The permutative λ-calculus

Beniamino Accattoli[1] and Delia Kesner[2]

[1] INRIA and LIX (École Polytechnique)
[2] PPS (CNRS and Université Paris-Diderot)

Abstract. We introduce the permutative λ-calculus, an extension of λ-calculus with three equations and one reduction rule for permuting constructors, generalising many calculi in the literature, in particular Regnier's sigma-equivalence and Moggi's assoc-equivalence. We prove confluence modulo the equations and preservation of beta-strong normalisation (PSN) by means of an auxiliary substitution calculus. The proof of confluence relies on M-developments, a new notion of development for λ-terms.

1 Introduction

Background. The standard operational semantics of λ-calculus is given by β-reduction. However, this unique notion of reduction is often extended with some other rewriting rules allowing to permute constructors. This arises in different contexts and comes with many different motivations. A typical example is the postponement of *erasing* steps, which is obtained by introducing one particular such permutation rule [5]. Four other notable motivations for introducing permutations are: making redexes more visible [10], analysing the relation between λ-terms and Proof-Nets [17], proving the completeness of CPS-translation for the call-by-value λ-calculus [18], translating Moggi's monadic metalanguage into λ-calculus [21]. The rewriting theory of these permutation rules is often tricky, in particular when proving strong normalisation or preservation of strong normalisation (PSN) [12, 4, 14, 6, 7]. This is indeed the major and usually difficult question arising in all these extensions: to prove that if t is a β-strongly-normalising λ-term then t is also strongly-normalising with respect to the extended reduction relation.

The permutative λ-calculus. The permutative λ-calculus $\Lambda_{\mathtt{p}}$ introduced in this paper extends λ-calculus with three *equations* and one rewriting rule for permuting constructors. It sensibly generalises all previous extended λ-calculi by taking — when possible — the permutations as *equivalences*, and not as *reductions*. This is a key point of our approach.

We show that the permutative λ-calculus preserves β-strong normalisation and is Church-Rosser modulo the equivalences, the strongest possible form of confluence for a reduction relation modulo an equivalence. Whenever an orientation of the equations (or a subset of them) yields a terminating reduction $\rightsquigarrow$ then the system where the equations are replaced by $\rightsquigarrow$ enjoys PSN. Thus, our result subsumes all PSN results of the kind in the literature.

The proof technique. We study the permutative λ-calculus through an auxiliary and new calculus with **explicit substitutions (ES)** called $\lambda\mathtt{sub}$. In this calculus β-reduction is split into two subsystems: $\rightarrow_{\mathtt{dB}}$ which creates a substituted term $t[x/u]$, *i.e.* a term t affected by a *delayed/explicit* substitution $[x/u]$, and $\rightarrow_{\mathtt{sub}}$ which executes the ES $[x/u]$ — getting $t\{x/u\}$ — and hence completes β-reduction. This simple calculus is then enriched with various equivalences — thus getting the equational $\lambda\mathtt{sub}$-calculus — obtained by what might be called an *extension by continuity*: if t and u are equivalent λ-terms in $\Lambda_{\hat{\mathtt{p}}}$ and they $\rightarrow_{\mathtt{dB}}$-reduce to t' and u', respectively, then t' and u' are equivalent in the equational $\lambda\mathtt{sub}$. This requires to consider equivalences on terms with ES and not only on λ-terms.

PSN. We prove PSN for the permutative λ-calculus by reducing this problem to PSN for the *equational* $\lambda\mathtt{sub}$-calculus, which in turn reduces to an existing result for the structural λ-calculus [2].

Confluence. Confluence of the permutative λ-calculus turns out to be delicate, and our proof is one of the main contributions of the paper. Indeed, confluence of $\Lambda_{\hat{\mathtt{p}}}$ does not follow from confluence of λ-calculus. The usual Tait–Martin Löf technique does not work, since the equations may create/hide redexes. While confluence of many reduction systems can usually be proved by means of developments [9], this notion does not suffice in the case of $\Lambda_{\hat{\mathtt{p}}}$, again because the equations create redexes. Its stronger variant, known as superdevelopments [13] or L-developments [1] — which also reduces some created redexes — does not work either. We then introduce a new form of development called M-development, show its good properties with respect to $\Lambda_{\hat{\mathtt{p}}}$ and then derive confluence for $\Lambda_{\hat{\mathtt{p}}}$. A key point is that M-developments are defined and studied through the equational $\lambda\mathtt{sub}$-calculus, where the splitting of β-reduction in terms of dB and sub becomes crucial to allow a fine analysis of redex creation. A nice fact is that our proof technique is modular, in the sense that one can choose to arbitrarily orient all or only some of the equations as rewriting rules while keeping the proof essentially unchanged. Moreover, our proof does not rely on confluence of λ-calculus.

Proof-Nets. Our work is the final product of a long-term study of the relation between ES and Linear Logic Proof-Nets. Here we present the implications of our study on λ-calculus, a language without ES, which is of a more general interest. No knowledge of Proof-Nets is assumed in this paper. However, in Sec. 4 the reader accustomed with Proof-Nets will find hints to the graphical intuitions for the main concepts. In particular, the equations of $\Lambda_{\hat{\mathtt{P}}}$ have a natural justification in terms of Proof-Nets.

Roadmap. Sec. 2 introduces the permutative λ-calculus. Sec. 3 explains the difficulties to prove confluence using the notion of development. Sec. 4 introduces the equational $\lambda\mathtt{sub}$-calculus and defines $\mathtt{M}$-developments. Sec. 5 proves Church-Rosser of the reduction relation $\to_\beta$ modulo the equations and Sec. 6 extends the result to the whole calculus. Sec. 7 proves PSN and Sec. 8 concludes the paper.

2 The permutative λ-calculus

The permutative λ-calculus $\Lambda_{\hat{\mathtt{P}}}$ is given by the set of λ-terms, written Λ, and a set of equations and reduction rules. As usual [3], the term x is called a **variable**, $\lambda x.t$ an **abstraction** and $t\ u$ an **application**. **Free** and **bound** variables of λ-terms are defined as usual and respectively written $\mathtt{fv}(t)$ and $\mathtt{bv}(t)$. The equivalence relation generated by the renaming of bound variables, written $\equiv_\alpha$ or simply $=$, is called **α-conversion**. The meta-level **substitution** operation is given, as usual, on α-equivalence classes; the notation $t\{x/u\}$ means that all the free occurrences of the variable x in the term t are substituted by u by avoiding capture of free variables. **Contexts** are defined as usual an denoted by C, we write $C[\![t]\!]$ for the context C where its unique hole has been replaced by the term t.

The rewriting rules and equations of the **permutative λ-calculus** $\Lambda_{\hat{\mathtt{P}}}$ are given in Fig. 1. The two equations $\hat{\sigma}_1$ and $\hat{\sigma}_2$ are exactly Regnier's σ-equivalence [17]. The equation $\widehat{\mathtt{box}}$ and the rule $\hat{\mathtt{u}}$, called **box** and **void unboxing** respectively, are instances of a more general equation called $\mathtt{badbox}$ obtained from $\widehat{\mathtt{box}}$ by removing the side condition "$x \in \mathtt{fv}(v)$". The equation $\mathtt{badbox}$ does not belong to $\Lambda_{\hat{\mathtt{P}}}$ because it is unsound: it breaks PSN as shown in Sec. 6. In order to simplify the presentation of our results we first treat the equations of $\Lambda_{\hat{\mathtt{P}}}$ (*i.e.* $\hat{\sigma}_1$, $\hat{\sigma}_2$ and $\widehat{\mathtt{box}}$), and consider the void unboxing rule $\mapsto_{\hat{\mathtt{u}}}$ only later, in Sec. 6. The *assoc* rule of [16, 14, 21] is a particular case of $\widehat{\mathtt{box}} \cup \hat{\mathtt{u}}$.

