Metaconfluence of λj: dealing with non-deterministic replacements

Fabien Renaud

PPS, CNRS and University Paris 7-Paris Diderot

Abstract. This paper focuses on the λj-calculus, a formalism with *jumps* inspired from linear logic, and based on the notion of multiplicity. We consider terms with metavariables, and we prove metaconfluence of the calculus equipped with an equivalence relation. This confers to λj all the good properties that one can expect.

1 Introduction

The structural [1] calculus λj [AK10] corresponds to a lambda-calculus with *jumps* which constitutes a term approach of a recent alternative [AG09] to proof-nets [Gir87]. The λj-calculus brings three main novelties which confers originality:

- Usually, calculi decomposing the beta reduction such as calculi with explicit substitutions are based on the notion of structure, that is substitutions go through the structure of terms until variables are reached. The rules of these systems are basically designed as being a case analysis on which term is at the left of the substitution. Here, rules are only defined by case analysis on the number of *occurrences* of the variable bound by the substitutions. There exists other calculi specified in this way, but they also incorporate an analysis on met constructors [KR09].
- When some substitution bounds only one occurrence of a variable, action at distance is performed, that is the usual meta-level substitution is triggered. This is why the term jump is used instead of explicit substitutions. Despite the use of an *implicit* operator for substitution, λj still belongs to the family of calculi with explicit substitutions. Indeed, when λj is implemented with graphs, the substitution of a unique occurrence is an atomic operation which does not require a global side-condition.
- The last one, which will make more complex our work, is the use of a non-deterministic replacement to specify contraction, an operation once again motivated by proof-nets.

The rewriting system of λj consists of only four rules, and has all the good properties that one can expect: confluence on terms, strong normalization of

[1] the notion of structure is the one of structural rules in logic (contraction, weakening), not the one of term structure

typed terms, preservation of the β-strong normalization, simulation of the λ-calculus, and full composition.

In this paper we study metaconfluence, which is stronger than confluence since it concerns metaterms i.e. terms containing metavariables [2] which represent incomplete proofs/programs. Metaterms can be used for instance to perform higher-order unification [DHKP96,Hue76] or implement metalanguages [Nad02]. Metaconfluence can be understood as confluence on terms enjoying some locality property in the sense that it is not necessary to reduce deeply in the term to find a common reduct between two terms having a common origin. This is not the case for calculi enjoying confluence but not metaconfluence [ACCL91,Blo97,SFM03].

To prove metaconfluence we combine the interpretation method [Har89] with ideas from Tait-Martin Löf's technique [Bar84]. However, before being able to apply those standard methods, we need to specify some subtle properties of the non-deterministic replacement used in the specification of the operational semantics of $\lambda\mathtt{j}$. As the equivalence (sub)relation is a strong bisimulation, we will be able to extend metaconfluence to the stronger Church-Rosser property.

While the standard method for λ-calculus and associated calculi is the one from Tait-Martin Löf, it is the interpretation method, which seems to be the simplest in this field. Indeed, one can benefit from the fact that almost all reduction relations of explicit substitutions calculi can be split into a rule creating substitutions and others propagating them. Furthermore, it is often easy to show that this last set of rules is confluent and terminating. When dealing with terms, one can show confluence of calculi with explicit substitutions thanks to the confluence of the λ-calculus [ACCL91,KR97], while with metavariables, it is necessary to combine the notion of simultaneous reduction introduced by Tait and Martin Löf to obtain the diamond property of the interpretation calculus [KR95,Kes07].

The result will immediatly extend to the case where other equivalences defining strong bisimulations on $\lambda\mathtt{j}$ metaterms are added, such as the σ-equivalence of Regnier [Reg91]. Furthermore, our result implies metaconfluence on $\lambda\mathtt{j}$-dags, the graph formalism from which $\lambda\mathtt{j}$ is inspired, since there is a strong bisimulation between those two systems.

Related works: Other techniques exist to show confluence of calculi with explicit substitutions such as the Yokouchi-Hikita [YH90] method, used for example for $\lambda\sigma_{\Uparrow}$ [CHL96] and $\lambda\mathtt{x}$ [RBL09]; the Z-technique [vO08b], used for example for $\lambda\mathtt{ex}$ [Kes09]; the technique of decreasing diagrams [vO08a] used for $\lambda\mathtt{x}||\mathtt{c}$ [Blo97].

Section 2 introduces the $\lambda\mathtt{j}$-calculus and its properties; Section 3 defines notations and basic lemmas needed to handle the non-deterministic notion of replacement ; Section 4 presents the proof of metaconfluence strictly speaking; Section 4.1 generalizes metaconfluence to the Church-Rosser property, and Section 5 concludes and presents future directions of work.

[2] the terms *instantiatable variable* and *open term* can be found in the literature instead of metavariable and metaterm

2 The meta λj-calculus and some basic properties

We extend the grammar of λj with metavariables $\mathbb{X}_\Delta$ in order to denote incomplete programs. To soundly instantiate metavariables with terms, the set of variables Δ indexing $\mathbb{X}$ denote the set of variables which are free in it. Thus $\mathbb{X}_\Delta$ is a term whose free variables are Δ.

Definition 1 (Metaterms). *Terms of the calculus are defined by the following grammar:* $t, u ::= x \mid \lambda x.t \mid t\ u \mid t[x{:=}u] \mid \mathbb{X}_\Delta$

The term x is a *v*ariable, $\lambda x.t$ an *a*bstraction, $t\ u$ an application, $t[x := u]$ a jump, $\mathbb{X}_\Delta$ a metavariable. In the abstraction and the jump, the variable x is bound in t. We will follow Barendregt's convention [Bar84] to avoid free variable capture, and consider terms modulo α-conversion.

The metaterm $y[x := v]\ \mathbb{X}_{y,z}$ can be instantiated for example by the term $y[x{:=}v]\ y\ z$ or $y[x{:=}v]\ ((\lambda x.xz)[w{:=}y])$.

Definition 2 (Variables and Occurences). *The set of all free variables of* t *is written* $\mathtt{fv}(t)$ *and defined by* $\mathtt{fv}(\mathbb{X}_\Delta) = \Delta$ *for metavariables; as expected for the other constructors. We write* $\mathtt{fmvar}(t)$ *for the set of all the free variables of all the metavariables of* t.

The notation $x \in_n \mathtt{fv}(t)$ means that x appears free n times in t (counting one in the case of $\mathbb{X}_\Delta$ with $x \in \Delta$). Similarly, $x \in_n \mathtt{fmvar}(t)$ means that x appears n times in the metavariables of t.

A list of jumps $[x_1 := u_1]...[x_n := u_n]$ is abbreviated as $\overline{[x{:=}u]}$ or simply $\mathtt{L}$ if the names of the variables and terms are not important. The list of substitutions $[x_1{:=}u]...[x_n{:=}u]$ can also be abbreviated as $[x_1, ..., x_n{:=}u]$.

Definition 3 (Non-deterministic replacement). *When* $x \in_n \mathtt{fv}(t)$ *with* $n \geq 2$*, we write* $t_{x \rightsquigarrow y}$ *for the non-deterministic replacement of* i *(*$1 \leq i \leq n-1$*) occurrences of* x *by* y.

For instance, the term $(x[z{:=}\mathbb{X}_x])_{x \rightsquigarrow y}$ denotes either $y[z{:=}\mathbb{X}_x]$ or $x[z{:=}\mathbb{X}_y]$ and $x\ \mathbb{X}_x[z{:=}\mathbb{Z}_x]_{x \rightsquigarrow \{y_1, y_2\}}$ denotes either $x\ \mathbb{X}_{y_1}[z{:=}\mathbb{Z}_{y_2}]$ or $x\ \mathbb{X}_{y_2}[z{:=}\mathbb{Z}_{y_1}]$.

Definition 4 (Implicit substitution). *The implicit substitution* $t\{x := u\}$ *is a meta-level operation yielding the term* t *where all the free occurrences of the variable* x *have been substituted by* u*. This operation is defined by induction on* t only *when* $x \notin \mathtt{fmvar}(t)$ [3].

$$
\begin{array}{lll}
x\{x{:=}u\} & := u & \\
y\{x{:=}u\} & := y & \\
(\lambda y.v)\{x{:=}u\} & := \lambda y.v\{x{:=}u\} & x \neq y \\
(v\ w)\{x{:=}u\} & := v\{x{:=}u\}\ w\{x{:=}u\} & \\
v[y{:=}w]\{x{:=}u\} & := v\{x{:=}u\}[y{:=}w\{x{:=}u\}] & x \neq y
\end{array}
$$

[3] It makes no sense to define the implicit substitution when x appears in a metavariable because by definition, we do not know what would be the result since the structure of the metavariable is unknown

Considering implicit substitution in the context of metavariables, other approaches exist, consisting of changing back $\mathbb{X}_\Delta\{x{:=}u\}$ into $\mathbb{X}_\Delta[x{:=}u]$ even when x appears in a metavariable [Kes07]. This makes an important difference because doing so, one can move the implicit substitution downward the term and then finally transform it in a jump. The implicit substitution thus switches from a static to a dynamic status which is not that what one expects.

Implicit substitution enjoys the following well-known property:

Lemma 1. *Let* $x, y \notin \mathtt{fmvar}(t) \cup \mathtt{fmvar}(u)$. *Then,* $t\{x := u\}\{y := v\} = t\{y := v\}\{x{:=}u\{y{:=}v\}\}$.

Proof. By induction on t.

We can now redefine rules of [AK10] in order to deal with metaterms as follows:

Definition 5 ($\lambda\mathtt{j}$ reduction rules).

$$
\begin{array}{llll}
(\lambda x.t)\mathtt{L}\ u & \mapsto_{\mathtt{dB}} & t[x{:=}u]\mathtt{L} & \\
t[x{:=}u] & \mapsto_{\mathtt{w}} & t & \text{if } x \in_0 \mathtt{fv}(t) \\
t[x{:=}u] & \mapsto_{\mathtt{d}} & t\{x{:=}u\} & \text{if } x \in_1 \mathtt{fv}(t) \text{ and } x \notin \mathtt{fmvar}(t) \\
t[x{:=}u] & \mapsto_{\mathtt{c}} & t_{x \rightsquigarrow y}[x{:=}u][y{:=}u] & \text{if } x \in_n \mathtt{fv}(t) \text{ with } n > 1 \\
t[x{:=}u][y{:=}v] & \sim & t[y{:=}v][x{:=}u] & x \notin \mathtt{fv}(v)\ \&\ y \notin \mathtt{fv}(u)
\end{array}
$$

The reduction relation $\to_{\mathtt{j}}$ (resp. $\to_{\mathtt{dB}}$) is defined to be the contextual closure of the rules $\mapsto_{\mathtt{w}}, \mapsto_{\mathtt{d}}$ and $\mapsto_{\mathtt{c}}$ (resp. $\mapsto_{\mathtt{dB}}$). The equivalence relation $\equiv$ is the contextual closure of $\sim$ and $\equiv^*$ (resp. $\to_R^*$) is the transitive and reflexive closure of $\equiv$ (resp. of any reduction relation $\to_R$). For sake of lightness, we will write $\equiv^*$ instead of $\equiv$. The reduction relation $\to_{\lambda\mathtt{j}}$ is $\to_{\mathtt{dB}} \cup \to_{\mathtt{j}}$. For any reduction relation $\mathtt{x}$, $t \to_{\mathtt{x}/\equiv} t'$ if $t \equiv u \to_{\mathtt{x}} u' \equiv t'$; $t \to^*_{\mathtt{x}/\equiv} t'$ if $t \to_{\mathtt{x}/\equiv} \cdot \to_{\mathtt{x}/\equiv} \ldots t'$ or $t \equiv t'$. A term t irreducible with a reduction $\mathtt{x}$ is said to be in normal form and written $t \in \mathcal{NF}_{\mathtt{x}}$.

