is a context, i.e. a set of declarations of the form $x : A$ (resp. $a : A$) where x is a λ-variable (resp. a is a μ-variable) and A is a formula.*

$$\frac{}{\Gamma, x : A\ \vdash x : A\ ;\ \Delta}ax$$

$$\frac{\Gamma, x : A \vdash t : B; \Delta}{\Gamma \vdash \lambda x.t : A \to B; \Delta}\to_i \qquad \frac{\Gamma \vdash u : A \to B; \Delta \quad \Gamma \vdash v : A; \Delta}{\Gamma \vdash (u\ \ v) : B; \Delta}\to_e$$

$$\frac{\Gamma \vdash u : A; \Delta \quad \Gamma \vdash v : B; \Delta}{\Gamma \vdash \langle u, v\rangle : A \wedge B; \Delta}\wedge_i$$

$$\frac{\Gamma \vdash t : A \wedge B; \Delta}{\Gamma \vdash (t\ \ \pi_1) : A; \Delta}\wedge_e^1 \quad \frac{\Gamma \vdash t : A \wedge B; \Delta}{\Gamma \vdash (t\ \ \pi_2) : B; \Delta}\wedge_e^2$$

$$\frac{\Gamma \vdash t : A; \Delta}{\Gamma \vdash \omega_1 t : A \vee B; \Delta}\vee_i^1 \quad \frac{\Gamma \vdash t : B; \Delta}{\Gamma \vdash \omega_2 t : A \vee B; \Delta}\vee_i^2$$

$$\frac{\Gamma \vdash t : A \vee B; \Delta \quad \Gamma, x : A \vdash u : C; \Delta \quad \Gamma, y : B \vdash v : C; \Delta}{\Gamma \vdash (t\ \ [x.u, y.v]) : C; \Delta}\vee_e$$

$$\frac{\Gamma \vdash t : A; \Delta, a : A}{\Gamma \vdash (a\ \ t) : \bot; \Delta, a : A}\bot_i \quad \frac{\Gamma \vdash t : \bot; \Delta, a : A}{\Gamma \vdash \mu a.t : A; \Delta}\bot_e$$

3. *A term in the form $(t\ [x.u, y.v])$ (resp $\mu a.t$) is called an $\vee_e$-term (resp $\bot_e$-term).*

4. *The cut-elimination procedure corresponds to the reduction rules given below. They are those we need to the subformula property.*

   - $(\lambda x.u\ \ v) \rhd_\beta u[x := v]$
   - $(\langle t_1, t_2\rangle\ \ \pi_i) \rhd_\pi t_i$
   - $(\omega_i t\ \ [x_1.u_1, x_2.u_2]) \rhd_D u_i[x_i := t]$
   - $((t\ \ [x_1.u_1, x_2.u_2])\ \ \varepsilon) \rhd_\delta (t\ \ [x_1.(u_1\ \varepsilon), x_2.(u_2\ \varepsilon)])$
   - $(\mu a.t\ \ \varepsilon) \rhd_\mu \mu a.t[a :=^* \varepsilon]$
     where $t[a :=^* \varepsilon]$ is obtained from t by replacing inductively each subterm in the form $(a\ v)$ by $(a\ (v\ \varepsilon))$.

5. *Let t and t' be terms. The notation $t \rhd t'$ means that t reduces to t' by using one step of the reduction rules given above. Similarly, $t \rhd^* t'$ means that t reduces to t' by using some steps of the reduction rules given above.*

The following result is straightforward

Theorem 2.1 *(Subject reduction) If $\Gamma \vdash t : A; \Delta$ and $t \triangleright^* t'$, then $\Gamma \vdash t' : A; \Delta$.*

We have also the following properties (see Andou (2003), David and Nour (2003), De Groote (2001), Matthes (2005), Nour and Saber (2005) and Nour and Saber (2006)).

