Confluence and Strong Normalisation of the Generalised Multiary $\boldsymbol{\lambda}$-calculus

José Espírito Santo and Luís Pinto*

Departamento de Matemática, Universidade do Minho
4710-057 Braga, Portugal
{jes,luis}@math.uminho.pt

Abstract. In a previous work we introduced the *generalised multiary* λ*-calculus* $\lambda\mathbf{J^m}$, an extension of the λ-calculus where functions can be applied to lists of arguments (a feature which we call "multiarity") and encompassing "generalised" eliminations of von Plato. In this paper we prove confluence and strong normalisation of the reduction relations of $\lambda\mathbf{J^m}$. Proofs of these results lift corresponding ones obtained by Joachimski and Matthes for the system ΛJ. Such lifting requires the study of how multiarity and some forms of generality can express each other. This study identifies a variant of ΛJ, and another system isomorphic to it, as being the subsystems of $\lambda\mathbf{J^m}$ with, respectively, minimal and maximal use of multiarity. We argue then that $\lambda\mathbf{J^m}$ is the system with the right use of multiarity.

1 Introduction

In [2] we defined the generalised multiary λ-calculus $\lambda\mathbf{J^m}$, an extension of the λ-calculus where application is generalised in two directions: (i) "generality", in the sense of von Plato's generalised eliminations [7]; and (ii) "multiarity", *i.e.* the ability of applying functions to lists of arguments. The original motivation was to extend Schwichtenberg's work on permutative conversions for intuitionistic cut-free sequent calculus [6]. $\lambda\mathbf{J^m}$ comes equipped with a set of permutative conversions for which the permutability theorem holds: two $\lambda\mathbf{J^m}$-terms determine the same λ-term iff they are inter-permutable. We established confluence and strong normalisation of these conversions.

In this paper we study confluence and strong normalisation for the reduction rules of $\lambda\mathbf{J^m}$. Our strategy is to use corresponding properties of the system ΛJ of Joachimski and Matthes [4, 5] (the type-theoretic counterpart to von Plato's natural deduction system with generalised eliminations). This is a natural approach because ΛJ may be seen as a notational variant of a subsystem of $\lambda\mathbf{J^m}$ called $\lambda\mathbf{J}$.

We lift the results of ΛJ to $\lambda\mathbf{J^m}$ via a mapping ν whose idea is to express multiarity by means of generality. To fully achieve this we also need another

* Both authors are supported by FCT through the Centro de Matemática da Universidade do Minho, and also by the thematic network APPSEM II; the second author was also supported by the thematic network TYPES.

mapping μ, which expresses certain uses of generality by multiarity and which calculates the normal forms for the reduction rule of $\lambda\mathbf{J^m}$ with the same name. It follows that μ and ν are inverse bijections between μ-normal forms and terms of $\lambda\mathbf{J}$. We develop this idea and investigate how these mappings preserve reduction. It turns out that a slight variant of $\lambda\mathbf{J}$ is isomorphic to the subsystem of $\lambda\mathbf{J^m}$ determined by the μ-normal forms.

This emphasis on how multiarity and generality may express each other contrasts with that in [2], where multiarity and generality are studied as independent features of $\lambda\mathbf{J^m}$.

This paper is organised as follows: Section 2 reviews $\lambda\mathbf{J^m}$ and its subsystem $\lambda\mathbf{J}$; Section 3 studies mappings μ and ν and establishes the above mentioned isomorphism; Section 4 proves various results of concluence and strong normalisation; Section 5 concludes.

Notations: Let R be a binary relation over an inductively defined set of expressions. $\rightarrow_R$ denotes the compatible closure of R. $\rightarrow^+_R$ and $\rightarrow^*_R$ denote respectively the transitive; and the reflexive and transitive closure of $\rightarrow_R$. Given relations R and S, we write R, S and RS for $R \cup S$ and $S \circ R$, respectively, whenever convenient.