A **β-redex** is any term of the form $(\lambda x.t)\ u$. We define $\hat{\mathtt{P}}$ as the set of equations $\{\hat{\sigma}_1, \hat{\sigma}_2, \widehat{\mathtt{box}}\}$. The **reduction relation** $\to_\beta$ (resp. $\to_{\hat{\mathtt{u}}}$) is generated by the contextual closure of the rewriting rule $\mapsto_\beta$ (resp. $\mapsto_{\hat{\mathtt{u}}}$).

$$
\begin{array}{lll}
(\lambda x.t)\ u & \mapsto_{\beta} & t\{x/u\} \\
t\ ((\lambda x.v)\ u) & \mapsto_{\hat{\mathtt{u}}} & (\lambda x.t\ v)\ u \qquad \text{if } x \notin \mathtt{fv}(t)\ \&\ x \notin \mathtt{fv}(v) \\
(\lambda x.\lambda y.t)\ u & \sim_{\hat{\sigma}_1} & \lambda y.((\lambda x.t)\ u) \qquad \text{if } y \notin \mathtt{fv}(u) \\
(\lambda x.t\ v)\ u & \sim_{\hat{\sigma}_2} & (\lambda x.t)\ u\ v \qquad \text{if } x \notin \mathtt{fv}(v) \\
(\lambda x.t\ v)\ u & \sim_{\widehat{\mathtt{box}}} & t\ ((\lambda x.v)\ u) \qquad \text{if } x \notin \mathtt{fv}(t)\ \&\ x \in \mathtt{fv}(v)
\end{array}
$$

Fig. 1: The permutative λ-calculus $\Lambda_{\hat{\mathtt{P}}}$

We write $\to_{\{\beta,\hat{\mathtt{u}}\}}$ for $\to_{\beta} \cup \to_{\hat{\mathtt{u}}}$. The **permutative equivalence relation** $\equiv_{\hat{\mathtt{P}}}$ is generated by the contextual and reflexive-transitive closure of α-conversion and all the equations in $\hat{\mathtt{P}}$.

Given a reduction (resp. equivalence) relation $\mathcal{R}$ (resp. $\mathcal{E}$), the **reduction relation modulo** $\to_{\mathcal{R}/\mathcal{E}}$ is defined as $\mathcal{R}$-reduction on $\mathcal{E}$-equivalence classes, *i.e.* $t \to_{\mathcal{R}/\mathcal{E}} t'$ iff $\exists t_0, t_1$ s.t. $t \equiv_{\mathcal{E}} t_0 \to_{\mathcal{R}} t_1 \equiv_{\mathcal{E}} t'$. In this paper we give particular attention to the reduction relations $\to_{\beta/\hat{\mathtt{P}}}$ and $\to_{\{\beta,\hat{\mathtt{u}}\}/\hat{\mathtt{P}}}$.

Given any reduction relation $\mathcal{R}$, we use $\to_{\mathcal{R}}^{+}$ (resp. $\to_{\mathcal{R}}^{*}$) for the transitive (resp. reflexive-transitive) closure of $\mathcal{R}$. The notation $\leftrightarrow_{\mathcal{R}}$ is used for $\to_{\mathcal{R}} \cup {}_{\mathcal{R}}\!\leftarrow$ and $\to_{\mathcal{R}}^{k}$ for k compositions of $\to_{\mathcal{R}}$ with itself. A term t is in **$\mathcal{R}$-normal form**, written $t \in \mathcal{R}$**-nf**, if there is no t' such that $t \to_{\mathcal{R}} t'$. A term t has an **$\mathcal{R}$-normal form** iff there exists $t' \in \mathcal{R}$-nf such that $t \to_{\mathcal{R}}^{*} t'$. When t has a **unique** $\mathcal{R}$-normal form, this one is denoted by $\mathcal{R}(t)$. A reduction system $\mathcal{R}$ is **confluent** iff $t \to_{\mathcal{R}}^{*} u$ and $t \to_{\mathcal{R}}^{*} v$ implies there exists t' s.t. $u \to_{\mathcal{R}}^{*} t'$ and $v \to_{\mathcal{R}}^{*} t'$.

3 Towards Confluence of $\to_{\beta/\hat{\mathtt{P}}}$

Well-known notions [3, 19] in λ-calculus equipped with β-reduction are those of residual, created redex, (complete) development, etc. Informally, a λ-term t is either a normal form or it contains some redexes. However, if one reduces all the redexes in t it is not always the case that the obtained term is a normal form: redexes can be dynamically created along a reduction. A **development** of a term t is a reduction sequence starting at t in which only *residuals* of redexes that already exist in t are contracted along the sequence. Developments always terminate [19]; moreover all complete (*i.e.* maximal) developments terminate on the same term [9].

From now on we only consider complete developments, and to simplify the text, we omit the adjective *complete*. Analogously for the other notions of developments to be introduced in a while. A known method to show confluence of reduction relations is based on developments, particularly using the so-called Z-property [20]. **A reduction relation $\mathcal{R}$ satisfies the Z-property** iff there exists a map $^{\sharp}$ s.t.

$$\text{for all } t, \text{ for all } u,\ \ t \to_{\mathcal{R}} u \text{ implies } u \to_{\mathcal{R}}^{*} t^{\sharp} \text{ and } t^{\sharp} \to_{\mathcal{R}}^{*} u^{\sharp}$$

The requirement $u \to_{\mathcal{R}}^* t^\sharp$ expresses the fact that $t^\sharp$ is obtained by reducing at least all redexes in t, hence it abstracts away the role of developments. Note that if $\mathcal{R}$ satisfies the Z-property, and $t = t^\sharp$ for every $\mathcal{R}$-normal form, then $t \to_{\mathcal{R}}^* t^\sharp$ holds for every term t.

Theorem 1. *[20] If $\mathcal{R}$ satisfies the Z-property, then $\mathcal{R}$ is confluent.*

Confluence of β-reduction in λ-calculus can be proved by defining the map $^\sharp$ to be the function which computes the (complete) development of a term. To use this same technique to prove confluence of the relation $\to_{\beta/\hat{\mathtt{P}}}$, one first needs to generalise the Z-property to reduction modulo. **The reduction relation $\mathcal{R}$ satisfies the Z-property modulo the equivalence relation** $\mathtt{E}$ if there exists a map $^\sharp$ s.t. for all t, for all u,

1. $t \to_{\mathcal{R}} u$ implies $u \to_{\mathcal{R}}^* t^\sharp$ and $t^\sharp \to_{\mathcal{R}}^* u^\sharp$, and
2. $t \ \mathtt{E} \ u$ implies $t^\sharp = u^\sharp$.

It is easy to show that if $\mathcal{R}$ satisfies the Z-property modulo $\mathtt{E}$, then the reduction relation $\to_{\mathcal{R}/\mathtt{E}}$ is confluent. Actually, it implies that $\mathcal{R}$ is Church-Rosser modulo $\mathtt{E}$, which is the strongest possible notion of confluence in the realm of reduction modulo.