Theorem 2.2 *(Confluence) If $t \triangleright^* t_1$ and $t \triangleright^* t_2$, then there exists t_3 such that $t_1 \triangleright^* t_3$ and $t_2 \triangleright^* t_3$.*

Theorem 2.3 *(Strong normalization) If $\Gamma \vdash t : A; \Delta$, then t is strongly normalizable.*

Remark 2.1 *Following the call-by-value evaluation discipline, in an application the evaluator has to diverge if the argument diverges. For example, in the call-by-value λ-calculus, we are allowed to reduce the β-redex $(\lambda x.u\ v)$ only when v is a value. In $\lambda\mu$-calculus, the terms $\mu a.u$ and $(u\ [x_1.u_1, x_2.u_2])$ cannot be taken as values, then the terms $(\lambda x.t\ \mu a.u)$ and $(\lambda x.t\ (u\ [x_1.u_1, x_2.u_2]))$ cannot be reduced. This will be able to prevent us from reaching many normal forms. To solve this problem, we introduce symmetrical rules (δ'_v and μ'_v) allowing to reduce these kinds of redexes.*

Now we introduce the call-by-value version of the $\lambda\mu^{\wedge\vee}$-calculus. From a logical point of view a value corresponds to an introduction of a connective; this is the reason why the Parigot's naming rule is considered as the introduction rule of $\perp$.

Definition 2.2 *1. The set of values $\mathcal{V}$ is given by the following grammar:*

$$\mathcal{V} \ := \ \mathcal{X} \mid \lambda\mathcal{X}.\mathcal{T} \mid \langle \mathcal{V}, \mathcal{V} \rangle \mid \omega_1\mathcal{V} \mid \omega_2\mathcal{V} \mid (\mathcal{A}\ \ \mathcal{T})$$

Values are denoted $U, V, W, \ldots$

2. *The reduction rules of the call-by-value $\lambda\mu^{\wedge\vee}$-calculus are the followings:*

   - $(\lambda x.t\ \ V) \triangleright_{\beta_v} t[x := V]$
   - $(\langle V_1, V_2 \rangle\ \ \pi_i) \triangleright_{\pi_v} V_i$
   - $(\omega_i V\ \ [x_1.t_1, x_2.t_2]) \triangleright_{D_v} t_i[x_i := V]$
   - $((t\ \ [x_1.t_1, x_2.t_2])\ \ \varepsilon) \triangleright_{\delta} (t\ \ [x_1.(t_1\ \varepsilon), x_2.(t_2\ \varepsilon)])$
   - $(V\ (t\ \ [x_1.t_1, x_2.t_2])) \triangleright_{\delta'_v} (t\ \ [x_1.(V\ t_1), x_2.(V\ t_2)])$
   - $(\mu a.t\ \ \varepsilon) \triangleright_{\mu} \mu a.t[a :=^* \varepsilon]$
   - $(V\ \ \mu a.t) \triangleright_{\mu'_v} \mu a.t[a :=_* V]$
     where $t[a :=_* V]$ is obtained from t by replacing inductively each subterm in t in the form $(a\ u)$ by $(a\ (V\ u))$.

   The first three rules are called logical rules and the others are called structural rules.

3. *The one-step reduction $\triangleright_v$ of the call-by-value $\lambda\mu^{\wedge\vee}$-calculus is defined as the union of the seven rules given above. As usual $\triangleright_v^*$ denotes the transitive and reflexive closure of $\triangleright_v$.*

The following lemma expresses the fact that the set of values is closed under reductions. In the remainder of this paper, this fact will be used implicitly.

Lemma 2.1 *If V is a value and $V \rhd_v^* W$, then W is a value.*

Proof: ¿From the definition of the set of values. □

Theorem 2.4 *(Subject reduction) If $\Gamma \vdash t : A \,;\triangle$ and $t \rhd_v^* t'$, then $\Gamma \vdash t' : A \,;\triangle$.*

Proof: Since the reduction rules correspond to the cut-elimination procedure, we check easily that the type is preserved from the redex to its reductom. □

The rest of this paper is an extention of Andou (2003) to our calculus according to the new considered reduction rules δ'_v and μ'_v. One can find all the notions given here in Andou (2003). Since the new symmetrical rules that we add don't create any critical pair with the existing rules, then in the examples and proofs that we give, one will mention only the cases related to these new rules and check that they don't affect the core of Andou (2003)'s work.