For example, the reduction of the term $(x\ x)[x{:=}y]$ gives either $(x\ x)[x{:=}y] \to_{\mathtt{c}} (x_1\ x_2)[x_1{:=}y][x_2{:=}y] \to^*_{\mathtt{d}} y\ y$ or $(x\ x)[x{:=}y] \to_{\mathtt{c}} (x_2\ x_1)[x_1{:=}y][x_2{:=}y]$. Reduction of the metaterm $((\lambda z.x\ z)\ \mathbb{X}_{\{x\}})[x := y]$ gives $((\lambda z.x\ z)\ \mathbb{X}_{\{x\}})[x := y] \to_{\mathtt{dB}} (x\ z)[z := \mathbb{X}_{\{x\}}][x := y] \to_{\mathtt{d}} (x\ \mathbb{X}_{\{x\}})[x := y] \to_{\mathtt{c}} (x_1\ \mathbb{X}_{\{x_2\}})[x_1 := y][x_2 := y] \to_{\mathtt{d}} (y\ \mathbb{X}_{\{x_2\}})[x_2{:=}y]$ with the last metaterm in normal form.

Notice that rule $\mathtt{d}$ is stated in the original version of $\lambda\mathtt{j}$ (defined only on terms) by "$t[x{:=}u] \to_{\mathtt{d}} t\{x{:=}u\}$ if $x \in_1 \mathtt{fv}(t)$". In order to be consistent with Definition 4 of implicit substitution we adopt here a slight variation. As stated before, the propagation rules are defined by case analysis on the number of occurrences of the bound variable. The letters in the rules stand for weakening, derelicition, and contraction. $\mathtt{dB}$ is the rule $\mathtt{B}$ which is the usual name for the creation of jumps at a distance. The rule $\mathtt{c}$ should be considered as rules schema. It does not correspond to a single rule but to as many possible partitions between x and y exist. In particular, one can choose a different partition for each possible $\mathtt{c}$-reduction. Of course, when implementing the calculus, one should specify a particular strategy to partition variables. However, since we work with the

calculus and not with a particular implementation of it, we will never specify it in the sequel.

The $\lambda\mathtt{j}$-calculus is firstly presented in [AK10] without $\equiv$ since the four rules are enough for it to enjoy almost all properties including confluence on terms. The equivalence relation is however needed to close the following critical pair:

$$\begin{array}{ccccc} (\mathbb{X}_x\ \mathbb{Z}_y)[x{:=}u][y{:=}u] & \leftarrow_{\mathtt{c}} & (\mathbb{X}_x\ \mathbb{Z}_x)[x{:=}u] & \rightarrow_{\mathtt{c}} & (\mathbb{X}_y\ \mathbb{Z}_x)[x{:=}u][y{:=}u] \\ \equiv & & & & =_\alpha \\ & & (\mathbb{X}_x\ \mathbb{Z}_y)[y{:=}u][x{:=}u] & & \end{array}$$

Lemma 2 (Full composition). *The $\lambda\mathtt{j}$-calculus enjoys full composition on terms [AK10] and metaterms i.e. if $x \notin \mathtt{fmvar}(t)$, then $t[x{:=}u] \rightarrow^*_{\lambda\mathtt{j}/\equiv} t\{x{:=}u\}$.*

Proof. By induction on the number of occurrences of x in t.

Definition 6. *Let $\rightarrow$ be a rewriting system and $\equiv$ an equivalence relation. Then,*

1. *$\rightarrow$ is confluent if ${}^*\!\leftarrow \cdot \rightarrow^* \subseteq \rightarrow^* \cdot {}^*\!\leftarrow$*
2. *$\rightarrow$ is confluent modulo $\equiv$ if ${}^*\!\leftarrow \cdot \equiv \cdot \rightarrow^* \subseteq \rightarrow^* \cdot \equiv \cdot {}^*\!\leftarrow$*
3. *$\rightarrow$ is Church-Rosser (CR) modulo $\equiv$ if $\leftrightarrow^*_\equiv \subseteq \rightarrow^* \cdot \equiv \cdot {}^*\!\leftarrow$ where $\leftrightarrow^*_\equiv := (\rightarrow \cup \leftarrow \cup \equiv)^*$*

Lemma 3. *CR modulo $\equiv$ is the stronger property i.e. $3 \Rightarrow 2$ and $3 \Rightarrow 1$ but $1 \not\Rightarrow 3$ and $2 \not\Rightarrow 3$.*

Proof. Cf. [Ter03].

Lemma 4. *The reduction $\rightarrow_{\mathtt{j}/\equiv}$ is terminating and confluent on metaterms. Thus for any term t, there is one unique normal form (up to $\equiv$) written $\downarrow(t)$.*

Proof. For the termination property, the proof of [AK10] is modified by extending the notion of *potential multiplicity* to metavariables with $M_x(\mathbb{X}_\Delta) = 1$ if $x \in \Delta$, 0 otherwise ; and by adding the case of the equation.

For confluence, all critical pairs are easily joinable, and we can conclude by Newman's lemma.

In general, all the proofs for $\lambda\mathtt{j}$ are the same when extending to metaterms since the kind (meta or not) of an occurrence is stable by reduction and substitution.

Lemma 5. *Metaterms in $\mathcal{NF}_\mathtt{j}$ are of the following shape with $u, v \in \mathcal{NF}_\mathtt{j}$:*

$$x \mid \mathbb{X}_\Delta \mid \lambda x.u \mid u\ v \mid u[x{:=}v] \text{ with } x \in_1 \mathtt{fmvar}(u) \text{ and } x \in_1 \mathtt{fv}(u)$$

For instance $(\mathbb{X}_x\ y)[x{:=}u]$ is in $\mathtt{j}$-normal form but $(\mathbb{X}_x\ x)[x{:=}u]$ is not.

Lemma 6. *If $t \rightarrow^*_{\lambda\mathtt{j}} t'$ then $\mathtt{fmvar}(t') \subseteq \mathtt{fmvar}(t)$.*

Proof. By induction of the rules of $\lambda\mathtt{j}$.

Lemma 6 will be used implicitly for cases where we have $t[y{:=}u] \to^*_{\lambda\mathtt{j}} t'[y{:=}u]$ with $x \notin \mathtt{fmvar}(t)$ and want to perform the rule $t'[y{:=}u] \to_{\mathtt{d}} t'\{y{:=}u\}$. We need to have $x \notin \mathtt{fmvar}(t')$, which is true by the lemma.

Lemma 7. *Let t be a metaterm and x a variable such that $x \notin \mathtt{fmvar}(t)$ Then, several properties from $\lambda\mathtt{j}$ on simple terms hold:*

1. *If $t \to^*_{\lambda\mathtt{j}/\equiv} t'$, $u \to^*_{\lambda\mathtt{j}/\equiv} u'$ then $t\{x{:=}u\} \to^*_{\lambda\mathtt{j}/\equiv} t'\{x{:=}u'\}$. In particular, if $t \equiv t'$ and $u \equiv u'$ then $t\{x{:=}u\} \equiv t'\{x{:=}u'\}$.*
2. *Let t, u be metaterms in $\mathtt{j}$-normal form. Then $t\{x{:=}u\}$ is also in $\mathtt{j}$-normal form.*

Lemma 8. *If $\downarrow((\lambda x.v)\mathtt{L}\ u) = (\lambda x.v')\mathtt{L}'\ u'$ then $\downarrow(v[x{:=}u]\mathtt{L}) =\downarrow(v'[x{:=}u']\mathtt{L}')$*

Proof. We first notice that u' is the normal form of u. Then we can notice that v' in $(\lambda x.v')\mathtt{L}'u'$ is the normal form of v with substitutions and renamings only determined by $\mathtt{L}$. In the same way, $\mathtt{L}'$ is determined by $\mathtt{L}$ and the number of occurrences of variables it binds in v. To conclude, we remark that the normals forms of the subterms of $\downarrow(v[x{:=}u]\mathtt{L})$ are determined in the same way.

3 Non-deterministic replacement

We state several properties dealing with the most subtle aspect of $\lambda\mathtt{j}$, the non-deterministic replacement. The aim is to fully expand substitutions with respect to the rules of $\mathtt{j}$, thus obtaining a term which is renaming-independent. In this section we will thus characterize $\downarrow(t[x := u])$ (with t, u in $\mathtt{j}$-normal form) as $t_{x\rightsquigarrow y_1,\ldots,y_n}[y_1 := u]\ldots[y_n := u]\{x := u\}$ where this last term is t where all free occurences of x in a metavariable have been renamed into different fresh variables $y_1, \ldots, y_n$ with an explicit substitution introduced for every y_i, anf finally an implicit substitution to deal with occurences of x not in metavariables. The main goal in this section will be to show that if $t \to^*_{\lambda\mathtt{j}/\equiv} t'$ and $u \to^*_{\lambda\mathtt{j}/\equiv} u'$ then $t_{x\rightsquigarrow y_1,\ldots,y_n}[y_1{:=}u]\ldots[y_n{:=}u] \to^*_{\lambda\mathtt{j}/\equiv} t'_{x\rightsquigarrow z_1,\ldots,z_m}[y_1{:=}u']\ldots[y_n{:=}u']$. This is necessary to show that the simultaneous reduction is included in $\to_{\lambda\mathtt{j}}$ (point 1. of Lemma 16).

All lemmas are completely independent from the confluence section and could be used for other studies of $\lambda\mathtt{j}$.

Definition 7 (Position). *A position in a term t describes a subterm in t by means of a sequence (eventually empty) of 0 and 1. If t is either $u\ v$, or $\lambda y.u$ or $u[y{:=}v]$ then the subterm at the position 0 is u and that at the position 1 is v (except for the abstraction where it is not defined). For instance, the subterm at position 010 in $y[z{:=}\lambda x.v]\ v$ is v.*

Definition 8. *Let ρ be an injective function from positions of $x \in_n \mathtt{fmvar}(t)$ to a set of fresh variables $\{y_1, \ldots, y_n\}$. Then $t_{\rho:x\rightsquigarrow\{y_1,\ldots,y_n\}}$ denotes the meta-level operation which applies ρ to t. All occurrences of x which do not appear in*

metavariables of t are not modified. The set $\{y_1, ..., y_n\}$ is the image of ρ, written $\mathtt{img}(\rho)$. Notice that the operation ρ is deterministic.