Lemma 1 (Z-property modulo $\Rightarrow$ Church-Rosser modulo). *If $\mathcal{R}$ satisfies the Z-property modulo $\mathtt{E}$, then $\mathcal{R}$ is Church-Rosser modulo $\mathtt{E}$, i.e. $\forall t, \forall u, \exists t_1, \exists u_1$ s.t. $t \ (\leftrightarrow_{\mathcal{R}} \cup \mathtt{E})^* \ u$ implies $t \to_{\mathcal{R}}^* t_1 \ \mathtt{E} \ u_1 \ {}_{\mathcal{R}}^*\!\leftarrow u$.*

Proof. Let $^\sharp$ be the map satisfying the Z-property for $\mathcal{R}$ modulo $\mathtt{E}$. Define $t^{\sharp\sharp} := t$ if t is an $\mathcal{R}$-nf, $t^{\sharp\sharp} := t^\sharp$ otherwise. Trivially, also $^{\sharp\sharp}$ satisfies the Z-property for $\mathcal{R}$ modulo $\mathtt{E}$. Moreover, $t \to_{\mathcal{R}}^* t^{\sharp\sharp}$ for every term t. Now, by the Z-property, $t \leftrightarrow_{\mathcal{R}} u$ implies $t^{\sharp\sharp} \leftrightarrow_{\mathcal{R}}^* u^{\sharp\sharp}$ and $t \ \mathtt{E} \ u$ implies $t^{\sharp\sharp} = u^{\sharp\sharp}$. Thus, $t \ (\leftrightarrow_{\mathcal{R}} \cup \mathtt{E})^* \ u$ implies $t^{\sharp\sharp} \leftrightarrow_{\mathcal{R}}^* u^{\sharp\sharp}$. Since $\mathcal{R}$ is confluent (Th. 1), then $\exists \ v$ s.t. $t^{\sharp\sharp} \to_{\mathcal{R}}^* v \ {}_{\mathcal{R}}^*\!\leftarrow u^{\sharp\sharp}$. We then conclude $t \to_{\mathcal{R}}^* t^{\sharp\sharp} \to_{\mathcal{R}}^* v \ {}_{\mathcal{R}}^*\!\leftarrow u^{\sharp\sharp} \ {}_{\mathcal{R}}^*\!\leftarrow u$.

Church-Rosser modulo has two important corollaries.

Corollary 1. *[19] Let $\mathcal{R}$ be Church-Rosser modulo $\mathtt{E}$. Then:*

1. **Uniqueness of Normal Forms***: if $t \ (\leftrightarrow_{\mathcal{R}} \cup \mathtt{E})^* \ u$ and $t \to_{\mathcal{R}}^* t_1$ and $u \to_{\mathcal{R}}^* u_1$ and t_1, u_1 are $\mathcal{R}$-nf, then $t_1 \ \mathtt{E} \ u_1$.*
2. **Confluence of the reduction modulo***: if $t \to_{\mathcal{R}/\mathtt{E}}^* u_i \ (i = 1, 2)$, then $\exists \ t'$ s.t. $u_i \ \to_{\mathcal{R}/\mathtt{E}}^* t' \ (i = 1, 2)$.*

The first natural attempt to prove confluence for the reduction relation $\rightarrow_{\beta/\hat{\mathtt{P}}}$ is then to use Lem. 1 by choosing $^\sharp$ as the development function. Unfortunately, this idea does not work: for instance $t = (\lambda x.\lambda y.y)\ z\ w \equiv_{\hat{\sigma}_1} u = (\lambda y.((\lambda x.y)\ z))\ w$ but $(\lambda y.y)\ w$, the development of t, is different from w, the development of u. The reason is that $\hat{\sigma}_1$ *creates redexes*.

In λ-calculus creation of redexes can be classified in three types [15]:

(Type 1) $((\lambda x.\lambda y.t)\ u)\ v \rightarrow_\beta (\lambda y.t\{x/u\})\ v$.
(Type 2) $(\lambda x.x)\ (\lambda y.t)\ u \rightarrow_\beta (\lambda y.t)\ u$.
(Type 3) $(\lambda x.C[x\ v])\ (\lambda y.u) \rightarrow_\beta C\{x/\lambda y.u\}[(\lambda y.u)\ v\{x/\lambda y.u\}]$

Developments do not work to show confluence of $\rightarrow_{\beta/\hat{\mathtt{P}}}$ because $\hat{\sigma}_1$ *anticipates/postpones* creations of Type 1, making these creations visible/hidden in the starting term. However, another well-known notion of development, called superdevelopment [13] or **L-development** [1], exists. A (complete) superdevelopment [13] of a term t is a reduction sequence starting at t in which only *residuals* of redexes that already exist in t and created redexes of Type 1 and 2 are allowed to be contracted along the sequence.

Unfortunately, superdevelopments do not work either: for instance $t = (\lambda x.(x\ y))\ I \equiv_{\hat{\sigma}_2} (\lambda x.x)\ I\ y = u$, where I is the identity function, but their superdevelopments are different. Now the reason is more subtle: $\hat{\sigma}_2$ does not anticipate creations of Type 2, but it turns future creations of Type 2 (e.g. in u) into creations of Type 3 (in t), or viceversa. The solution is to weaken the notion of L-development to that of M-development. A (complete) **M-development** of a term t is a reduction sequence starting at t in which only *residuals* of redexes that already exist in t and created redexes of Type 1 — but not of Type 2 — are allowed to be contracted along the sequence. We are going to define the result t° of a (complete) M-development by using an auxiliary calculus, λsub, having explicit substitutions. On one hand this seems to be necessary, because apparently there is no way to describe such a term by induction on t inside $\Lambda_{\hat{\mathtt{P}}}$, as it is the case for (L-)developments. On the other hand the use of λsub will not be costly: we shall obtain a concise proof of confluence for $\rightarrow_{\beta/\hat{\mathtt{P}}}$.

4 The auxiliary λsub-calculus

This section introduces the λsub-calculus which is used as an auxiliary tool to show confluence of the permutative λ-calculus. The set $\mathcal{T}$ of terms of the λsub-calculus is given by **variables** x, **abstractions** $\lambda x.t$, **applications** $t\ u$ and **substituted terms** $t[x/u]$. The object $[x/t]$, which is not a term, is called an **explicit substitution (ES)**. We consider **free**

$$(\lambda x.t)\mathtt{L}\ u \mapsto_{\mathtt{dB}} t[x/u]\mathtt{L}$$
$$t[x/u] \mapsto_{\mathtt{sub}} t\{x/u\}$$

Fig. 2: The $\lambda\mathtt{sub}$-calculus

and **bound** variables of terms with ES as usual [11]. The meta-level **substitution** operation and the α-conversion operation are extended from Λ to $\mathcal{T}$ as expected. We use $\mathtt{L}$ to denote a possibly empty list of ES $[y_1/t_1]\dots[y_m/t_m]$. We write C to denote a **context** in $\lambda\mathtt{sub}$. The rewriting rules of the $\lambda\mathtt{sub}$-calculus are given in Fig. 2.

One feature of $\lambda\mathtt{sub}$ is that rule $\mathtt{dB}$ acts **at a distance**, as in Proof-Nets [8]. Indeed, the list $\mathtt{L}$ of ES introduces some distance between the function $\lambda x.t$ and its argument u in a term of the form $(\lambda x.t)\mathtt{L}\ u$. Rule $\to_{\mathtt{dB}}$ (resp. $\to_{\mathtt{sub}}$) corresponds exactly to the multiplicative (resp. exponential) cut-elimination rule of Pure Proof-Nets. Another feature of $\lambda\mathtt{sub}$ is that it splits $\to_\beta$, which does not always terminate, into two terminating and confluent reduction systems $\mathtt{dB}$ and $\mathtt{sub}$ (property which follows from [1]), through which $\to_\beta$ can be finely studied.

Lemma 2. *The reduction system* $\mathtt{dB}$ *(resp.* $\mathtt{sub}$*) is confluent and terminating, thus* $\mathtt{dB}$ *(resp.* $\mathtt{sub}$*)-normal forms always exist and are unique.*

From now on, we write $\mathtt{dB}(t)$ (resp. $\mathtt{sub}(t)$) for the unique $\mathtt{dB}$-normal form (resp. $\mathtt{sub}$-normal form) of the term t.