3 The extended structural reduction

Definition 3.1 *1. Let t be a term, we define a binary relation denoted by $\sqsupset_t$ on subterms of t as follows:*

- $(u\ \ [x_1.u_1, x_2.u_2]) \sqsupset_t u_i$
- *$\mu a.u \sqsupset_t v$, where v occurs in u in the form $(a\ v)$*

If $u \sqsupset_t v$ holds, then v is called a segment-successor of u, and u is called a segment-predecessor of v. We denote by $\sqsupseteq_t$ the reflexive and transitive closure of $\sqsupset_t$.

2. *Let r be a subterm of a term t, such that r is a $\vee_e$- or $\perp_e$-term and r has no segment-predecessor in t. A segment-tree from r in t is a set $\mathcal{O}$ of subterms of t, such that for each $w \in \mathcal{O}$:*
   - $r \sqsupseteq_t w$
   - *w is a $\vee_e$- or $\perp_e$-term*
   - *For each subterm s of t, such that $r \sqsupseteq_t s \sqsupseteq_t w$ then $s \in \mathcal{O}$*

   r is called the root of $\mathcal{O}$.

3. *Let $\mathcal{O}$ be a segment-tree from r in t, a subterm v of t is called an acceptor of $\mathcal{O}$ iff v is a segment-successor of an element of $\mathcal{O}$ and v is not in $\mathcal{O}$.*

4. *A segment-tree $\mathcal{O}$ from r in t is called the maximal segment-tree iff no acceptor of $\mathcal{O}$ has a segment successor in t.*

5. *The acceptors of $\mathcal{O}$ are indexed by the letter $\mathcal{O}$.*

6. *Let $\mathcal{O}$ be a segment-tree from t in t itself, and $t \rhd_v^* t'$, then we define canonically a corresponding segment-tree to $\mathcal{O}$ in t' by the transformation of indexes from redexes to their residuals. This new segment-tree is denoted also by $\mathcal{O}$ if there is no ambiguity.*

Remark 3.1 *For typed terms, all the elements of a segment-tree have the same type.*

Definition 3.2 *Let $\mathcal{O}$ be a segment-tree from r in t, suppose that r occurs in t in the form $(V\,r)$ (resp $(r\,\varepsilon)$). The extended structural reduction of t along $\mathcal{O}$ is the transformation to a term t' obtained from t by replacing each indexed term $v_{\mathcal{O}}$ (the acceptors of $\mathcal{O}$) by $(V\,v)$ (resp $(v\,\varepsilon)$) and erasing the occurence of V (resp ε) in $(V\,r)$ (resp $(r\,\varepsilon)$) . This reduction is denoted by $t \succ_{\mathcal{O}} t'$.*

Remark 3.2 *By the definition above, every structural reduction is an extended structural reduction. It corresponds to the particular case where the segment-tree consists only of its root.*

Example 3.1 *Here are two examples of segment-trees and the extended structural reduction. Let $t = (u\;[x.\mu a.(a\,\langle x,(a\;w)\rangle), y.v])$ and V a value.*

1. *The set $\mathcal{O}_1 = \{t\}$ is a segment-tree from t in t itself. The acceptors of $\mathcal{O}_1$ are $\mu a.(a\,\langle x,(a\;w)\rangle)$ and v. Then t is represented as follows:*

   $t = (u\;[x.(\mu a.(a\,\langle x,(a\;w)\rangle))_{\mathcal{O}_1}, y.v_{\mathcal{O}_1}])$,
   and $(V\;t) \succ_{\mathcal{O}_1} (u\;[x.(V\;\mu a.(a\,\langle x,(a\;w)\rangle)), y.(V\;v)])$.