We also need a variant, which still renames all occurrences of x in metavariables, but which is not injective: it is denoted by $t_{\rho:x\rightarrowtail\{y_1,...,y_n\}}$.

A renaming $\rho : x$ is valid w.r.t. a term t if for every position defined by the mapping there is a variable x in a metavariable in t.

In the sequel, we will only consider valid renamings w.r.t. the terms in which they are applied. We will simply write $t_{\rho:x}$ instead of $t_{\rho:x\rightsquigarrow\{y_1,...,y_n\}}$ or $t_{\rho:x\rightarrowtail\{y_1,...,y_n\}}$ when names of fresh variables are irrelevant or clear from the context. The metaterm $t_{\rho:x\rightsquigarrow\{y_1,...,y_n\}}[y_1:=u]...[y_n:=u]$ is abbreviated as $t_{\rho:x,u}$. If $x \in_0 \mathtt{fmvar}(t)$ then $t_{\rho:x,u}$ should be read as t.

For instance, consider the metaterm $\mathbb{Z}_x[y{:=}\mathbb{X}_x]_{\rho:x\rightsquigarrow y_1,y_2}$ where ρ maps the occurrence of x at the position 0 to y_1 and the occurrence of x at the position 1 to y_2 is exactly $\mathbb{Z}_{y_1}[y{:=}\mathbb{X}_{y_2}]$. Taking the same ρ, the metaterm $\mathbb{Z}_x[z{:=}\mathbb{X}_x]_{\rho:x,u}$ is $\mathbb{Z}_{y_1}[z{:=}\mathbb{X}_{y_2}][y_1{:=}u][y_2{:=}u]$.

The metaterm $(x\ \mathbb{X}_x)\ \mathbb{X}_x$ affected by the non injective renaming ρ associating all occurrences of x in metavariables to y is: $((x\ \mathbb{X}_x)\ \mathbb{X}_x)_{\rho:x\rightarrowtail y} = (x\ \mathbb{X}_y)\ \mathbb{X}_y$.

The notation $t_{\rho:x,u}$ reflects our methodology: rename all occurrences of x in metavariables to benefit from $\equiv$ which makes the renaming ρ irrelevant. Non injective renamings will be used when talking about metaterms in which renamed variables are not bound by independent, and thus permutable, substitutions.

Lemma 9. *The following assertions are true regardless the injectivity of the renamings:*

- $\forall\rho, \exists!\rho_v, \rho_w$ *s.t.* $(v\ w)_{\rho:x} = v_{\rho_v:x}\ v_{\rho_w:x}$
- $\forall\rho_v, \rho_w, \exists!\rho$ *s.t.* $(v\ w)_{\rho:x} = v_{\rho_v:x}\ w_{\rho_w:x}$.
- $\forall\rho, \exists!\rho_v$ *s.t.* $(\lambda y.v)_{\rho:x} = \lambda y.v_{\rho_v:x}$
- $\forall\rho_v, \exists!\rho$ *s.t.* $(\lambda y.v)_{\rho:x} = \lambda y.v_{\rho_v:x}$
- $\forall\rho, \exists!\rho_v, \rho_w$ *s.t.* $v[y{:=}w]_{\rho:x} = v_{\rho_v:x}[y{:=}w_{\rho_w:x}]$
- $\forall\rho_v, \rho_w, \exists!\rho$ *s.t.* $v[y{:=}w]_{\rho:x} = v_{\rho_v:x}[y{:=}w_{\rho_w:x}]$
- $\forall\rho, \exists!\rho_v, \rho_w$ *s.t.* $v\{y{:=}w\}_{\rho:x} = v_{\rho_v:x}\{y{:=}w_{\rho_w:x}\}$ *if* $y \in_1 \mathtt{fv}(v)$
- $\forall\rho_v, \rho_w, \exists!\rho$ *s.t.* $v\{y{:=}w\}_{\rho:x} = v_{\rho_v:x}\{y{:=}w_{\rho_w:x}\}$ *if* $y \in_1 \mathtt{fv}(v)$

Furthermore we have $\mathtt{img}(\rho) = \mathtt{img}(\rho_v) \cup \mathtt{img}(\rho_w)$ *for all cases except the abstraction case where* $\mathtt{img}(\rho) = \mathtt{img}(\rho_v)$.

Proof. We only show the first case as the others are similar. For each mapping like $\rho(0p) = y_i$ (resp. $\rho(1p) = y_i$) with p a position, we construct ρ_v (resp. ρ_w) like $\rho_v(p) = y_i$ (resp. $\rho_w(p) = y_i$).

For instance, consider the metaterm $\mathbb{Z}_x[y{:=}\mathbb{X}_x]_{\rho:x\rightsquigarrow y_1,y_2}$ with a renaming ρ associating 0 to y_1 and 1 to y_2, then there exists two renamings ρ_1 and ρ_2 (which respectively associate 0 to y_1 and 0 to y_2) such that $\mathbb{Z}_x[y{:=}\mathbb{X}_x]_{\rho:x\rightsquigarrow y_1,y_2} = \mathbb{Z}_{y_1}[y{:=}\mathbb{X}_{y_2}] = \mathbb{Z}_{x\,\rho_1:x\rightsquigarrow y_1}[y{:=}\mathbb{X}_{x\,\rho_2:x\rightsquigarrow y_2}]$.

Lemma 10. *For any metaterm t and renaming ρ, there exists unique $\rho', z_1, ..., z_m$ such that* $\downarrow(t_{\rho:x\rightarrowtail\{y_1,...,y_n\}}) = \downarrow(t)_{\rho':x\rightarrowtail\{z_1,...z_m\}}$ *with* $\mathtt{img}(\rho') \subseteq \mathtt{img}(\rho)$.

Proof. By induction on t using Lemma 9.

Example 1. Consider the metaterm $(z\ z)[z{:=}\mathbb{X}_x]$ with the particular case where the renaming ρ is injective such that $(z\ z)[z{:=}\mathbb{X}_x]_{\rho:x\rightarrowtail y} = (z\ z)[z{:=}\mathbb{X}_y]$. Thus $\downarrow((z\ z)[z{:=}\mathbb{X}_y]) = (z_1\ z_2)[z{:=}\mathbb{X}_y][z_2{:=}\mathbb{X}_y] = (z_1\ z_2)[z{:=}\mathbb{X}_x][z_2{:=}\mathbb{X}_x]_{\rho':x\rightarrowtail\{y\}} = \downarrow((z\ z)[z{:=}\mathbb{X}_x])_{\rho':x\rightarrowtail\{y\}}$ with ρ' being the renaming associating to each occurrence of x the variable y.

We can now state a technical property of the paper which let us deal with the non-deterministic replacement in a "deterministic" way.

Property 1. *Let t, u be metaterms in* `j` *normal form. Then, for any ρ we have $\downarrow(t[x{:=}u]) \equiv t_{\rho:x,u}\{x{:=}u\}$ and thus for any ρ, ρ', $t_{\rho:x,u} \equiv t_{\rho':x,u}$.*

Proof. If $x \in_0 \mathtt{fv}(t)$ or $x \in_1 \mathtt{fv}(t)$ then this is straightforward. Otherwise, $t[x{:=}u]$ reduces in many `c`-steps to $t_{\rho:x\rightsquigarrow\{y_1\ldots y_n\}}[x{:=}u][y_1{:=}u]\ldots[y_n{:=}u]$ for any ρ. We can apply Full Composition and obtain $t_{\rho:x,u}\{x{:=}u\}$ which is in `j`-normal form since $\forall i.y_i \in_1 \mathtt{fmvar}(t)$ and because implicit substitution preserves `j`-normal form by Lemma 7:2.

This property implies that the non-determinism is irrelevant thanks to substitutions permutation. We can easily generalize to the case where subterms are not in `j`-normal form:

Corollary 1. *Let t, u be metaterms. Then, for any ρ we have $\downarrow(t[x := u]) \equiv \downarrow(t)_{\rho:x,\downarrow(u)}\{x{:=}\downarrow(u)\}$ and thus for any ρ, ρ', $\downarrow(t)_{\rho:x,\downarrow(u)} \equiv \downarrow(t)_{\rho':x,\downarrow(u)}$.*

Lemma 11. *If $t \equiv t'$ then for any ρ, there exists a unique ρ' s.t. $t_{\rho:x} \equiv {t'}_{\rho':x}$*

Proof. We first show that if $t \equiv t'$ then for any ρ, there exists a unique ρ' such that $t_{\rho:x} \equiv {t'}_{\rho':x}$. If $t = v[y{:=}w][z{:=}u] \equiv v[z{:=}u][y{:=}w] = t'$, we have unique ρ_v, ρ_u and ρ_w s.t $t_{\rho:x} =_{L.\ 9} v_{\rho_v:x}[y := w_{\rho_w:x}][z := u_{\rho_u:x}] \equiv v_{\rho_v:x}[z := u_{\rho_u:x}][y := w_{\rho_w:x}] =_{L.\ 9} {t'}_{\rho':x}$. The inductive cases are straightforward. The general case when $t \equiv t'$ is done by induction.

Notice that the Lemma 11 is not true for any ρ'. For instance, take the reflexive case where $t = t' = \mathbb{X}_x\ \mathbb{X}_x$. Two different assignments for t are $\mathbb{X}_{x_1}\ \mathbb{X}_{x_2}$ and $\mathbb{X}_{x_2}\ \mathbb{X}_{x_1}$, but they are not equivalent modulo $\equiv$.

Lemma 12. *If $u \rightarrow^*_{\lambda\mathtt{j}/\equiv} u'$ then $t_{\rho:x,u} \rightarrow^*_{\lambda\mathtt{j}/\equiv} t_{\rho:x,u'}$. In particular, if $u \equiv u'$ then $t_{\rho:x,u} \equiv t_{\rho:x,u'}$.*

Proof. Straightforward.

Lemma 13. *If $t \equiv t'$, $u \equiv u'$ then for any ρ, ρ', $t_{\rho:x,u} \equiv {t'}_{\rho':x,u'}$. Furthermore, $t_{\rho:x,u}\{x{:=}u\} \equiv {t'}_{\rho':x,u'}\{x{:=}u'\}$.*

Proof. The second part is obvious from the first by Lemma 7:1. For the first, by Lemma 11, there exists a unique ρ'' s.t. $t_{\rho:x} \equiv {t'}_{\rho'':x}$. As $\equiv$ is closed by context, we have $t_{\rho:x,u} \equiv {t'}_{\rho'':x,u}$ and ${t'}_{\rho'':x,u} \equiv {t'}_{\rho'':x,u'}$ by Lemma 12. Property 1 concludes with ${t'}_{\rho'':x,u'} \equiv {t'}_{\rho':x,u'}$.