Lemma 3. *The* $\mathtt{sub}$*-function enjoys the following equalities.*

$$\mathtt{sub}(x) = x \qquad \mathtt{sub}(t[x/u]) = \mathtt{sub}(t)\{x/\mathtt{sub}(u)\}$$
$$\mathtt{sub}(\lambda x.t) = \lambda x.\mathtt{sub}(t) \qquad \mathtt{sub}(t\ u) = \mathtt{sub}(t)\ \mathtt{sub}(u)$$

The following property is of course expected, it is shown by induction on the reduction relation by using the previous characterisation.

Lemma 4 (Projection on $\to_\beta$). *If* $t_0 \to_{\lambda\mathtt{sub}} t_1$*, then* $\mathtt{sub}(t_0) \to_\beta^* \mathtt{sub}(t_1)$.

We now define the **$\mathtt{M}$-development** of a term $t \in \mathcal{T}$ as a special normal form in $\lambda\mathtt{sub}$:

$$t^\circ := \mathtt{sub}(\mathtt{dB}(t))$$

The $\mathtt{M}$-development of t thus reduces all its multiple applications, *i.e.* applications of functions to several arguments. Consider a simple example of creation of Type 1 which applies a function to two arguments: $((\lambda x.\lambda y.y)\ z)\ z' \to_\beta (\lambda y.y)\ z'$. The reduction to $\mathtt{dB}$-normal form:

$$((\lambda x.\lambda y.y)\ z)\ z' \to_{\mathtt{dB}} (\lambda y.y)[x/z]\ z' \to_{\mathtt{dB}} y[y/z][x/z']$$

$$
\begin{array}{lll}
(\lambda x.t)\mathtt{L}\ u & \mapsto_{\mathtt{dB}} & t[x/u]\mathtt{L} \\
t[x/u] & \mapsto_{\mathtt{sub}} & t\{x/u\}
\end{array}
$$

$$
\begin{array}{lll}
(\lambda x.\lambda y.t)\ u & \sim_{\hat{\sigma}_1} & \lambda y.((\lambda x.t)\ u) \\
(\lambda x.t\ v)\ u & \sim_{\hat{\sigma}_2} & (\lambda x.t)\ u\ v \\
t\ ((\lambda x.v)\ u) & \sim_{\widehat{\mathtt{box}}} & (\lambda x.t\ v)\ u
\end{array}
\qquad
\begin{array}{l}
\\
\text{if } y \notin \mathtt{fv}(u) \\
\text{if } x \notin \mathtt{fv}(v) \\
\text{if } x \notin \mathtt{fv}(t)\ \&\ x \in \mathtt{fv}(v)
\end{array}
$$

$$
\begin{array}{lll}
t[x/s][y/v] & \sim_{\mathtt{CS}} & t[y/v][x/s] \\
\lambda y.(t[x/s]) & \sim_{\sigma_1} & (\lambda y.t)[x/s] \\
t[x/s]\ v & \sim_{\sigma_2} & (t\ v)[x/s] \\
(t\ v)[x/u] & \sim_{\mathtt{box}_1} & t\ v[x/u] \\
t[y/v][x/u] & \sim_{\mathtt{box}_2} & t[y/v[x/u]]
\end{array}
\qquad
\begin{array}{l}
\text{if } x \notin \mathtt{fv}(v)\ \&\ y \notin \mathtt{fv}(s) \\
\text{if } y \notin \mathtt{fv}(s) \\
\text{if } x \notin \mathtt{fv}(v) \\
\text{if } x \notin \mathtt{fv}(t)\ \&\ x \in \mathtt{fv}(v) \\
\text{if } x \notin \mathtt{fv}(t)\ \&\ x \in \mathtt{fv}(v)
\end{array}
$$

Fig. 3: The calculus $\lambda\mathtt{sub}/\Pi\!\!\Pi$

reduces in particular a created dB-redex, then, reduction to sub-normal form $y[y/z][x/z] \to^*_{\mathtt{sub}} z$ completes the M-development. Note that the second dB-step is possible only because the rule acts at a *distance*.

Note that the definition of **M-development** uses the reduction rules of $\lambda\mathtt{sub}$, which are external to $\Lambda_{\hat{\mathtt{P}}}$, since they are defined on $\mathcal{T}$ and not on Λ. However, this definition makes sense also when one looks only at $\Lambda_{\hat{\mathtt{P}}}$, as stated by next Lemma, which can be shown by induction using Lem. 4.

Lemma 5. *Let t be a λ-term. Then t° is a λ-term and $t \to^*_\beta t^\circ$.*

The Z-property for the permutative λ-calculus can be proved by means of M-developments. One of the properties to be verified is that $t_0 \equiv_{\hat{\mathtt{P}}} t_1$ implies $t_0^\circ = t_1^\circ$, which is quite tricky. To simplify this proof, and also to later show PSN for $\Lambda_{\hat{\mathtt{P}}}$, we need to extend $\lambda\mathtt{sub}$ with some equations, resulting in the equational calculus $\lambda\mathtt{sub}/\Pi\!\!\Pi$ in Fig. 3.

The equations are divided in two groups $\hat{\mathtt{P}} = \{\hat{\sigma}_1, \hat{\sigma}_2, \widehat{\mathtt{box}}\}$ and $\mathtt{P} = \{\mathtt{CS}, \sigma_1, \sigma_2, \mathtt{box}_1, \mathtt{box}_2\}$. We write $\equiv_{\hat{\mathtt{P}}}$, $\equiv_{\mathtt{P}}$ and $\equiv_{\Pi\!\!\Pi}$ for the contextual, reflexive-transitive closure of α-conversion and all the equations in $\hat{\mathtt{P}}$, $\mathtt{P}$ and $\hat{\mathtt{P}} \cup \mathtt{P}$, respectively. We use $\to_{\lambda\mathtt{sub}/\Pi\!\!\Pi}$ for reduction $\to_{\lambda\mathtt{sub}}$ modulo $\equiv_{\Pi\!\!\Pi}$.

The first group $\hat{\mathtt{P}}$ of equations is the same of $\Lambda_{\hat{\mathtt{P}}}$, but now also on terms with ES. The second group $\mathtt{P}$ is obtained by projecting the equations of $\hat{\mathtt{P}}$ acting on λ-terms into terms with ES by means of $\to_{\mathtt{dB}}$:

$$
\begin{array}{ccc}
(\lambda x.\lambda y.t)\ u & \equiv_{\hat{\sigma}_1} & \lambda y.((\lambda x.t)\ u) \\
\downarrow_{\mathtt{dB}} & & \downarrow_{\mathtt{dB}} \\
(\lambda y.t)[x/u] & \equiv_{\sigma_1} & \lambda y.(t[x/u])
\end{array}
\qquad
\begin{array}{ccc}
((\lambda x.t)\ u)\ v & \equiv_{\hat{\sigma}_2} & (\lambda x.(t\ v))\ u \\
\downarrow_{\mathtt{dB}} & & \downarrow_{\mathtt{dB}} \\
(t[x/u])\ v & \equiv_{\sigma_2} & (t\ v)[x/u]
\end{array}
\qquad (\mathrm{A})
$$