2. *The set $\mathcal{O}_2 = \{t, \mu a.(a\,\langle x,(a\;w)\rangle)\}$ is also a segment-tree from t in t. The acceptors of $\mathcal{O}_2$ are $\langle x,(a\;w)\rangle$, w and v. Then t is represented as follows:*

   $t = (u\;[x.\mu a.(a\,\langle x,(a\;w_{\mathcal{O}_2})\rangle_{\mathcal{O}_2}), y.v_{\mathcal{O}_2}])$,
   and $(V\;t) \succ_{\mathcal{O}_2} (u\;[x.\mu a.(a\;(V\;\langle x,(a\;(V\;w))\rangle)), y.(V\;v)])$.

Definition 3.3 *The parallel reduction $\succ$ is defined inductively by the following rules:*

- $x \succ x$
- *If $t \succ t'$, then $\lambda x.t \succ \lambda x.t'$, $\mu a.t \succ \mu a.t'$, $(a\;t) \succ (a\;t')$ and $\omega_i t \succ \omega_i t'$*
- *If $t \succ t'$ and $u \succ u'$, then $\langle t,u\rangle \succ \langle t',u'\rangle$*
- *If $t \succ t'$ and $\varepsilon \tilde{\succ} \varepsilon'$, then $(t\;\varepsilon) \succ (t'\;\varepsilon')$*
- *If $t \succ t'$ and $V \succ V'$, then $(\lambda x.t\;V) \succ t'[x := V']$*
- *If $V_i \succ V_i'$, then $(\langle V_1, V_2\rangle\;\pi_i) \succ V_i'$*
- *If $V \succ V'$ and $u_i \succ u_i'$, then $(\omega_i V\;[x_1,u_1,x_2,u_2]) \succ u_i'[x_i := V']$*
- *If $t \succ t'$, $V \succ V'$ (resp $\varepsilon \tilde{\succ} \varepsilon'$), and $\mathcal{O}$ is a segment-tree from t in t, and $(V'\,t') \succ_{\mathcal{O}} w$ (resp $(t'\,\varepsilon') \succ_{\mathcal{O}} w$), then $(V\,t) \succ w$ (resp $(t\,\varepsilon) \succ w$), where $\varepsilon \tilde{\succ} \varepsilon'$ means that:*
  - *$\varepsilon = \varepsilon' = \pi_i$, or*
  - *($\varepsilon = u$ and $\varepsilon' = u'$) or ($\varepsilon = [x.u, y.v]$ and $\varepsilon' = [x.u', y.v']$) such that $u \succ u'$ and $v \succ v'$.*

It is easy to see that $\triangleright_v^*$ is the transitive closure of $\succ$.

Definition 3.4 *Let t be a term , we define the complete development t^* as follows:*

- $x^* = x$

- $(\lambda x.t)^* = \lambda x.t^*$
- $(\mu a.t)^* = \mu a.t^*$
- $\langle t_1, t_2\rangle^* = \langle t_1^*, t_2^*\rangle$
- $(\omega_i t)^* = \omega_i t^*$
- $(a\, t)^* = (a\, t^*)$,
- $(t\ \varepsilon)^* = (t^*\ \varepsilon^*)$, *if* $(t\, \varepsilon)$ *is not a redex*
- $(\lambda x.t\ V)^* = t^*[x := V^*]$
- $(\langle V_1, V_2\rangle\ \pi_i)^* = V_i^*$
- $(\omega_i V\ [x_1.u_1, x_2.u_2])^* = u_i^*[x := V^*]$
- *Let* $\mathcal{O}_m$ *be the maximal segment-tree from* t *in* t, *and* $(V^*\ t^*) \succ_{\mathcal{O}_m} w$ *(resp* $(t^*\ \tilde{\varepsilon^*}) \succ_{\mathcal{O}_m} w$*), then* $(V\ t)^* = w$ *(resp* $(t\ \varepsilon)^* = w$*), where* $\tilde{\varepsilon^*}$ *means:*
  - ε, *if* $\varepsilon = \pi_i$
  - u^*, *if* $\varepsilon = u$
  - $[x.u^*, y.v^*]$, *if* $\varepsilon = [x.u, y.v]$

Lemma 3.1 *1. If* $t \succ t'$ *and* $V \succ V'$, *then* $t[x := V] \succ t'[x := V']$.