We illustrate the first case by simply considering $u = u'$. Take for instance $t = v[z_1 := \mathbb{X}_x][z_2 := \mathbb{Z}_x]$ and consider the renamed metaterm $t_{\rho:x} = v[z_1 := \mathbb{X}_{x_1}][z_2{:=}\mathbb{Z}_{x_2}]$. Then for $t' = v[z_2{:=}\mathbb{Z}_x][z_1{:=}\mathbb{X}_x]$ we have:

– if $t_{\rho':x} = v[z_2{:=}\mathbb{Z}_{x_1}][z_1{:=}\mathbb{X}_{x_2}]$ then

$$\begin{array}{ll} t_{\rho:x,u} & = \\ v[z_1{:=}\mathbb{X}_{x_1}][z_2{:=}\mathbb{Z}_{x_2}][x_1{:=}u][x_2{:=}u] & \equiv \\ v[z_2{:=}\mathbb{Z}_{x_2}][z_1{:=}\mathbb{X}_{x_1}][x_1{:=}u][x_2{:=}u] & \equiv \\ v[z_2{:=}\mathbb{Z}_{x_2}][z_1{:=}\mathbb{X}_{x_1}][x_2{:=}u][x_1{:=}u] & =_\alpha \\ v[z_2{:=}\mathbb{Z}_{x_1}][z_1{:=}\mathbb{X}_{x_2}][x_1{:=}u][x_2{:=}u] & = {t'}_{\rho':x,u} \end{array}$$

– if ${t'}_{\rho'':x} = v[z_2{:=}\mathbb{Z}_{x_2}][z_1{:=}\mathbb{X}_{x_1}]$ then

$$\begin{array}{ll} t_{\rho:x,u} & = \\ v[z_1{:=}\mathbb{X}_{x_1}][z_2{:=}\mathbb{Z}_{x_2}][x_1{:=}u][x_2{:=}u] & \equiv \\ v[z_2{:=}\mathbb{Z}_{x_2}][z_1{:=}\mathbb{X}_{x_1}][x_1{:=}u][x_2{:=}u] & = {t'}_{\rho'':x}u \end{array}$$

Lemma 14. *If $x \notin \mathtt{fmvar}(t)$ then $\downarrow(t\{x{:=}u\}) \equiv \downarrow(t)\{x{:=}\Rsh\downarrow(u)\}$*

Proof. By induction on t. All cases are straightforward except the substitution case. Let $t = v[y{:=}w]$. By α-conversion $y \notin \mathtt{fv}(u)$.

$$\begin{array}{ll} \downarrow(v[y{:=}w]\{x{:=}u\}) & = \\ \downarrow(v\{x{:=}u\}[y{:=}w\{x{:=}u\}]) & \equiv_{C.\ 1} \\ \downarrow(v\{x{:=}u\})_{\rho:y,\downarrow(w\{x:=u\})}\{y{:=}\Rsh\downarrow(w\{x{:=}u\})\} & \equiv_{i.h.} \\ (\downarrow(v)\{x{:=}\Rsh\downarrow(u)\})_{\rho:y,\downarrow(w)\{x:=\downarrow(u)\}}\{y{:=}\Rsh\downarrow(w)\{x{:=}\Rsh\downarrow(u)\}\} & =_{(\alpha)} \\ (\downarrow(v))_{\rho:y,\downarrow(w)}\{x{:=}\Rsh\downarrow(u)\}\{y{:=}\Rsh\downarrow(w)\{x{:=}\Rsh\downarrow(u)\}\} & =_{L.1} \\ \downarrow(v)_{\rho':y,\downarrow(w)}\{y{:=}\Rsh\downarrow(w)\}\{x{:=}\Rsh\downarrow(u)\} & \equiv_{C.\ 1} \\ \downarrow(v[y{:=}w])\{x{:=}\Rsh\downarrow(u)\} & \end{array}$$

The next technical lemma will be useful in the next section (Lemma 23).

Lemma 15. *Let t be a metaterm and x a variable. If $t \to^*_{\lambda\mathtt{j}/\equiv} t'$ and $u \to^*_{\lambda\mathtt{j}/\equiv} u'$ then for any ρ, ρ', $t_{\rho:x,u} \to^*_{\lambda\mathtt{j}/\equiv} {t'}_{\rho':x,u'}$.*

Proof. We reason by induction on the length of the reduction from t to t'.

If the length is zero then $t \equiv t'$. By Lemma 13, taking $u = u'$, we have $t_{\rho:x,u} \equiv {t'}_{\rho':x,u}$. By Lemma 12 ${t'}_{\rho':x,u} \to^*_{\lambda\mathtt{j}/\equiv} {t'}_{\rho':x,u'}$ which concludes this subcase.

Otherwise, $t \to^*_{\lambda\mathtt{j}/\equiv} t_1 \to_{\lambda\mathtt{j}/\equiv} t'$. By induction hypothesis, for any ρ, there exists a ρ_1 s.t. $t_{\rho:x,u} \to^*_{\lambda\mathtt{j}/\equiv} {t_1}_{\rho_1:x,u'}$. For the last step, we have to show that for any ρ_1, if $t_1 \to_{\lambda\mathtt{j}/\equiv} t'$ there exists a ρ' s.t. ${t_1}_{\rho_1:x,u} \to^*_{\lambda\mathtt{j}/\equiv} {t'}_{\rho':x,u}$. It is actually enough to show that for any ρ_1, ρ', if $t_1 \to_{\lambda\mathtt{j}} t'$ then ${t_1}_{\rho_1:x,u} \to^*_{\lambda\mathtt{j}/\equiv} {t'}_{\rho':x,u}$. We reason by cases on $t_1 \to_{\lambda\mathtt{j}} t'$:

– $t_1 = (\lambda y.v)\mathtt{L}\ w \to_{\mathtt{dB}} v[y{:=}w]\mathtt{L} = t'$

$$\begin{array}{ll}
t_{1\,\rho:x,u} & = \\
((\lambda y.v)\mathtt{L}\ w)_{\rho_1:x\rightsquigarrow\overline{z}}\overline{[z:=u]} & =_{L.\ 9} \\
((\lambda y.v_{\rho_v:x})\mathtt{L}_{\rho_L:x}\ w_{\rho_w:x})\overline{[z:=u]} & \to_{\mathtt{dB}} \\
v_{\rho_v:x}[y:=w_{\rho_w:x}]\mathtt{L}_{\rho_L:x}\overline{[z:=u]} & =_{L.\ 9} \\
(v[y:=w]\mathtt{L})_{\rho'':x\rightsquigarrow\overline{z}}\overline{[z:=u]} & \equiv_{C.1}\ t'_{\rho':x,u}
\end{array}$$

– $t_1 = v[y:=w] \to_{\mathtt{d}} v\{y:=w\} = t'$ As y appears exactly once in v, we have the same number of z_i in t_1 and in t'. Notice that we can apply the rule $\to_{\mathtt{d}}$ since the multiplicity of y cannot be changed by a renaming of x.

$$\begin{array}{ll}
t_{1\,\rho:x,u} & = \\
(v[y:=w])_{\rho_1:x\rightsquigarrow\overline{z}}\overline{[z:=u]} & =_{L.\ 9} \\
v_{\rho_v:x}[y:=w_{\rho_w:x}]\overline{[z:=u]} & \to_{\mathtt{d}} \\
v_{\rho_v:x}\{y:=w_{\rho_w:x}\}\overline{[z:=u]} & =_{L.\ 9} \\
(v\{y:=w\})_{\rho'':x}\overline{[z:=u]} & \equiv_{C.1}\ t'_{\rho':x,u}
\end{array}$$

– $t_1 = v[y:=u] \to_{\mathtt{w}} v = t'$ with $y \notin \mathtt{fv}(v)$

$$\begin{array}{ll}
t_{1\,\rho:x,u} & = \\
(v[y:=u])_{\rho_1:x\rightsquigarrow\overline{z}}[z_1:=u]...[z_n:=u] & =_{L.\ 9} \\
v_{\rho_v:x\rightsquigarrow\{z'_1...z'_m\}}[y:=u_{\rho_u:x}][z_1:=u]...[z_n:=u] & \to_{\mathtt{w}} \\
\text{ with } \{z'_1,...,z'_m\} \text{ a subset of } \{z_1,...,z_n\} & \\
v_{\rho_v:x\rightsquigarrow\{z'_1...z'_m\}}[z_1:=u]...[z_n:=u] & \to^*_{\mathtt{w}} \\
v_{\rho_v:x\rightsquigarrow\{z'_1,...,z'_m\}}[z'_1:=u]...[z'_m:=u] & \equiv_{C.1}\ t'_{\rho_v:x,u}
\end{array}$$

– $t_1 = v[y:=w] \to_{\mathtt{c}} v_{y\rightsquigarrow z}[y:=w][z:=w] = t'$ with $y \in_n \mathtt{fv}(v)$ and $n > 1$

$$\begin{array}{ll}
t_{1\,\rho_1:x,u} & = \\
(v[y:=w])_{\rho_1:x\rightsquigarrow\{z_1,...,z_n\}}[z_1:=u]...[z_n:=u] & =_{L.\ 9} \\
v_{\rho_v:x}[y:=w_{\rho_w:x}][z_1:=u]...[z_n:=u] & \to_{\mathtt{c}} \\
(v_{y\rightsquigarrow z})_{\rho_v:x}[y:=w_{\rho_w:x}][z:=w_{\rho_w:x}][z_1:=u]...[z_n:=u] & = \\
(v_{y\rightsquigarrow z})_{\rho_v:x}\ [y:=w_{\rho_w:x\rightsquigarrow\{z'_1...z'_m\}}] & \\
\qquad [z:=w_{\rho_w:x\rightsquigarrow\{z'_1...z'_m\}}] & \equiv \\
\qquad [z_1:=u]...[z_n:=u] & \\
\text{(with } \{z'_1...z'_m\} \text{ a subset of } \{z_1...z_n\} \text{ s.t.} & \\
\{z_1,...,z_n\} = \{z'_1,...,z'_m\} \cup \{z'_{m+1},...,z'_n\}) & \\
(v_{y\rightsquigarrow z})_{\rho_v:x}[y:=w_{\rho_w:x\rightsquigarrow\{z'_1...z'_m\}}]\ [z:=w_{\rho_w:x\rightsquigarrow\{z'_1...z'_m\}}] & \\
\qquad [z'_1:=u]...[z'_m:=u] & \to^*_{\mathtt{c}} \\
\qquad [z'_{m+1}:=u]...[z'_n:=u] & \\
(v_{y\rightsquigarrow z})_{\rho_v:x}[y:=w_{\rho_w:x\rightsquigarrow\{z'_1...z'_m\}}]\ [z:=w_{\rho_w:x\rightsquigarrow z''_1...z''_m}] & \\
\qquad [z'_1:=u]...[z'_m:=u] & \\
\qquad [z''_1:=u]...[z''_m:=u] & =_{L.\ 9} \\
\qquad [z'_{m+1}:=u]...[z'_n:=u] & \\
(v_{y\rightsquigarrow z}[y:=w][z:=w])_{\rho':x}\ [z'_1:=u]...[z'_m:=u] & \\
\qquad [z''_1:=u]...[z''_m:=u] & \equiv_{C.1}\ t'_{\rho':x,u} \\
\qquad [z'_{m+1}:=u]...[z'_n:=u] &
\end{array}$$

4 Confluence on metaterms

We combine the interpretation method with the idea of simultaneous reduction defined in Tait-Martin Löf's technique.