Analogously, $\equiv_{\mathtt{box}_1}$ and $\equiv_{\mathtt{box}_2}$ are obtained by projecting $\equiv_{\widehat{\mathtt{box}}}$. Thus,

$$
\begin{array}{ccc}
t\ ((\lambda x.v)\ u) & \equiv_{\widehat{\mathtt{box}}} & ((\lambda x.t\ v)\ u) \\
\downarrow_{\mathtt{dB}} & & \downarrow_{\mathtt{dB}} \\
t\ v[x/u] & \equiv_{\mathtt{box}_1} & (t\ v)[x/u]
\end{array}
\qquad
\begin{array}{ccc}
(\lambda y.t)\ ((\lambda x.v)\ u) & \equiv_{\widehat{\mathtt{box}}} & ((\lambda x.(\lambda y.t)\ v)\ u) \\
\downarrow_{*\mathtt{dB}} & & \downarrow_{*\mathtt{dB}} \\
t[y/v[x/u]] & \equiv_{\mathtt{box}_2} & t[y/v][x/u]
\end{array}
$$

Obtaining $\equiv_{\mathtt{CS}}$ is more subtle, indeed $t[y/v][x/u] \equiv_{\mathtt{CS}} t[x/u][y/v]$ can be understood as the $\mathtt{dB}$-projection of $(\lambda x.((\lambda y.t)v))u \equiv_{\hat{\sigma}_1,\hat{\sigma}_2} (\lambda y.((\lambda x.t)u))v$. The equivalence relation generated by the equations $\{\mathtt{CS}, \sigma_1, \sigma_2\}$ on the set $\mathcal{T}$ can be understood by means of the translation from terms with ES to Pure Proof-Nets. Equations $\{\mathtt{box}_1, \mathtt{box}_2\}$ are obtained by taking the box-box commutative rule of Proof-Nets as an equation rather than a rule (which is a novelty of our study). The equations of $\hat{\mathtt{P}}$, which are exactly those of $\Lambda_{\hat{\mathtt{P}}}$, are obtained by lifting those of $\mathtt{P}$ from terms with ES to λ-terms, which is the reason for writing $\hat{\_}$.

5 The *Z*-property modulo by means of M-developments

We prove here that $\to_\beta$ satisfies the Z-property modulo $\equiv_{\hat{\mathtt{P}}}$ by means of the notion of $\mathtt{M}$-developments introduced in Sec. 4. The new equivalence $\equiv_{\mathtt{P}}$ on terms with ES allows to prove that $t_0 \equiv_{\hat{\mathtt{P}}} t_1$ implies $t_0^\circ = t_1^\circ$ by *continuously* extending $\equiv_{\hat{\mathtt{P}}}$ through reduction. This notion of continuity is a strong form of the so-called **local coherence** (see [19], pp. 769-770).

Lemma 6. *Let $t, u, u_1, u_2 \in \mathcal{T}$.*

1. $\mathtt{dB}$**-Continuity of** $\equiv_\Pi$*: if $t \equiv_\Pi u_1$ and $t \to_{\mathtt{dB}} u_2$ then exists v s. t. $u_1 \to_{\mathtt{dB}} v$ and $u_2 \equiv_\Pi v$.*
2. **Projecting** $\equiv_\Pi$ **by** $\mathtt{dB}$**-nf***: if $t \equiv_\Pi u$ then $\mathtt{dB}(t) \equiv_{\mathtt{P}} \mathtt{dB}(u)$.*
3. **Projecting** $\equiv_{\mathtt{P}}$ **by** $\mathtt{sub}$**-nf***: if $t \equiv_{\mathtt{P}} u$ then $\mathtt{sub}(t) = \mathtt{sub}(u)$.*
4. **Projecting** $\equiv_\Pi$ **by** $\mathtt{M}$**-developments***: if $t \equiv_\Pi u$ then $t^\circ = u^\circ$.*

Proof. 1. By induction on $\equiv_\Pi$. The base case is as in the equations labelled (A) on Page 8. For instance: if $t = (\lambda x.(s\ w))\ r \sim_{\widehat{\mathtt{box}}} s\ ((\lambda x.w)\ r) = u_1$ with $x \notin \mathtt{fv}(s)$ and $x \in w$, and if $t \to_{\mathtt{dB}} (s\ w)[x/r] = u_2$ then $u_1 \to_{\mathtt{dB}} s\ w[x/r] = v$ and $u_2 \sim_{\mathtt{box}_1} v$. The inductive cases are straightforward.
2. By induction on the length of $t \to_{\mathtt{dB}}^* \mathtt{dB}(t)$ using Point 1. one gets $\mathtt{dB}(t) \equiv_\Pi \mathtt{dB}(u)$. We conclude since $\equiv_\Pi$ coincides with $\equiv_{\mathtt{P}}$ on $\mathtt{dB}$-nfs.
3. By induction on $\equiv_{\mathtt{P}}$ using the characterisation in Lem. 3.
4. By composing the two previous points.

Now we need to show the Z-property for $\to_\beta$ with respect to $\mathtt{M}$-developments. This is done by analysing the commutation of $\to_{\mathtt{dB}}$ and $\to_{\mathtt{sub}}$. We first prove the result for $\to_{\lambda\mathtt{sub}}$, thus obtaining the Z-property for $\to_\beta$ modulo $\hat{\mathtt{P}}$ as a corollary.

Lemma 7 (Commutation of $\to_{\mathtt{sub}}^*$ **and** $\to_{\mathtt{dB}}^*$**).** *Let $t, u_1, u_2 \in \mathcal{T}$. If $t \to_{\mathtt{sub}}^k u_1$ and $t \to_{\mathtt{dB}}^h u_2$ then there exists v s.t. $u_2 \to_{\mathtt{sub}}^k v$ and $u_1 \to_{\mathtt{dB}}^* v$.*

Proof. By induction on the pair $\langle k, h\rangle$ ordered lexicographically, using local commutation (case $k = h = 1$), which is proved by induction on t.

Lemma 8 (Z-property for $\to_{\lambda\mathtt{sub}}$). *Let $t, u \in \mathcal{T}$. Then:*

1. *If $t \to_{\lambda\mathtt{sub}} u$ then $u \to^*_{\lambda\mathtt{sub}} t^\circ$.*
2. *If $t \to_{\lambda\mathtt{sub}} u$ then $t^\circ \to^*_{\lambda\mathtt{sub}} u^\circ$.*

Proof. 1. If $t \to_{\mathtt{dB}} u$ then $t^\circ = u^\circ$ (thus $u \to^*_{\lambda\mathtt{sub}} u^\circ = t^\circ$ holds). If $t \to_{\mathtt{sub}} u$ then by Lem. 7 $\exists\ v$ s.t. $\mathtt{dB}(t) \to^*_{\mathtt{sub}} v$ and $u \to^*_{\mathtt{dB}} v$. We have $\mathtt{sub}(\mathtt{dB}(t)) = \mathtt{sub}(v)$ and so $u \to^*_{\mathtt{dB}} v \to^*_{\mathtt{sub}} \mathtt{sub}(\mathtt{dB}(t)) = t^\circ$.
2. If $t \to_{\mathtt{dB}} u$ then $t^\circ = u^\circ$. If $t \to_{\mathtt{sub}} u$ then by Lem. 7 exists v s.t. $u \to^*_{\mathtt{dB}} v$ and $\mathtt{dB}(t) \to^*_{\mathtt{sub}} v$. By definition $v \to^*_{\mathtt{dB}} \mathtt{dB}(u)$ and $v \to^*_{\mathtt{sub}} t^\circ$. By Lem. 7 exists w s.t. $\mathtt{dB}(u) \to^*_{\mathtt{sub}} w$ and $t^\circ \to^*_{\mathtt{dB}} w$. By definition $w \to^*_{\mathtt{sub}} \mathtt{sub}(\mathtt{dB}(u)) = u^\circ$, therefore $t^\circ \to^*_{\lambda\mathtt{sub}} u^\circ$.