2. If $t \succ t'$ *and* $\varepsilon \succ \varepsilon'$, *then* $t[a :=^* \varepsilon] \succ t'[a :=^* \varepsilon']$.

3. If $t \succ t'$ *and* $V \succ V'$, *then* $t[a :=_* V] \succ t'[a :=_* V']$.

Proof: By a straightforward induction on the structure of $t \succ t'$. □

Lemma 3.2 *(The key lemma) If* $t \succ t'$, *then* $t' \succ t^*$.

Proof: The proof of this lemma will be the subject of the next section. □

Theorem 3.1 *(The Diamond Property) If* $t \succ t_1$ *and* $t \succ t_2$, *then there exists* t_3 *such that* $t_1 \succ t_3$ *and* $t_2 \succ t_3$.

Proof: It is enough to take $t_3 = t^*$, then theorem holds by the key lemma. □

Since $\triangleright_v^*$ is identical to the transitive closure of $\succ$, we have the confluence of the call-by-value $\lambda\mu^{\wedge\vee}$-calculus.

Theorem 3.2 *If* $t \triangleright_v^* t_1$ *and* $t \triangleright_v^* t_2$, *then there exists a term* t_3 *such that* $t_1 \triangleright_v^* t_3$ *and* $t_2 \triangleright_v^* t_3$.

4 Proof of the key lemma

For technical reasons (see the example below), we start this section by extending the notion of the segment-tree.

Definition 4.1 *1. Let v be a subterm in a term t, v is called a bud in t iff v is t itself or v occurs in t in the form $(a\ v)$ where a is a free variable in t.*

2. *Let $\mathcal{O}_1$,...,$\mathcal{O}_n$ be segment-trees from respectively $r_1, ..., r_n$ in a term t, and $\mathcal{P}$ a set of buds (possibly empty) in t. Then a segment-wood is a pair $\langle \mathcal{O}_1 \cup ... \cup \mathcal{O}_n, \mathcal{P}\rangle$ such that:*

   - *r_i is a bud in t for each i,*
   - *$\mathcal{O}_1, ..., \mathcal{O}_n$ and $\mathcal{P}$ are mutually disjoints.*

3. *Let $\mathcal{Q} = \langle \mathcal{O}_1 \cup ... \cup \mathcal{O}_n, \mathcal{P}\rangle$ be a segment-wood in t, the elements of $\mathcal{O}_1 \cup ... \cup \mathcal{O}_n$ are called trunk-pieces of $\mathcal{Q}$, and those of $\mathcal{P}$ are called proper-buds of $\mathcal{Q}$.*

   (a) *We denote by $Bud(\mathcal{Q})$ the set of buds $\mathcal{P} \cup \{r_1, ..., r_n\}$ in t.*

   (b) *An acceptor of a segment-wood $\mathcal{Q}$ is either an acceptor of $\mathcal{O}_i$ for some i, either a proper-bud.*

   (c) *The acceptors of $\mathcal{Q}$ are indexed by $\mathcal{Q}$.*

   (d) *If the root r of a segment-tree $\mathcal{O}$ in t is a bud in t, then we identify $\mathcal{O}$ with the segment-wood $\langle \mathcal{O}, \emptyset\rangle$.*

4. *Let $\mathcal{Q}$ be a segment-wood in t, and s a subterm in t. The restriction of indexed subterms by $\mathcal{Q}$ to s constrcuts a segment-wood in s, which we will denote also by $\mathcal{Q}$ if there is no ambiguity.*

Remark 4.1 *1. If v is a bud in t, then v has no segment-predecessor in t. Therefore any segment-successor is not a bud.*