We first isolate the difficulties thanks to the interpretation method and reduce the confluence of $\lambda\mathtt{j}$ to the confluence of a system defined on $\mathtt{j}$-normal forms, called the interpretation calculus. There are several possibilities to define this latter one leading to different proofs. We follow ideas from Tait-Martin Löf and choose a calculus reducing simultaneously metaterms.

The second part of the proof is to finally show that the interpretation calculus enjoys the diamond property, which implies that it is confluent. This finally implies that $\lambda\mathtt{j}$ is confluent.

The interpretation method can be easily generalized in the case where there is an equivalence relation:

Lemma 16 (Interpretation method modulo). *Let $R = R_1 \cup R_2/\approx$ where R_1 is terminating on A, R_2 an arbitrary reduction, and $\approx$ an equivalence relation on A. If there exists a reduction $\Rrightarrow$ on the set of R_1 normal forms satisfying:*

1. $\Rrightarrow \subseteq\ R$
2. $t \to_{R_1/\approx} t' \ \Rightarrow \downarrow_{R_1}(t) \approx \downarrow_{R_1}(t')$
3. $t \to_{R_2/\approx} t' \ \Rightarrow \downarrow_{R_1}(t) \Rrightarrow^* \downarrow_{R_1}(t')$

then confluence of $\Rrightarrow$ implies confluence of R.

Proof. Suppose $t \to^*_{R/\approx} t_1$ and $t \to^*_{R/\approx} t_2$. By definition, $t_1 \to^*_{R_1/\approx} \downarrow_{R_1}(t_1)$, $t_2 \to^*_{R_1/\approx} \downarrow_{R_1}(t_2)$, $t \to^*_{R_1/\approx} \downarrow_{R_1}(t)$. It is easy to show by induction on the length of the derivation that if $u \to_{R/\approx} u'$, then $\downarrow_{R_1}(u) \Rrightarrow^* \downarrow_{R_1}(u')$. Hence $\downarrow_{R_1}(t) \Rrightarrow^* \downarrow_{R_1}(t_1)$ and $\downarrow_{R_1}(t) \Rrightarrow^* \downarrow_{R_1}(t_2)$. By confluence of $\Rrightarrow$, it exists t_3 such that $\downarrow_{R_1}(t_1) \Rrightarrow^* {t_3}^* \Lleftarrow \downarrow_{R_1}(t_2)$. Finally the first hypothesis ensures that $t_1 \to^*_{R/\approx} t_3$ and $t_2 \to^*_{R/\approx} t_3$.

Taking $R = \lambda\mathtt{j}$, $R_1 = \mathtt{j}$, $R_2 = \mathtt{dB}$, and $\approx = \equiv$ we thus have to find a relation $\Rrightarrow$, prove Points 1 and 3 since Point 2 was proved by Lemma 4, and of course, show that $\Rrightarrow$ is confluent.

If we had wanted to show confluence on terms and not on metaterms we would be in the situation where normal forms are actually lambda terms. We could thus easily conclude by using the fact that the λ-calculus is confluent.

In the rest of this section we will this:

1. Following Tait-Martin Löf, define $\Rrightarrow$ as a simultaneous reduction wich mimics β reductions by taking the $\mathtt{j}$-normal form after each $\mathtt{dB}$ step.
2. Prove Points 1 and 3 of Lemma 16 [4]
3. Prove confluence of $\Rrightarrow$ by showing that it enjoys the diamond property, that is any critical pair can be closed in one step.

[4] instead of showing $\to \subseteq \Rrightarrow \subseteq \to^*$ as done in the Tait-Martin Löf's technique since it is not true in our case

There are several ways to define the simultaneous reduction relation. We choose this high-level definition, which let us use all the properties we have proved in Section 3.

Definition 9 (Simultaneous reduction). *We define the simultaneous reduction* $\Rrightarrow$ *on* $\mathtt{j}$*-normal forms as follows:*

- $x \Rrightarrow x$
- $\mathbb{X} \Rrightarrow \mathbb{X}$
- *If* $u \Rrightarrow u'$ *then* $\lambda x.u \Rrightarrow \lambda x.u'$
- *If* $u \Rrightarrow u'$ *and* $v \Rrightarrow v'$ *then* $u\ v \Rrightarrow u'\ v'$
- *If* $t \Rrightarrow t'$, $u \Rrightarrow u'$ *for any* ρ, ρ' , $t_{\rho:x,u} \Rrightarrow {t'}_{\rho':x,u'}$.
- *If* $v \Rrightarrow v'$, $u \Rrightarrow u'$, $L \Rrightarrow L'$ $(\lambda x.v)\mathtt{L}\ u \Rrightarrow\!\downarrow (v'[x{:=}u']\mathtt{L}')$

Example 2. For instance, $(\lambda x.(\mathbb{X}_x\ x)(\mathbb{X}_x\ y))[y{:=}z]\ u$ can simultaneously reduce to $((\mathbb{X}_{x_1}\ u)\ (\mathbb{X}_{x_2}\ z))[x_1{:=}u][x_2{:=}u]$ or to $((\mathbb{X}_{x_2}\ u)\ (\mathbb{X}_{x_1}\ z))[x_1{:=}u][x_2{:=}u]$. The two terms are equivalent modulo $\equiv$.

The difficulty of the proof relies on the fact that $\lambda\mathtt{j}$ combines non-deterministic replacement and action at distance. Indeed, in all calculi defined by a case analysis on the constructor at the left of the substitution, it is possible to have a rule $\mathbb{X}_\Delta[x_1 := u_1]...[x_n := u_n] \Rrightarrow \mathbb{X}_\Delta[x_1 := u'_1]...[x_n := u'_n]$ with $x_i \notin \mathtt{fv}(u_j)$ and $u_i \Rrightarrow u'_i$, instead of the complex $t_{\rho:x,u} \Rrightarrow {t'}_{\rho':x,u'}$.

As a normal form may contain jumps we need to handle equations in the simultaneous reductions. To do so, we introduce a new reduction:

Definition 10 (Simultaneous reduction modulo). *The simultaneous reduction modulo* $t \Rrightarrow_\equiv t'$ *occurs if there exists* t_1, t_2 *s.t.* $t \equiv t_1 \Rrightarrow t_2 \equiv t'$.

Lemma 17. *If* $t \Rrightarrow t'$, $u \Rrightarrow u'$, $x \notin \mathtt{fmvar}(t)$ *then* $t\{x{:=}u\} \Rrightarrow_\equiv t'\{x{:=}u'\}$.

Proof. By induction on $t \Rrightarrow t'$. We only consider the interesting cases:

- $v_{\rho:y,w} \Rrightarrow {v'}_{\rho':y,w'}$ coming from $v \Rrightarrow v'$ and $w \Rrightarrow w'$. By α-conversion $x \neq y$ and $y \notin \mathtt{fv}(u)$. By induction hypothesis, $v\{x := u\} \Rrightarrow v'\{x := u'\}$ and $w\{x := u\} \Rrightarrow w'\{x := u'\}$. By definition of the implicit substitution, $t\{x{:=}u\} = v\{x{:=}u\}_{\rho:y,w\{x:=u\}}$ and by definition of $\Rrightarrow$, $v\{x{:=}u\}_{\rho:y,w\{x:=u\}} \Rrightarrow v'\{x{:=}u'\}_{\rho:y,w'\{x:=u'\}}$ which is equal to $t'\{x{:=}u'\}$.
- $(\lambda y.v)L\ w \Rrightarrow\!\downarrow (v'[y := w']L')$ coming from $v \Rrightarrow v'$, $\mathtt{L} \Rrightarrow \mathtt{L}'$, and $w \Rrightarrow w'$. By α-conversion $x \neq y$. By induction hypothesis, $v\{x := u\} \Rrightarrow v'\{x := u'\}$, $w\{x{:=}u\} \Rrightarrow w'\{x{:=}u'\}$ and $\mathtt{L}\{x{:=}u\} \Rrightarrow \mathtt{L}'\{x{:=}u'\}$.

$$
\begin{array}{ll}
t\{x{:=}u\} & = \\
(\lambda y.v\{x{:=}u\})\mathtt{L}\{x{:=}u\}\ w\{x{:=}u\} & \Rrightarrow \\
\downarrow (v'\{x{:=}u'\}[y{:=}w'\{x{:=}u'\}](\mathtt{L}'\{x{:=}u'\})) & = \\
\downarrow ((v'[y{:=}w']\mathtt{L})\{x{:=}u'\}) & \equiv_{L.\ 14} \\
\downarrow (v'[y{:=}w']\mathtt{L})\{x{:=}\!\downarrow (u')\} & =_{(u' \in \mathcal{NF}_{\mathtt{j}})} \\
\downarrow (v'[y{:=}w']L')\{x{:=}u'\} &
\end{array}
$$

Lemma 18. *If $t \Rrightarrow t'$ and $u \Rrightarrow u'$ then $\downarrow (t[x{:=}u]) \Rrightarrow_{\equiv} \downarrow (t'[x{:=}u'])$.*

Proof. By hypothesis, both t and u are in $\mathtt{j}$-normal form. Then,

$$\begin{array}{ll} \downarrow (t[x{:=}u]) & \equiv_{P.1} \\ t_{\rho:x,u}\{x{:=}u\} & \Rrightarrow_{\equiv}(\text{hyp, def and L. 17}) \\ {t'}_{\rho':x,u'}\{x{:=}u'\} & \equiv_{P.1} \\ \downarrow (t'[x{:=}u']) & \end{array}$$

It is thanks to our definition of simultaneous reduction that this proof is short. Otherwise, a proof by induction on $\Rrightarrow$ is needed, and it can take several pages as in [Kes07]. We can generalize the previous lemma to the case where the terms t and t' are affected by a list of substitutions:

Lemma 19. *If $t \Rrightarrow t'$, $\mathtt{L} \Rrightarrow \mathtt{L}'$ and $u \Rrightarrow u'$ then $\downarrow (t[x{:=}u]\mathtt{L}) \Rrightarrow_{\equiv} \downarrow (t'[x{:=}u']\mathtt{L}')$.*

Proof. By hypothesis, both t, u and $\mathtt{L}$ are in $\mathtt{j}$-normal form. We proceed by induction on the length of $\mathtt{L}$. If the length is equal to 0 we then fall in Lemma 18. Otherwise,

$$\begin{array}{ll} \downarrow (t[x{:=}u]\mathtt{L}) & = \\ \downarrow (t[x{:=}u][y_1{:=}w_1]...[y_n{:=}w_n]) & \equiv_{P.1} \\ \downarrow (t[x{:=}u][y_1{:=}w_1]...[y_{n-1}{:=}w_{n-1}])_{\rho:y_n,w_n}\{y_n{:=}w_n\} & \Rrightarrow(\text{i.h., def and L. 17} \\ \downarrow (t'[x{:=}u'][y_1{:=}w'_1]...[y_{n-1}{:=}w'_{n-1}])_{\rho':y_n,w'_n}\{y_n{:=}w'_n\} & \equiv_{P.1} \\ \downarrow (t'[x{:=}u'][y_1{:=}w'_1]...[y_n{:=}w'_n]) & = \\ \downarrow (t'[x{:=}u']\mathtt{L}') & \end{array}$$

This technical lemma will be useful to show an admissible rule necessary to show the diamond property. It is the only one allowing to go from an injective renaming to a non injective one and vice-versa.