Corollary 2 (Z-property for $\to_\beta$). *Let $t, u \in \Lambda$. Then:*

1. *If $t \to_\beta u$ then $u \to^*_\beta t^\circ$.*
2. *If $t \to_\beta u$ then $t^\circ \to^*_\beta u^\circ$.*

Proof. 1. If $t \to_\beta u$ then $t \to_{\mathtt{dB}} u_1 \to_{\mathtt{sub}} u$. We have $t^\circ = u_1^\circ$ and by Lem. 8:1 $u \to^*_{\lambda\mathtt{sub}} u_1^\circ$, hence $u \to^*_{\lambda\mathtt{sub}} t^\circ$. By Lem. 4 we get $\mathtt{sub}(u) \to^*_\beta \mathtt{sub}(t^\circ)$ and we conclude since both $\mathtt{sub}(u) = u$ and $\mathtt{sub}(t^\circ) = t^\circ$ hold, given that both u and t° are λ-terms.
2. If $t \to_\beta u$, then $t \to_{\mathtt{dB}} u_1 \to_{\mathtt{sub}} u$, Lem. 8:2 gives $t^\circ \to^*_{\lambda\mathtt{sub}} u_1^\circ \to^*_{\lambda\mathtt{sub}} u^\circ$. As in the previous point we conclude $t^\circ \to^*_\beta u^\circ$ by Lem. 4.

Thus we get:

Theorem 2. *1. The relation $\to_{\lambda\mathtt{sub}}$ is Church-Rosser modulo $\equiv_{\Pi}$.*
2. The relation $\to_\beta$ is Church-Rosser modulo $\equiv_{\hat{\mathtt{P}}}$.

Proof. Lem. 8 and Lem. 6:3 prove the Z-property for $\to_{\lambda\mathtt{sub}}$ modulo $\equiv_{\Pi}$. Cor. 2 and Lem. 6:4 prove the Z-property for $\to_\beta$ modulo $\hat{\mathtt{P}}$. Church-Rosser modulo follows in both cases from Lem. 1.

Note that our proof of confluence for $\Lambda_{\hat{\mathtt{P}}}$ and $\lambda\mathtt{sub}$ modulo $\equiv_{\Pi}$ *does not* use confluence of λ-calculus.

6 Adding the unboxing rule

In this section we add the void unboxing rule in order to lift our confluence result from $\to_{\beta/\hat{\mathtt{P}}}$ to $\to_{\{\beta,\hat{\mathtt{u}}\}/\hat{\mathtt{P}}}$.

The application construct $t\ u$ is *linear* in t and *non-linear* in u, in the sense that u can be duplicated/erased (for instance if $t = \lambda x.x\ x$) while t cannot. The translation of λ-calculus into Linear Logic makes this point explicit: u is placed in a !-box—the construction allowing non-linearity—while t is not. The natural permutation (note the absence of the side condition "$x \in \mathtt{fv}(v)$"):

$$t\ ((\lambda x.v)\ u) \sim_{\mathtt{badbox}} (\lambda x.t\ v)\ u \quad \text{if } x \notin \mathtt{fv}(t)$$

is delicate, because it permutes a redex in/out of a non-linear sub-term, and thus affects reduction lengths. Indeed, we now show that $\to_\beta$ plus the equations $\{\hat{\sigma}_2, \mathtt{badbox}\}$ does *not* preserve β-strong normalisation, *i.e.* there exists $t \in \mathcal{SN}_\beta$ s.t. $t \notin \mathcal{SN}_{\beta/\{\hat{\sigma}_2,\mathtt{badbox}\}}$ as the following example shows. Let $t = (\lambda x.u)\ u$, where $u = (\lambda z.z\ z)\ y$. Then,

$$\begin{array}{lllll} t = (\lambda x.u)\ u & = & (\lambda x.((\lambda z.z\ z)\ y))\ u & \equiv_{\mathtt{badbox}} \\ (\lambda z.z\ z)((\lambda x.y)\ u) & \to_\beta & ((\lambda x.y)\ u)\ ((\lambda x.y)\ u) & \equiv_{\hat{\sigma}_2} \\ (\lambda x.y\ ((\lambda x.y)\ u))\ u & \equiv_{\mathtt{badbox}} & (\lambda x.(\lambda x.y\ y)\ u)\ u & \equiv_{\mathtt{badbox}} \\ (\lambda x.y\ y)\ ((\lambda x.u)\ u) & = & (\lambda x.y\ y)\ t & \end{array}$$

The term t reduces to a term containing t so that $t \notin \mathcal{SN}_{\beta/\{\hat{\sigma}_2,\mathtt{badbox}\}}$. Note that in the counter-example the equation `badbox` is used with respect to a λ-abstraction binding a variable which does not occur in the body.

Thus we split the previous equation $\sim_{\mathtt{badbox}}$ in two cases: the case "$x \in \mathtt{fv}(v)$" goes to the equation $\widehat{\mathtt{box}}$, while the case "$x \notin \mathtt{fv}(v)$" is captured by the void unboxing rewriting rule $\mapsto_{\hat{\mathtt{u}}}$ in Fig. 1, which is just an orientation from left to right of the dangerous equation $\sim_{\mathtt{badbox}}$. The idea behind a reduction step $t\ ((\lambda x.v)\ u) \to_{\hat{\mathtt{u}}} (\lambda x.t\ v)\ u$ is that both sides of the rule β-reduce to $t\ v$, or, equivalently, the permuted redex simply erases u, which therefore can be considered as *garbage*. The interest in permuting garbage is to get it out of the arguments, so that it does not get duplicated. Indeed, consider the case where t (in the rule) is $\lambda y.y\ y$: a β-reduction step from the left-hand side duplicates u, while this is not the case for the right-hand side.

Void unboxing is a rewriting rule but it behaves exactly as the other equations with respect to $\mathtt{M}$-developments, *i.e.* $t_0 \to_{\hat{\mathtt{u}}} t_1$ implies $t_0^\circ = t_1^\circ$. To show this property we proceed as for the other equations, *i.e.* we extend $\lambda\mathtt{sub}/\Pi$ with a rule $\mapsto_{\mathtt{u}}$ reflecting $\mapsto_{\hat{\mathtt{u}}}$ on terms with ES. To specify the rewriting rule $\mapsto_{\mathtt{u}}$, we first need to define a special notion of context. A **boxed context** B is given by the following grammar:

$$B ::= t\ C \mid t[x/C] \mid B\ t \mid B[x/t] \mid \lambda y.B$$

$$B[\![t[x/u]]\!] \mapsto_{\mathtt{u}} B[\![t]\!][x/u] \quad \text{if } B \text{ does not capture variables in } \mathtt{fv}(u)$$

Fig. 4: The unboxing rule

where C denotes a context in $\lambda\mathtt{sub}$. The name *boxed context* is justified by the Proof-Net representation of λ-terms (with explicit substitutions): every argument of an application or content of an ES is denoted by a !-box, hence the hole of a boxed context necessarily occurs inside a !-box. The unboxing rule for terms with ES is the context closure of the rule in Fig. 4 (for technical reasons $\mapsto_{\mathtt{u}}$ is more general than the projection of $\mapsto_{\hat{\mathtt{u}}}$ by $\mathtt{dB}$-steps). Next lemma relates unboxing and $\mathtt{M}$-developments.