2. *Let $\mathcal{Q} = \langle \mathcal{O}_1 \cup ... \cup \mathcal{O}_n, \mathcal{P}\rangle$ be a segment-wood, since a segment-successor is not a bud, then any acceptor of any $\mathcal{O}_i$ is not in $Bud(\mathcal{Q})$.*

3. *The two conditions in (2) of the above definition are equivalent to the fact that all the elements of $\mathcal{P}$ and the buds $r_1, ..., r_n$ are distincts.*

4. *If $\mathcal{O}$ is a segment-tree from t in t, and s is a subterm in t, then the restriction of $\mathcal{O}$ to s constructs a segment-wood in s.*

5. *Proper-buds and trunk-pieces cannot be treated in a uniform way, since in a term, what will be indexed are the proper-buds themselves and the acceptors of the trunk-pieces, thing which is allowed by a formulation which makes difference between these two notions.*

Definition 4.2 *Let t, ε be $\mathcal{E}$-terms, V a value and $\mathcal{Q}$ a segment-wood in t, we define the term $t[V/\mathcal{Q}]$ (resp $t[\varepsilon/\mathcal{Q}]$) which is obtained from t by replacing each indexed term $v_{\mathcal{Q}}$ (the acceptors of $\mathcal{Q}$) in t by $(V\ v)$ (resp $(v\, \varepsilon)$).*

Remark 4.2 *It's clear that if $(V\ t) \succ_{\mathcal{O}} w$ (resp $(t\, \varepsilon) \succ_{\mathcal{O}} w$), then $w = t[V/\mathcal{O}]$ (resp $w = t[\varepsilon/\mathcal{O}]$).*

Example 4.1 *Let $t = \mu a.(a\ \mu b.(b\ \omega_2\lambda s.(a\ \omega_1 s)))$ be a term and r the subterm $\mu b.(b\ \omega_2\lambda s.(a\ \omega_1 s))$ in t. We define two segment-trees from t in t, $\mathcal{O}_1 = \{t\}$ and $\mathcal{O}_2 = \{t, r\}$, observe that the acceptors of $\mathcal{O}_1$ are r and $\omega_1 s$, however those of $\mathcal{O}_2$ are $\omega_2\lambda s.(a\ \omega_1 s)$ and $\omega_1 s$. The restriction $\mathcal{Q}_1$ (resp $\mathcal{Q}_2$) of $\mathcal{O}_1$ (resp $\mathcal{O}_2$) to r is the following segment-wood: $\mathcal{Q}_1 = \langle\emptyset, \{r, \omega_1 s\}\rangle$ (resp $\mathcal{Q}_2 = \langle\{r\}, \{\omega_1 s\}\rangle$). Remark also that $Bud(\mathcal{Q}_1) = Bud(\mathcal{Q}_2)$ and the set of trunk-pieces of $\mathcal{Q}_1$ is a subset of that of $\mathcal{Q}_2$. Suppose that V is a value then:*

- $t[V/\mathcal{Q}_1] = \mu a.(a\ (V\ \mu b.(b\ \omega_2\lambda s.(a\ (V\ \omega_1 s)))))$.
- $t[V/\mathcal{Q}_2] = \mu a.(a\ \mu b.(b\ (V\ \omega_2\lambda s.(a\ (V\ \omega_1 s)))))$.
- $t[V/\mathcal{Q}_1] \succ t[V/\mathcal{Q}_2]$

Lemma 4.1 *Let $\mathcal{Q}_1$ and $\mathcal{Q}_2$ be two segment-woods in a term t such that: $Bud(\mathcal{Q}_1) = Bud(\mathcal{Q}_2)$ and the set of all trunke-pieces of $\mathcal{Q}_1$ is a subset of that of $\mathcal{Q}_2$. Suppose also that $t \succ t'$ and $V \succ V'$ (resp $\varepsilon\tilde{\succ}\varepsilon'$), then $t[V/\mathcal{Q}_1] \succ t'[V'/\mathcal{Q}_2]$ (resp $t[\varepsilon/\mathcal{Q}_1] \succ t'[\varepsilon'/\mathcal{Q}_2]$).*

Proof: By induction on t. We look at the last rule used for $t \succ t'$. We examine only one case. The others are either treated similarly, either by a straightforward induction.