Lemma 20. *If $t \Rrightarrow t'$ then for any ρ there exists a unique non injective ρ' and variables $z_1, ..., z_m$ such that*

1. $t_{\rho:x\rightsquigarrow\{y_1...y_n\}} \Rrightarrow {t'}_{\rho':x\rightarrowtail\{z_1...z_m\}}$ *with* $\mathtt{img}(\rho') \subseteq \mathtt{img}(\rho)$.
2. $\downarrow ({t'}_{\rho':x\rightarrowtail\{z_1...z_m\}}[y_1 := u_1]...[y_n := u_n]) = {t'}_{\rho_2:x\rightsquigarrow\{z'_1...z'_k\}}[z'_1 := u_1]...[z'_k := u_n]$ *for any ρ_2 and $u_i \in \mathcal{NF}_{\mathtt{j}}$.*

Proof. 1. By induction on $t \Rrightarrow t'$.

- The case $t = x \Rrightarrow x = t'$ is straightforward.
- The case $t = \mathbb{X}_{\Delta} \Rrightarrow \mathbb{X}_{\Delta} = t'$ is only interesting when $x \in \mathtt{fv}(\Delta)$. Then there is only one renaming possible since x appears only once and thus taking $z_1 = y$ we have $\mathbb{X}_{\Delta x\rightsquigarrow y} \Rrightarrow \mathbb{X}_{\Delta x\rightarrowtail y}$. Furthermore, $\downarrow (\mathbb{X}_{\Delta x\rightsquigarrow y}[y{:=}u]) = \mathbb{X}_{\Delta x\rightsquigarrow y}[y{:=}u]$, since by hypothesis y is a fresh variable and thus cannot appear in Δ which is equal to ${t'}_{\rho_2:x,u}$ with ρ_2 being the unique injective renaming possible in t'.

– $t = v\ w \Rrightarrow v'\ w' = t'$.
$(vw)_{\rho:x\rightsquigarrow\{y_1\dots y_n\}} =_{L.\ 9} v_{\rho_v:x\rightsquigarrow\{y'_1\dots y'_k\}}\ w_{\rho_w:x\rightsquigarrow\{y''_1\dots y''_p\}}$
By induction hypothesis,
$v_{\rho:x\rightsquigarrow\{y'_1\dots y'_n\}} \Rrightarrow {v'}_{\rho'_v:x\rightsquigarrow\{z'_1\dots z'_m\}}$ and $w_{\rho:x\rightsquigarrow\{y''_1\dots y''_p\}} \Rrightarrow {w'}_{\rho'_w:x\rightsquigarrow\{z''_1\dots z''_q\}}$ with $\mathtt{img}(\rho'_v) \subseteq \mathtt{img}(\rho_v)$ and $\mathtt{img}(\rho'_w) \subseteq \mathtt{img}(\rho_w)$
Thus, $v_{\rho:x\rightsquigarrow\{y'_1\dots y'_k\}}\ w_{\rho:x\rightsquigarrow\{y''_1\dots y''_p\}} \Rrightarrow {v'}_{\rho'_v:x\rightarrowtail\{z'_1\dots z'_m\}}\ {w'}_{\rho'_w:x\rightarrowtail\{z''_1\dots z''_q\}}$ which is equal by Lemma 9 to $(v'\ w')_{\rho':x\rightarrowtail\{z'_1\dots z'_m, z''_1\dots z''_q\}}$. Furthermore, $\mathtt{img}(\rho') = \mathtt{img}(\rho'_v) \cup \mathtt{img}(\rho'_w)$ which are a subset of $\mathtt{img}(\rho_v) \cup \mathtt{img}(\rho_w)$ by hypothesis and $\mathtt{img}(\rho_v) \cup \mathtt{img}(\rho_w) = \mathtt{img}(\rho)$ by Lemma 9.
– Cases $\lambda z.v \Rrightarrow \lambda z.v'$ and $v_{\rho:x,w} \Rrightarrow {v'}_{\rho':x,w'}$ are similar (i.h. and Lemma 9).
– $(\lambda z.v)\mathtt{L}\ w \Rrightarrow{\downarrow} (v'[z{:=}w']\mathtt{L}')$.

$$
\begin{array}{ll}
((\lambda z.v)\mathtt{L}\ w) & =_{L.\ 9} \\
(\lambda z.v_{\rho_v:x\rightsquigarrow y'_1\dots y'_\alpha})\mathtt{L}_{\rho_L:x\rightsquigarrow y'''_1\dots y'''_\gamma}\ w_{\rho_w:x\rightsquigarrow y''_1\dots y''_\beta} & \Rrightarrow \\
{\downarrow}\,({v'}_{\rho'_v:x\rightsquigarrow\{z'_1\dots z'_\delta\}}[z{:=}{w'}_{\rho'_w:x\rightsquigarrow\{z''_1\dots z''_\epsilon\}}]\mathtt{L}'_{\rho'_\mathtt{L}:x\rightsquigarrow\{z'''_1\dots z'''_\zeta\}}) & =_{L.\ 9} \\
{\downarrow}\,((v'[z{:=}w']\mathtt{L}')_{\rho'':x\rightsquigarrow\{z'_1\dots z'_\delta, z''_1\dots z''_\epsilon, z'''_1\dots z'''_\zeta\}}) & =_{L.\ 10} \\
{\downarrow}\,(v'[z{:=}w']\mathtt{L}')_{\rho':x\rightsquigarrow\{z_1\dots z_m\}} & =\!{\downarrow}\,(t')_{\rho':x\rightsquigarrow\{z_1\dots z_m\}}
\end{array}
$$

The fact that $\mathtt{img}(\rho') \subseteq \mathtt{img}(\rho)$ is easy using Lemmas 9 and 10.

2. By applying rules `w` and `c`.

Example 3. Take $t = (\lambda y.y\ y)\ \mathbb{X}_x \Rrightarrow \mathbb{X}_x\ \mathbb{X}_x = t'$ together with the unique valid renaming for t which associates the variable z to the unique occurrence of x. Then $t_{\rho:x\rightsquigarrow\{z\}} = (\lambda y.y\ y)\ \mathbb{X}_z \Rrightarrow \mathbb{X}_z\ \mathbb{X}_z = {t'}_{\rho':x\rightarrowtail\{z\}}$ with ρ' the renaming associating to all the occurrences of x in t' the variable z.

To illustrate the second point of Lemma 20, take any $u \in \mathcal{NF}_\mathtt{j}$ and notice that ${\downarrow}\,((\mathbb{X}_y\ \mathbb{X}_y)[y := u])$ can reduce either to $(\mathbb{X}_{y_1}\ \mathbb{X}_{y_2})[y_1 := u][y_2 := u]$ or to $(\mathbb{X}_{y_2}\ \mathbb{X}_{y_1})[y_1 := u][y_2 := u]$ which correspond to ${t'}_{\rho_2:x\rightsquigarrow\{y_1,y_2\}}[y_1 := u][y_2 := u]$ or ${t'}_{\rho'_2:x\rightsquigarrow\{y_1,y_2\}}[y_1{:=}u][y_2{:=}u]$, ρ_2 and ρ'_2 being the two valid renamings for t'.

Lemma 21. *We have the following admissible rules:*

1. *If $t \Rrightarrow t'$, $u \Rrightarrow u'$, $x \in_1 \mathtt{fmvar}(t)$ then $\underline{t[x{:=}u]} \Rrightarrow{\downarrow} (t'[x{:=}u'])$.*
2. *If $t \Rrightarrow t'$, $u_i \Rrightarrow u'_i$ then $t_{\rho:x\rightsquigarrow\{y_1,\dots,y_n\}}\overline{[y_i{:=}u_i]} \Rrightarrow {t'}_{\rho:x\rightsquigarrow\{z_1,\dots,z_m\}}\overline{[z_i{:=}u'_i]}$*

Lemma 22. *If $t \to_\mathtt{dB} t'$ then ${\downarrow}\,(t) \Rrightarrow_{\equiv}{\downarrow}\,(t)'$.*

Proof. By induction on the relation $t \to_\mathtt{dB} t'$. If the reduction occurs at the root of t i.e. if $t = (\lambda x.v)\mathtt{L}\ u \to_\mathtt{dB} v[x := u]\mathtt{L} = t'$, notice that ${\downarrow}\,(t) = (\lambda x.v')\mathtt{L}'\ u'$. Then by Lemma 8, ${\downarrow}\,(t') =\!{\downarrow}\,(v'[x{:=}u']\mathtt{L}')$ and we conclude by definition of $\Rrightarrow$.

Otherwise, for the subterm u where the `dB` redex is performed we have $u \Rrightarrow u'$. We easily conclude by induction hypothesis except for the two following cases:

– $t = v[x := u] \to_\mathtt{dB} v[x := u'] = t'$. ${\downarrow}\,(t) \equiv {\downarrow}\,(v)_{\rho:x,{\downarrow}(u)}\{x :=\!\downarrow\ (u)\}$ with ${\downarrow}\,(u) \Rrightarrow{\downarrow}\,(u')$ by induction hypothesis. By Lemma 17 and definition of $\Rrightarrow$ we get that ${\downarrow}\,(v)_{\rho:x,{\downarrow}(u)}\{x :=\!\downarrow\ (u)\} \Rrightarrow {\downarrow}\,(v)_{\rho:x,{\downarrow}(u')}\{x :=\!\downarrow\ (u')\}$
– The case $t = u[x{:=}v] \to_\mathtt{dB} u'[x{:=}v] = t'$ is similar to the previous one.

We can now deduce Point 3 of Lemma 16:

Corollary 2. *If* $t \to_{\mathtt{dB}/\equiv} t'$ *then* $\downarrow(t) \Rrightarrow_{\equiv} \downarrow(t')$.

And Point 1 of Lemma 16:

Lemma 23. *If* $t \Rrightarrow_{\equiv} t'$ *then* $t \to^*_{\lambda\mathtt{j}/\equiv} t'$.

Proof. The case $t \equiv t'$ is straightforward. Otherwise, it is enough to show that $t \Rrightarrow t_1$ implies $t \to^*_{\lambda\mathtt{j}/\equiv} t_1$. Indeed if $t \equiv t_0 \Rrightarrow t'_0 \equiv t_1$ then by hypothesis, $t_0 \to^*_{\lambda\mathtt{j}/\equiv} t'_0$ and thus $t \to^*_{\lambda\mathtt{j}/\equiv} t_1$.

The interesting case is the following one:

- $t = v_{\rho:x,u} \Rrightarrow {v'}_{\rho':x,u'} = t_1$ coming from $v \Rrightarrow v'$, $u \Rrightarrow u'$. By induction hypothesis, $v \to^*_{\lambda\mathtt{j}/\equiv} v'$ and $u \to^*_{\lambda\mathtt{j}/\equiv} u'$. We can then conclude with Lemma 15.