Lemma 9. *Let $t, u_1, u_2 \in \mathcal{T}$.*

1. **Commutation of $\to_{\{\mathtt{u},\hat{\mathtt{u}}\}}$ and $\to^*_{\mathtt{dB}}$***: if $t \to_{\{\mathtt{u},\hat{\mathtt{u}}\}} u_2$ and $t \to^k_{\mathtt{dB}} u_1$ then there exists v s.t. $u_2 \to^k_{\mathtt{dB}} v$ and $u_1 \to_{\{\mathtt{u},\hat{\mathtt{u}}\}} v$.*
2. **Projecting $\to_{\{\mathtt{u},\hat{\mathtt{u}}\}}$ by $\mathtt{dB}$-nf***: If $t \to_{\{\mathtt{u},\hat{\mathtt{u}}\}} u$, then $\mathtt{dB}(t) \to_{\mathtt{u}} \mathtt{dB}(u)$.*
3. **Projecting $\to_{\{\mathtt{u},\hat{\mathtt{u}}\}}$ by $\mathtt{M}$-developments***: If $t \to_{\{\mathtt{u},\hat{\mathtt{u}}\}} u$, then $t^\circ = u^\circ$.*

Proof. 1. By induction on k. The case $k = 1$ is by induction on $t \to_{\{\mathtt{u},\hat{\mathtt{u}}\}} u_2$. The only interesting subcases are the root ones for $t \to_{\hat{\mathtt{u}}} u_2$ and $t \to_{\mathtt{u}} u_2$, given by the following diagrams:

$$\begin{array}{ccc} t = s\ ((\lambda x.w)\ r) & \to_{\hat{\mathtt{u}}} & (\lambda x.s\ w)\ r = u_2 \\ \downarrow_{\mathtt{dB}} & & \downarrow_{\mathtt{dB}} \\ s\ w[x/u] & \to_{\mathtt{u}} & (s\ w)[x/r] \end{array} \qquad \begin{array}{ccc} t = (\lambda y.s)\mathtt{L}\ w[x/u] & \to_{\mathtt{u}} & ((\lambda y.s)\mathtt{L}\ w)[x/u] = u_2 \\ \downarrow_{\mathtt{dB}} & & \downarrow_{\mathtt{dB}} \\ s[y/w[x/u]]\mathtt{L} & \to_{\mathtt{u}} & s[y/w]\mathtt{L}[x/u] \end{array}$$

The other subcases and inductive cases are all straightforward.
2. By the previous point there exists v s.t. $\mathtt{dB}(t) \to_{\mathtt{u},\hat{\mathtt{u}}} v$ and $u \to^*_{\mathtt{dB}} v$. But $\mathtt{dB}$-normal forms cannot $\to_{\hat{\mathtt{u}}}$-reduce, so $\mathtt{dB}(t) \to_{\mathtt{u}} v$. Moreover, $\to_{\mathtt{u}}$ cannot create $\mathtt{dB}$-redexes, so $v = \mathtt{dB}(v) = \mathtt{dB}(u)$.
3. From 2. we get $\mathtt{dB}(t) \to_{\mathtt{u}} \mathtt{dB}(u)$. Thus $t^\circ = \mathtt{sub}(\mathtt{dB}(t)) = \mathtt{sub}(\mathtt{dB}(u)) = u^\circ$ since $\mathtt{u}$-reduction only moves one void substitution.

Thus we can extend our confluence results to void unboxing.

Corollary 3 (Z-property for unboxing). *Let $t, t_0 \in \Lambda$ and $u, u_0 \in \mathcal{T}$.*

1. *If $t \to_{\{\lambda\mathtt{sub},\hat{\mathtt{u}},\mathtt{u}\}} t_0$ then $t_0 \to^*_{\{\lambda\mathtt{sub},\hat{\mathtt{u}},\mathtt{u}\}} t^\circ$ and $t^\circ \to^*_{\{\lambda\mathtt{sub},\hat{\mathtt{u}},\mathtt{u}\}} t^\circ_0$.*
2. *If $u \to_{\{\beta,\hat{\mathtt{u}}\}} u_0$ then $u_0 \to^*_{\{\beta,\hat{\mathtt{u}}\}} u^\circ$ and $u^\circ \to^*_{\{\beta,\hat{\mathtt{u}}\}} u^\circ_0$.*

Proof. 1. For $\to_{\lambda\mathtt{sub}}$ use Lem. 8. Suppose $t \to_{\{\hat{\mathtt{u}},\mathtt{u}\}} t_0$. Then $t_0 \to^*_{\lambda\mathtt{sub}} t^\circ_0$ by definition and $t^\circ_0 = t^\circ$ by Lem. 9:3, which allow us to conclude.
2. For $\to_\beta$ use Cor. 2. For $\to_{\hat{\mathtt{u}}}$ use $u_0 \to^*_\beta u^\circ_0$ (Lem. 5) and Lem. 9:3.

As before we get:

Theorem 3. *1. The relation* $\to_{\{\lambda\mathtt{sub},\hat{\mathtt{u}},\mathtt{u}\}}$ *is Church-Rosser modulo* $\equiv_{\Pi}$*.*
2. The relation $\to_{\{\beta,\hat{\mathtt{u}}\}}$ *is Church-Rosser modulo* $\equiv_{\hat{\mathtt{P}}}$*.*

The extension of Church-Rosser modulo to unboxing relies on the fact that $t \to_{\hat{\mathtt{u}}} u$ implies $t^\circ = u^\circ$. Let $\to_{\hat{\mathtt{P}}}$ be the reduction system obtained by an arbitrary orientation of the equations in the set $\hat{\mathtt{P}}$. The reduction $\to_{\hat{\mathtt{P}}}$ enjoys the same property above for $\to_{\hat{\mathtt{u}}}$. Hence, we easily get the Z-property for $\to_{\{\beta,\hat{\mathtt{u}},\hat{\mathtt{P}}\}}$, and thus confluence holds. Note that even a stronger fact holds: it is possible to orient only some of the equations in $\hat{\mathtt{P}}$ keeping the other(s) as equations, and Church-Rosser modulo still holds.

7 Preservation of β-Strong Normalisation

In this section we show that $\Lambda_{\hat{\mathtt{P}}}$ enjoys PSN. As before, we shall actually prove PSN for $\{\lambda\mathtt{sub},\mathtt{u},\hat{\mathtt{u}}\}/\Pi$ and then deduce PSN for $\Lambda_{\hat{\mathtt{P}}}$. The proof simply consists in reducing the problem to the following result from [2].

Theorem 4. *The calculus* $\{\lambda\mathtt{j},\mathtt{u}\}/\mathtt{P}$ *enjoys PSN.*

Since $\lambda\mathtt{sub}$ can be seen as a sub-calculus of $\lambda\mathtt{j}$ (because $\to_{\lambda\mathtt{sub}} \subseteq \to_{\lambda\mathtt{j}}$, see [1]), from Th. 4 we immediately get the following corollary:

Theorem 5. *The calculus* $\{\lambda\mathtt{sub},\mathtt{u}\}/\mathtt{P}$ *enjoys PSN.*

In order to infer PSN for $\{\lambda\mathtt{sub},\mathtt{u},\hat{\mathtt{u}}\}/\Pi$ we need to show that $\equiv_{\Pi}$ and $\to_{\hat{\mathtt{u}}}$ preserve strong normalisation. The idea is to project reductions of $\{\lambda\mathtt{sub},\mathtt{u},\hat{\mathtt{u}}\}/\Pi$ over $\mathtt{dB}$-normal forms, since $\equiv_{\Pi}$ (resp. $\to_{\hat{\mathtt{u}}}$) collapses on $\equiv_{\mathtt{P}}$ (resp. $\to_{\mathtt{u}}$), and then show that this projection preserves strong normalisation. But this is trivial: $\to_{\mathtt{dB}}$ cannot erase any redex except the one it reduces. For $\to_{\hat{\mathtt{u}}}$ this is given by Lem. 9:2, while for $\to_{\mathtt{sub}}$ it is given by the the following lemma, where we get $\to^+_{\lambda\mathtt{sub}}$ and not just $\to^*_{\lambda\mathtt{sub}}$.