$t = (W\ u)$ and $t' = u'[W'/\mathcal{O}]$, where $\mathcal{O}$ is a segment-tree from u in u, $u \succ u'$ and $W \succ W'$.

- If t is not an acceptor of $\mathcal{Q}_1$ and then nor of $\mathcal{Q}_2$: By the induction hypothesis, $u[V/\mathcal{Q}_1] \succ u'[V'/\mathcal{Q}_2]$ and $W[V/\mathcal{Q}_1] \succ W'[V'/\mathcal{Q}_2]$. Since $\mathcal{O}$ is a segment-tree from u in u, we have: $t[V/\mathcal{Q}_1] = (W[V/\mathcal{Q}_1]\ u[V/\mathcal{Q}_1]) \succ u'[V'/\mathcal{Q}_2][W'[V'/\mathcal{Q}_2]/\mathcal{O}] = u'[W'/\mathcal{O}][V'/\mathcal{Q}_2] = t'[V'/\mathcal{Q}_2]$.
- If t is an acceptor of $\mathcal{Q}_1$ but not of $\mathcal{Q}_2$: Let $\mathcal{Q}_2 = \langle\mathcal{O}_t \cup \mathcal{O}_{r_1} \cup ... \cup \mathcal{O}_{r_n}, \mathcal{P}\rangle$ and $\mathcal{Q}_2^- = \langle\mathcal{O}_{r_1} \cup ... \cup \mathcal{O}_{r_n}, \mathcal{P}\rangle$, where $\mathcal{O}_s$ denotes a segment-tree from the bud s in t. By the induction hypothesis, $u[V/\mathcal{Q}_1] \succ u'[V'/\mathcal{Q}_2^-]$ and $W[V/\mathcal{Q}_1] \succ W'[V'/\mathcal{Q}_2^-]$. Moreover $(W'[V'/\mathcal{Q}_2^-]\ u'[V'/\mathcal{Q}_2^-]) \succ_{\mathcal{O}} u'[V'/\mathcal{Q}_2^-][W'[V'/\mathcal{Q}_2^-]/\mathcal{O}]$. Hence $(W[V/\mathcal{Q}_1]\ u[V/\mathcal{Q}_1]) \succ u'[V'/\mathcal{Q}_2^-][W'[V'/\mathcal{Q}_2^-]/\mathcal{O}]$. Therefore, $t[V/\mathcal{Q}_1] = (V\ (W[V/\mathcal{Q}_1]\ u[V/\mathcal{Q}_1])) \succ u'[V'/\mathcal{Q}_2][W'[V'/\mathcal{Q}_2^-]/\mathcal{O}][V'/\mathcal{O}_t] = u'[W'/\mathcal{O}][V'/\mathcal{Q}_2^-][V'/\mathcal{O}_t] = u'[W'/\mathcal{O}][V'/\mathcal{Q}_2] = t'[V'/\mathcal{Q}_2]$.
- If t is an acceptor of $\mathcal{Q}_1$ and $\mathcal{Q}_2$, then $t[V/\mathcal{Q}_1] = (V\ (W[V/\mathcal{Q}_1]\ u[V/\mathcal{Q}_1])) \succ (V'\ u'[V'/\mathcal{Q}_2][W'[V'/\mathcal{Q}_2]/\mathcal{O}]) = (V'\ u'[W'/\mathcal{O}][V'/\mathcal{Q}_2]) = t'[V'/\mathcal{Q}_2]$.