Lemma 24. *If* $t \Rrightarrow_{\equiv} t'$, $u \Rrightarrow_{\equiv} u'$ *then for any* ρ, ρ' $t_{\rho:x,u} \Rrightarrow_{\equiv} {t'}_{\rho':x,u'}$

Proof. By hypothesis, $t \equiv t_1 \Rrightarrow t_2 \equiv t'$, $u \equiv u_1 \Rrightarrow u_2 \equiv u'$. By Lemma 13, $t_{\rho:x,u} \equiv {t_1}_{\rho:x,u_1}$. By hypothesis and definition of $\Rrightarrow$, for any ρ, ${t_1}_{\rho:x,u_1} \Rrightarrow {t_2}_{\rho:x,u_2}$. Again, by Lemma 13, ${t_2}_{\rho:x,u_2} \equiv {t'}_{\rho:x,u'}$ which concludes.

Lemma 25 ($\Rrightarrow_{\equiv}$ has the diamond property). *If* $t_1 \Lleftarrow_{\equiv} t \Rrightarrow_{\equiv} t_2$, *then there exists* t_3 *such that* $t_1 \Rrightarrow_{\equiv} t_3 \Lleftarrow_{\equiv} t_2$.

1. We first prove that if $t' \overset{i}{\Lleftarrow} t \equiv^1 t_1$ then it exists t'_1 s.t $t' \equiv t'_1 \overset{i}{\Lleftarrow} t_1$. We proceed by induction on $\equiv^1$. The base cases are the following ones:
   - ${u'}_{\rho_x:x,v'}{}_{\rho_y:y,w'} \Lleftarrow u[x{:=}v][y{:=}w] \equiv^1 u[y{:=}w][x{:=}v]$ with:
     - $x \notin \mathtt{fv}(w)$ and $y \notin \mathtt{fv}(v)$
     - $x, y \in_1 \mathtt{fmvar}(u)$
     - $u \Rrightarrow u'$, $v \Rrightarrow v'$, $w \Rrightarrow w'$

     Then,

$$
\begin{array}{ll}
u[y{:=}w][x{:=}v] & \Rrightarrow \\
{u''}_{\rho'_y:y,w''}{}_{\rho'_x:x,v''} & = \\
{u''}_{\rho'_y:y\rightsquigarrow\{b_1,\ldots,b_m\}}\overline{[b{:=}w'']}{}_{\rho'_x:x\rightsquigarrow\{a_1,\ldots,a_n\}}\overline{[a{:=}v'']} & =_{L.\ 9} \\
{u''}_{\rho'_y:y\rightsquigarrow\{b_1,\ldots,b_m\}}{}_{\rho''_x:x\rightsquigarrow\{a_1,\ldots,a_n\}}\overline{[b{:=}w''][a{:=}v'']} & \equiv \\
{u''}_{\rho'_y:y\rightsquigarrow\{b_1,\ldots,b_m\}}{}_{\rho''_x:x\rightsquigarrow\{a_1,\ldots,a_n\}}\overline{[a{:=}v''][b{:=}w'']} & = \\
{u''}_{\rho''_x:x\rightsquigarrow\{a_1,\ldots,a_n\}}{}_{\rho'_y:y\rightsquigarrow\{b_1,\ldots,b_m\}}\overline{[a{:=}v''][b{:=}w'']} & = \\
{u''}_{\rho''_x:x\rightsquigarrow\{a_1,\ldots,a_n\}}{}_{\rho'_y:y\rightsquigarrow\{b_1,\ldots,b_m\}}\overline{[a{:=}v''][b{:=}w'']} & =_{L.\ 9} \\
{u''}_{\rho'_x:x\rightsquigarrow\{a_1,\ldots,a_n\}}\overline{[a{:=}v'']}{}_{\rho''_y:y\rightsquigarrow\{b_1,\ldots,b_m\}}\overline{[b{:=}w'']} & \\
{u''}_{\rho''_x:x,v''}{}_{\rho''_y:y,w''} &
\end{array}
$$

     By hypothesis, $u' \Lleftarrow u \equiv^1 u$, $v' \Lleftarrow v \equiv^1 v$, and $w' \Lleftarrow w \equiv^1 w$. Thus, by induction hypothesis, $u' \equiv u'' \Lleftarrow u$, $v' \equiv v'' \Lleftarrow v$, and $w' \equiv w'' \Lleftarrow w$. We thus have:

$$\begin{array}{l}{u''}_{\rho''_x:x,v''})_{\rho''_y:y,w''} \equiv \\ ({u'}_{\rho''_x:x,v'})_{\rho''_y:y,w'} \quad \equiv_{P.\ 1} \\ ({u'}_{\rho_x:x,v'})_{\rho_y:y,w'}\end{array}$$

- ${u'}_{\rho':x,v'} \Leftarrow u_{\rho:x,v} \equiv^1 {u_1}_{\rho_1:x,v}$. By induction hypothesis, $u' \equiv u'_1 \Leftarrow u_1$, $v' \equiv v'_1 \Leftarrow v$ and thus ${u'}_{\rho':x,v'} \equiv_{L.\ 13} {u'_1}_{\rho'_1:x,v'} \equiv {u'_1}_{\rho'_1:x,v'_1} \Leftarrow {u_1}_{\rho_1:x,v}$
- $$\begin{array}{rcl} {u'}_{\rho':x,v'} \Leftarrow & & u_{\rho:x\rightsquigarrow\{y_1,\ldots,y_n\}}\overline{[y:=v]} \\ & & \equiv^1 \\ & & u_{\rho_1:x\rightsquigarrow\{y_1,\ldots,y_n\}}[y_1:=v]...[y_i:=v_1]...[y_n:=v] \end{array}$$

  By induction hypothesis, $v' \equiv v'_1 \Leftarrow v$ and $v' \equiv v'_1 \Leftarrow v_1$.
  By Lemma 21, $u_{\rho_1:x\rightsquigarrow\{y_1,\ldots,y_n\}}[y_1:=v]...[y_i:=v_1]...[y_n:=v] \Rrightarrow u_{\rho'_1:x,v'_1}$.
  Finally $u_{\rho'_1:x,v'_1} \equiv_{L.\ 12} u_{\rho'_1:x,v'} \equiv_{P.\ 1} {u'}_{\rho':x,v'}$.
- $\downarrow (v'[x:=w']\mathtt{L}') \Leftarrow (\lambda x.v)\mathtt{L}\ w \equiv (\lambda x.v)\mathtt{L}_1\ w$ with $\mathtt{L} \equiv^1 \mathtt{L}_1$. By induction hypothesis, $w' \equiv w'_1 \Leftarrow w$, $v' \equiv v'_1 \Leftarrow v$, $\mathtt{L}' \equiv \mathtt{L}'_1 \Leftarrow \mathtt{L}_1$. We thus have to show that $\downarrow (v'[x:=w']\mathtt{L}') \equiv\downarrow (v'_1[x:=w'_1]\mathtt{L}'_1)$ which is true by definition of $\downarrow(\ )$.
- The cases $\downarrow (v'[x := w']\mathtt{L}') \Leftarrow (\lambda x.v)\mathtt{L}\ w \equiv (\lambda x.v_1)\mathtt{L}\ w$ and $\downarrow (v'[x := w']\mathtt{L}') \Leftarrow (\lambda x.v)\mathtt{L}\ w \equiv (\lambda x.v)\mathtt{L}\ w_1$ are similar to the previous case.

2. We prove $t' \Leftarrow t \equiv t_1$ implies $t' \equiv t'_1 \Leftarrow t_1$.

   Proof. By induction on the number k of $\equiv$ steps between t and t_1. If $k = 0$, then this is straightforward. If $k = k' + 1$, we conclude with the following diagram:

$$\begin{array}{ccccc} t & \equiv & t_0 & \equiv\ ... & t_1 \\ & \text{Point 1} & & \textit{i.h.} & \\ \Rrightarrow & & \Rrightarrow & & \Rrightarrow \\ t' & \equiv & t'_0 & \equiv\ ... & t'_1 \end{array}$$

3. We prove $t' \Leftarrow_{\equiv} t \equiv t_1$ implies $t' \equiv t'_1 \Leftarrow t_1$.

   Proof. If $t' \Leftarrow_{\equiv} t \equiv t_1$ then we have t_0 and t'_0 s.t. $t' \equiv t'_0 \Leftarrow t_0 \equiv t_1$. By the previous point, $t_0 \Rrightarrow t'_0$. We can conclude by applying one more time the previous point.

4. We prove $t_1 \Leftarrow t \Rrightarrow t_2$ implies there exists t_3 such that $t_1 \Rrightarrow_{\equiv} t_3 \Leftarrow_{\equiv} t_2$.

   Proof. By induction on $\Rrightarrow$. The interesting cases are the following ones:
   - $(\lambda x.v_1)\mathtt{L}_1\ u_1 \Leftarrow (\lambda x.v)\mathtt{L}\ u \Rrightarrow (\lambda x.v_2)\mathtt{L}_2\ u_2$ with $u \Rrightarrow u_1$, $u \Rrightarrow u_2$, $v \Rrightarrow v_1$, $v \Rrightarrow v_2$. We can conclude by induction hypothesis.
   - $(\lambda x.v_1)\mathtt{L}_1\ u_1 \Leftarrow (\lambda x.v)\mathtt{L}\ u \Rrightarrow\downarrow (v_2[x := u_2]\mathtt{L}_2)$ with $u \Rrightarrow u_1$, $u \Rrightarrow u_2$, $v \Rrightarrow v_1$, $v \Rrightarrow v_2$. By induction hypothesis, $u_1 \Rrightarrow u_3 \Leftarrow u_2$ and $v_1 \Rrightarrow v_3 \Leftarrow v_2$. By hypothesis, $(\lambda x.v_1)\mathtt{L}_1\ u_1 \Rrightarrow\downarrow (v_3[x := u_3]\mathtt{L}_3)$ and by Lemma 19 $\downarrow (v_2[x:=u_2]\mathtt{L}_2) \Rrightarrow\downarrow (v_3[x:=u_3]\mathtt{L}_3)$ which concludes.
   - The case $\downarrow (v_1[x:=u_1]\mathtt{L}_1) \Leftarrow (\lambda x.v)\ u \Rrightarrow\downarrow (v_2[x:=u_2]\mathtt{L}_2)$ is similar to the previous one.

– $t_{1_{\rho_1:x,u_1}} \Leftarrow t_{\rho:x,u} \Rrightarrow t_{2_{\rho_2:x,u_2}}$. By i.h., $t_1 \Rrightarrow_{\equiv} t_3 \Leftarrow_{\equiv} t_2$ and $u_1 \Rrightarrow_{\equiv} u_3 \Leftarrow_{\equiv} u_2$. By Lemma 24, $t_{1_{\rho_1:x,u_1}} \Rrightarrow_{\equiv} t_{3_{\rho_3:x,u_3}}$ and $t_{2_{\rho_2:x,u_2}} \Rrightarrow_{\equiv} t_{3_{\rho'_3:x,u_3}}$. We can conclude with Property 1 since $t_{3_{\rho_3:x,u_3}} \equiv t_{3_{\rho'_3:x,u_3}}$.