Lemma 10. *If* $t \to_{\mathtt{sub}} u$ *then* $\mathtt{dB}(t) \to^+_{\lambda\mathtt{sub}} \mathtt{dB}(u)$.

Proof. Lem. 7 applied to the hypothesis and $t \to^*_{\mathtt{dB}} \mathtt{dB}(t)$ gives v s.t. $u \to^*_{\mathtt{dB}} v$ and $\mathtt{dB}(t) \to_{\mathtt{sub}} v$. We conclude since $v \to^*_{\mathtt{dB}} \mathtt{dB}(v) = \mathtt{dB}(u)$ and so $\mathtt{dB}(t) \to_{\mathtt{sub}} v \to^*_{\mathtt{dB}} \mathtt{dB}(u)$.

Corollary 4. *The* $\{\lambda\mathtt{sub},\mathtt{u},\hat{\mathtt{u}}\}/\Pi$*-calculus enjoys PSN.*

Proof. Let $t \in \mathcal{SN}_\beta$ and suppose $t \notin \mathcal{SN}_{\{\lambda\mathtt{sub},\mathtt{u},\hat{\mathtt{u}}\}/\Pi}$. Then, there is an infinite $\{\lambda\mathtt{sub},\mathtt{u},\hat{\mathtt{u}}\}/\Pi$-reduction starting at t, and since $\mathtt{dB}$ modulo Π is a trivial well-founded relation, this reduction has necessarily the following form: $t \to^*_{\mathtt{dB}/\Pi} t_1 \to^+_{\{\mathtt{sub},\mathtt{u},\hat{\mathtt{u}}\}/\Pi} t_2 \to^*_{\mathtt{dB}/\Pi} t_3 \to^+_{\{\mathtt{sub},\mathtt{u},\hat{\mathtt{u}}\}/\Pi} t_4 \dots$. By Lem. 6:2, Lem. 10 and Lem. 9:2, we can transform this infinite reduction into an infinite $\{\lambda\mathtt{sub},\mathtt{u}\}/\mathtt{P}$-reduction starting at t. Since $t \in \mathcal{SN}_{\{\lambda\mathtt{sub},\mathtt{u}\}/\mathtt{P}}$ by Cor. 5, then also $\mathtt{dB}(t) \in \mathcal{SN}_{\{\lambda\mathtt{sub},\mathtt{u}\}/\mathtt{P}}$, so we get a contradiction.

The permutative λ-calculus can be (strictly) simulated into the reduction relation $\{\lambda\mathtt{sub},\mathtt{u},\hat{\mathtt{u}}\}/\Pi$ and thus it enjoys PSN.

Corollary 5 (PSN for $\{\beta,\hat{\mathtt{u}}\}/\hat{\mathtt{P}}$). *The permutative λ-calculus enjoys PSN, i.e. if $t \in \mathcal{SN}_\beta$, then $t \in \mathcal{SN}_{\{\beta,\hat{\mathtt{u}}\}/\hat{\mathtt{P}}}$.*

This last corollary is a generalisation of René David's results [4], where $\hat{\sigma}_1$ is taken from left to right while $\{\hat{\sigma}_2, \widehat{\mathtt{box}}\}$ are taken from right to left.

More generally, consider any orientation of (a subset) of our equations that yields a terminating reduction $\rightsquigarrow$. Then, the system where these equations are replaced by $\rightsquigarrow$ turns out to enjoy PSN. Thus, our result strictly subsumes previous results in the literature [4, 7].

To appreciate the power of Cor. 5 note that whenever t is typable with respect to a system $\mathcal{S}$ guaranteing β-strong normalisation (for instance, simple types, intersection types, second-order types) then t is strongly normalising (SN) in $\Lambda_{\hat{\mathtt{P}}}$.

Theorem 6 (SN). *Typability implies strong normalisation.*

8 Conclusions and future work

This paper proposes the permutative λ-calculus as a natural generalization of existing λ-calculi for reasoning about permutation of constructors. In all these frameworks permutations are reduction rules, while in $\Lambda_{\hat{\mathtt{P}}}$ they are treated as equations, which is more general and also requires more sophisticated rewriting techniques. The one we use for confluence, based on the new notion of $\mathtt{M}$-developments, is simple and yet powerful: we believe it is interesting by itself. We think that $\mathtt{M}$-developments can also be used for proving meta-confluence of $\Lambda_{\hat{\mathtt{P}}}$.

It would be also interesting to understand if it is possible to state some abstract conditions on *equational* extensions of λ-calculus implying the good behaviour of equations. Indeed, [7] gives sufficient conditions on *reduction systems* extending λ-calculus to guarantee that they enjoy PSN. However, the method in [7] does not seem to be naturally applicable to equational extensions.

References

1. B. Accattoli and D. Kesner. The structural lambda-calculus. In CSL, LNCS 6247, Springer, 2010.
2. B. Accattoli and D. Kesner. Preservation of strong normalisation modulo permutations for the structural calculus, 2011. Submitted to LMCS, available at `https://sites.google.com/site/beniaminoaccattoli/PSN-modulo.pdf`.
3. H. Barendregt. The Lambda Calculus: Its Syntax and Semantics. North-Holland, 1984. Revised Edition.
4. R. David. A short proof that adding some permutation rules to preserves SN. TCS, 412(11):1022–1026, 2011.
5. P. de Groote. The conservation theorem revisited. In TLCA, LNCS 664, pages 163–178. Springer, 1993.
6. J. Espírito Santo. Delayed Substitutions. In RTA, LNCS 4533, pages 169–183. Springer, 2007.
7. J. Espírito Santo. A note on preservation of strong normalisation in the λ-calculus. TCS, 412(11):1027–1032, 2011.
8. J.-Y. Girard. Linear logic. TCS, 50, 1987.
9. J. R. Hindley. Reductions of residuals are finite. Transactions of the American Mathematical Society, 240:345–361, 1978.
10. F. Kamareddine. Postponement, conservation and preservation of strong normalization for generalized reduction. JLC, 10(5):721–738, 2000.
11. D. Kesner. A theory of explicit substitutions with safe and full composition. LMCS, 5(3:1):1–29, 2009.
12. A. J. Kfoury and J. B. Wells. New notions of reduction and non-semantic proofs of beta-strong normalization in typed lambda-calculi. In LICS, pages 311–321. IEEE Computer Society Press, 1995.
13. J.-W. Klop, V. van Oostrom, and F. van Raamsdonk. Combinatory reduction systems: introduction and survey. TCS, 121(1/2):279–308, 1993.
14. S. Lengrand. Termination of lambda-calculus with the extra call-by-value rule known as assoc. CoRR, abs/0806.4859, 2008.
15. J.-J. Lévy. Réductions correctes et optimales dans le lambda-calcul. PhD thesis, Univ. Paris VII, France, 1978.
16. E. Moggi. Computational lambda-calculus and monads. In LICS, pages 14–23. IEEE Computer Society Press, 1989.
17. L. Regnier. Une équivalence sur les lambda-termes. TCS, 2(126):281–292, 1994.
18. A. Sabry and M. Felleisen. Reasoning about programs in continuation-passing style. In LFP, pages 288–298. ACM, New York, USA, 1992.
19. Terese. Term Rewriting Systems, volume 55 of Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2003.
20. V. van Oostrom. Z. Slides available on `http://www.phil.uu.nl/~oostrom/publication/rewriting.html`.
21. J. Espírito Santo., R. Matthes and L. Pinto. Monadic Translation of Intuitionistic Sequent Calculus. in Types for Proofs and Programs, Springer-Verlag, 2009.