□

Proof of the key lemma:

By induction on t. We look at the last rule used in $t \succ t'$. Only one case is mentioned: $t = (V\ u)$ and $t' = u'[V'/\mathcal{O}]$ where $\mathcal{O}$ is a segment-tree from u in u, $u \succ u'$ and $V \succ V'$. In this case $t^* = u^*[V^*/\mathcal{O}_m]$, where $\mathcal{O}_m$ is the maximal segment-tree from u in u. Therefore, by the previous lemma (it' s clear that $\mathcal{O}$ and $\mathcal{O}_m$ as segment-woods satisfy the hypothesis of this lemma 4.1) and the induction hypothesis, $u'[V'/\mathcal{O}] \succ u^*[V^*/\mathcal{O}_m]$.

□

5 Future work

The strong normalization of this system cannot be directly deduced from that of $\lambda\mu^{\wedge\vee}$-calculus, since we consider the symmetric structural reductions μ'_v and δ'_v. Even if the strong normalization of $\lambda\mu\mu'$-calculus is well known (see David and Nour (2005a)), the presence of μ'_v and δ'_v complicates the management of the duplication and the creation of redexes when the other reductions are considered.

Acknowledgements

We wish to thank P. De Groote for helpful discussions.

References

Y. Andou. *Church-Rosser* of simple reduction for full first-order classical natural deduction. *Annals of Pure and Applied logic*, 119:225–237, 2003.

Y. Andou. A normalization-procedure for the first order classical natural deduction with full logical symbols. *Tsukuba J. Math.*, 19:153–162, 1995.

R. David and K. Nour. A short proof of the strong normalization of classical natural deduction with disjunction. *Journal of symbolic Logic*, 68(4):1277–1288, 2003.

R. David and K. Nour. Arithmetical proofs of the strong normalization results for the symmetric $\lambda\mu$-calculus. In *TLCA'05*, pages 162–178, 2005a.

R. David and K. Nour. Why the usual candidates of reducibility do not work for the symmetric $\lambda\mu$-calculus. *Electronic Notes in Theoretical Computer Science*, 140:101–111, 2005b.

P. De Groote. On the strong normalization of natural deduction with permutation-conversions. In *RTA'99*, pages 45–59, 2005.

P. De Groote. Strong normalization of classical natural deduction with disjunction. In *TLCA'01*, pages 182–196, 2001.

G. Gentzen. *Recherches sur la déduction logique*. Press Universitaires de France, 1955.

R. Matthes. Non-strictly positive fixed points for classical natural deduction. *APAL*, 133:205–230, 2005.

K. Nakazawa. Confluence and strong normalizability of call-by-value $\lambda\mu$-calculus. *Theoretical Computer Science*, 290:429–463, 2003.

K. Nour and K. Saber. A semantical proof of strong normalization theorem for full propositional classical natural deduction. *Archive of Mathematical Logic*, 45:357–364, 2005.

K. Nour and K. Saber. Some properties of full propositional classical natural deduction. *Manuscript*, 2006.

C.-H. L. Ong and C. A. Stewart. *A Curry-Howard* foundation for functional computation with control. In *POPL'97*, pages 215–227, 2001.

M. Parigot. $\lambda\mu$-calculus: An algorithm interpretation of classical natural deduction. *Lecture Notes in Artificial Intelligence*, 624:190–201, 1992.

D. Prawitz. *Natural deduction- A Proof Theoretical Study*. Almqvist & Wiksell, 1965.

D. Prawitz. Idea and result in proof theory. In *2nd Scandinavian Logic Symp.*, pages 235–307, 1971.

W. Py. *Confluence en $\lambda\mu$-calcul*. PhD thesis, University of Savoie, 1998.

E. Ritter and D. Pym. On the semantic of classical disjunction. *Journal of Pure and Applied Algebra*, 159:315–338, 2001.

E. Ritter, D. Pym, and L. Wallen. On the intuitionistic force of classical search. *Theoretical Computer Science*, 232:299–333, 2000a.

E. Ritter, D. Pym, and L. Wallen. Proof-terms for classical and intuitionistic resolution. *Journal of Logic and Computation*, 10(2):173–207, 2000b.