5. We finally prove that $t_1 \Leftarrow_{\equiv} t \Rrightarrow_{\equiv} t_2$ implies there exists t_3 such that $t_1 \Rrightarrow_{\equiv} t_3 \Leftarrow_{\equiv} t_2$.

   Proof. Let $t_1 \Leftarrow_{\equiv} t \equiv u \Rrightarrow u' \equiv t_2$. By the third point, there is u_1 such that $t_1 \equiv u_1 \Leftarrow u$ and by the fourth point there is t_3 such that $u_1 \Rrightarrow_{\equiv} t_3 \Leftarrow_{\equiv} u'$. We conclude that $t_1 \Rrightarrow_{\equiv} t_3 \Leftarrow_{\equiv} t_2$.

We finally conclude since the diamond property obviously implies confluence:

Corollary 3. *The simultaneous reduction $\Rrightarrow_{\equiv}$ is confluent.*

Theorem 1 (Metaconfluence of $\to_{\mathtt{dB}\cup\mathtt{j}/\equiv}$). *The reduction relation $\to_{\mathtt{dB}\cup\mathtt{j}/\equiv}$ is confluent on metaterms.*

Proof. Thanks to Lemma 16, it is implied by Lemmas 23 and 4, and Corollaries 2 and 3.

4.1 Confluence Extensions

Several strong extensions can be easily implied by metaconfluence thanks to the fact that the equivalence $\equiv$ can always be postponed with respect to $\lambda\mathtt{j}$ because it is a strong bisimulation, the strongest property that an equivalence relation can define over terms.

Lemma 26. *The reduction relation $\equiv$ is an internal strong bisimulation i.e. if $t \equiv u$ then $t \to_{\lambda\mathtt{j}} t'$ implies $u \to_{\lambda\mathtt{j}} u' \equiv t'$.*

Proof. Assume $t \equiv t'$ holds in n steps, which is written as $t \equiv^n t'$, and that $t' \to_{\lambda\mathtt{j}} s'$. We show that it exists s s.t. $t \to_{\lambda\mathtt{j}} s$ and $s \equiv s'$. We proceed by induction on n. The proof is almost the same as the one for $\lambda\mathtt{j}$ and is straightforward.

Lemma 27 ($\equiv$ postponement). *If $t \to^*_{\lambda\mathtt{j}/\equiv} t'$ then $t \to^*_{\lambda\mathtt{j}} \equiv t'$. Furthermore, the number of $\lambda\mathtt{j}$ steps is preserved during the reduction.*

Proof. By definition of internal strong bisimulation.

Confluence modulo and Church-Rosser modulo: As shown in Lemma 3, there exists stronger properties than metaconfluence for a given reduction relation modulo.

Theorem 2. *The reduction relation $\lambda\mathtt{j}$ is confluent on metaterms modulo $\equiv$.*

Proof. If $t_1 \,{}^*_{\lambda\mathtt{j}}\!\leftarrow t \equiv t' \to^*_{\lambda\mathtt{j}} t_2$ then by Theorem 1, $t_1 \to^*_{\lambda\mathtt{j}/\equiv} t_3 \,{}^*_{\lambda\mathtt{j}/\equiv}\!\leftarrow t_2$. By Lemma 27, we can postpone $\equiv$ and get $t_1 \to^*_{\lambda\mathtt{j}} \equiv t_3 \equiv {}^*_{\lambda\mathtt{j}}\!\leftarrow t_2$ which concludes.

Theorem 3. *The reduction relation $\lambda\mathtt{j}$ is Church-Rosser modulo $\equiv$ on metaterms.*

Proof. Let $t \leftrightarrow^*_{\equiv} t'$ in n steps. We proceed by induction on n. The case $n = 1$ is straightforward. Otherwise, we analyse the first step.

- If $t \to_{\lambda\mathtt{j}} t_1 \leftrightarrow^*_{\equiv} t'$ then we have $t_1 \to^*_{\lambda\mathtt{j}} \equiv {}^*_{t'}\!\leftarrow$ by induction hypothesis and can immediately conclude.
- If $t \equiv t_1 \leftrightarrow^*_{\equiv} t'$ then we have $t_1 \to^*_{\lambda\mathtt{j}} s_1 \equiv s_2 \, {}^*_{t'}\!\leftarrow$. By Theorem 2 $t \to_{\lambda\mathtt{j}} s_0 \equiv s_1$ and we can conclude by transitivity of $\equiv$.
- If $t \ {}_{\lambda\mathtt{j}}\!\leftarrow t_1 \leftrightarrow^*_{\equiv} t'$ then we have $t_1 \to^*_{\lambda\mathtt{j}} s_1 \equiv s_2 \, {}^*_{t'}\!\leftarrow$. By Theorem 2 $t \to_{\lambda\mathtt{j}} t_3 \equiv s_0 \, {}^*_{\lambda\mathtt{j}}\!\leftarrow s_1$. To conclude, we transform the reduction $t' \to^*_{\lambda\mathtt{j}} s_2 \equiv s_1 \to_{\lambda\mathtt{j}} \equiv s_0 \equiv t_3$ in $t' \to^*_{\lambda\mathtt{j}} s_2 \to_{\lambda\mathtt{j}} \equiv s_0 \equiv t_3$ with Lemma 27.

Sigma equivalence: Other equivalence relations are defined in [AK10] including σ-equivalence on λ-terms [Reg91]. It is specified by means of the following equations:

$$\begin{array}{ll} (\lambda y.t)[x{:=}u] & \equiv_{\sigma_1} \lambda y.t[x{:=}u] \ y \notin \mathtt{fv}(u) \\ (t \ v)[x{:=}u] & \equiv_{\sigma_2} t[x{:=}u] \ v \quad x \notin \mathtt{fv}(v) \end{array}$$

The equivalence relation $\equiv \cup \equiv_{\sigma_1} \cup \equiv_{\sigma_2}$ also defines a strong bisimulation on terms, and metaterms. We can thus extend the Church-Rosser property as we have done for $\equiv$, thus obtaining that the reduction relation $\lambda\mathtt{j}$ is Church-Rosser modulo $\equiv \cup \equiv_{\sigma_1} \cup \equiv_{\sigma_2}$ on metaterms.

$\lambda\mathtt{j}$-dags: An other easy consequence of our work is that $\lambda\mathtt{j}$-dags [AG09], the graph formalism of $\lambda\mathtt{j}$, extended with metaterms is confluent. Indeed, there is a strong bisimulation between the two formalisms. All the notions of equivalence on terms do not apply in this graphical representation since they become simply equalities.

5 Conclusion

We have first defined different properties on the non deterministic replacement operation used by the $\lambda\mathtt{j}$-calculus. All of them are indepedent from any notion of confluence and could be thus used to show other properties. We also proved that the reduction relation $\lambda\mathtt{j}$ equipped with an equivalence relation enjoys meta-confluence by using relatively standard techniques. This was possible thanks to the efforts done to fully expand substitutions with respect to, among others, the non deterministic replacements. We finally proved the Church-Rosser modulo property which is the stronger property that a calculus can enjoy, thanks to the fact that the equivalence relation can be always postponed.

References

ACCL91. Martín Abadi, Luca Cardelli, Pierre Louis Curien, and Jean-Jacques Lévy. Explicit substitutions. *Journal of Functional Programming*, 4(1):375–416, 1991.

AG09. Beniamino Accattoli and Stefano Guerrini. Jumping boxes. In Erich Grädel and Reinhard Kahle, editors, *CSL*, volume 5771 of *Lecture Notes in Computer Science*, pages 55–70. Springer, 2009.

AK10. Beniamino Accattoli and Delia Kesner. The structural lambda-calculus. In *CSL*. Springer, 2010.

Bar84. Henk Barendregt. *The Lambda Calculus: Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 1984. Revised Edition.

Blo97. Roel Bloo. *Preservation of Termination for Explicit Substitution*. PhD thesis, Eindhoven University of Technology, 1997.

CHL96. Pierre-Louis Curien, Thérèse Hardin, and Jean-Jacques Lévy. Confluence properties of weak and strong calculi of explicit substitutions. *Journal of the ACM*, 43(2):362–397, March 1996.

DHKP96. Gilles Dowek, Thérèse Hardin, Claude Kirchner, and Frank Pfenning. Higher-order unification via explicit substitutions: the case of higher-order patterns. In Michael Maher, editor, *Joint International Conference and Symposium on Logic Programming*, pages 259–273, September 1996.

Gir87. Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–101, 1987.

Har89. Thérèse Hardin. Confluence results for the strong pure categorical logic CCL; λ-calculi as subsystems of CCL. *Theoretical Computer Science*, 65, 1989.

Hue76. Gérad Huet. *Résolution d'équations dans les langages d'ordre* $1, 2, \ldots, \omega$. Thèse de doctorat d'état, Université Paris VII, 1976.

Kes07. Delia Kesner. The theory of explicit substitutions revisited. In *16th EACSL Annual Conference on Computer Science and Logic (CSL)*, volume 4646 of *Lecture Notes in Computer Science*, pages 238–252. Springer-Verlag, 2007.

Kes09. Delia Kesner. A theory of explicit substitutions with safe and full composition. *Logical Methods in Computer Science*, pages 1–29, 2009.

KR95. Fairouz Kamareddine and Alejandro Ríos. The confluence of the λs_e-calculus via a generalized interpretation method, 1995. Draft.

KR97. Fairouz Kamareddine and Alejandro Ríos. Extending a λ-calculus with explicit substitution which preserves strong normalisation into a confluent calculus on open terms. *Journal of Functional Programming*, 7(4):395–420, 1997.

KR09. Delia Kesner and Fabien Renaud. The prismoid of resources. In Rastislav Královic and Damian Niwinski, editors, *MFCS*, volume 5734 of *Lecture Notes in Computer Science*, pages 464–476. Springer, 2009.

Nad02. Gopalan Nadathur. The suspension notation for lambda terms and its use in metalanguage implementations. *Electr. Notes Theor. Comput. Sci.*, 67:35–48, 2002.

RBL09. Kristoffer Rose, Roel Bloo, and Frédéric Lang. On explicit substitution with names, 2009. IBM Research Report.

Reg91. Laurent Regnier. *λ-calcul et réseaux*. PhD thesis, Université de Paris VII, 1991. Thèse de doctorat de mathématiques.

SFM03. Francois-Régis Sinot, Maribel Fernández, and Ian Mackie. Efficient reductions with director strings. In Robert Nieuwenhuis, editor, *14th International Conference on Rewriting Techniques and Applications (RTA)*, volume 2706 of *Lecture Notes in Computer Science*, pages 46–60. Springer-Verlag, June 2003.

Ter03. Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.

vO08a. Vincent van Oostrom. Confluence by decreasing diagrams. In Andrei Voronkov, editor, *RTA*, volume 5117 of *Lecture Notes in Computer Science*, pages 306–320. Springer, 2008.

vO08b. Vincent van Oostrom. The z property. Slides available on `http://www.phil.uu.nl/~oostrom/publication/rewriting.html`, 2008.

YH90. Hirofumi Yokouchi and Teruo Hikita. A rewriting system for categorical combinators with multiple arguments. *SIAM J. Comput.*, 19(1):78–97, 1990.