2 $\lambda\mathbf{J^m}$: the generalised multiary λ-calculus

2.1 Expressions and typing rules

Let $\mathbf{V}$ denote a denumerable set of variables and x, y, w, z range over it. In the generalised multiary λ-calculus $\lambda\mathbf{J^m}$ there are two kinds of expressions: terms and lists.

Definition 1. *Terms and lists of $\lambda\mathbf{J^m}$ are described in the following grammar:*

$$\begin{array}{ll} (\textit{terms of } \lambda\mathbf{J^m}) & t, u, v ::= x \mid \lambda x.t \mid t(u, l, (x)v) \\ (\textit{lists of } \lambda\mathbf{J^m}) & \qquad\quad l ::= t :: l \mid [] \end{array}$$

The sets of $\lambda\mathbf{J^m}$-terms and $\lambda\mathbf{J^m}$-lists are denoted by $\mathbf{\Lambda J^m}$ and $\mathcal{L}\mathbf{J^m}$ respectively. A term construction of the form $t(u, l, (x)v)$ is called a generalised multiary application *(*gm-application *for short) and t is called its* head*. In terms $\lambda x.v$ and $t(u, l, (x)v)$, occurrences of x in v are bound. The list $[]$ is called the* empty list *and lists of the form $t :: l$ are called* cons-lists*. The notation $[u_1, \ldots, u_n]$ abbreviates $u_1 :: \ldots :: u_n :: []$.*

Two definitions that play a special role in the following are:

Definition 2. *A gm-application is called a* cut *if its head is not a variable.*

Definition 3. *A variable x is* main and linear *in a term t if $t = x$ or t is of the form $x(u, l, (y)v)$ where $x \notin u, l, v$. We write $\mathbf{mla}(x, v)$ if v is a gm-**a**pplication and x is **m**ain and **l**inear in v.*

Formulas (= *types*) A, B, C, ... are built up from propositional variables using just $\supset$ (for implication) and *contexts* Γ are finite sets of *variable : formula* pairs, associating at most one formula to each variable.

Sequents of $\lambda\mathbf{J^m}$ are of one of the following two forms

$$\Gamma; -\vdash t : A$$
$$\Gamma; B \vdash l : C,$$

called *term sequents* and *list sequents* respectively. The distinguished position in the LHS of sequents is called the *stoup* and may either be empty (as in term sequents) or hold a formula (the case of list sequents). Read a list sequent $\Gamma; B \vdash l : C$ as "list l leads the formula B to its instance C in context Γ". C is an *instance* of B if B is of the form $B_1 \supset ... \supset B_k \supset C$, for some $k \geq 0$.

Definition 4. *The typing rules of* $\lambda\mathbf{J^m}$ *are as follows:*

$$\frac{}{x : A, \Gamma; -\vdash x : A}\ Axiom$$

$$\frac{x : A, \Gamma; -\vdash t : B}{\Gamma; -\vdash \lambda x.t : A \supset B}\ Right$$

$$\frac{\Gamma; -\vdash t : A \supset B \quad \Gamma; -\vdash u : A \quad \Gamma; B \vdash l : C \quad x : C, \Gamma; -\vdash v : D}{\Gamma; -\vdash t(u, l, (x)v) : D}\ gm - Elim$$

$$\frac{}{\Gamma; C \vdash [] : C}\ Ax$$

$$\frac{\Gamma; -\vdash u : A \quad \Gamma; B \vdash l : C}{\Gamma; A \supset B \vdash u :: l : C}\ Lft$$

with the proviso that $x : A$ *does not belong to* Γ *in* Right *and the proviso that* $x : C$ *does not belong to* Γ *in* gm-Elim*.*

An instance of rule $gm - Elim$ is called a *generalised multiary elimination* (or gm-elimination, for short). [2] explains in which sense these typing rules define a sequent calculus which extends with cuts Schwichtenberg's multiary cut-free sequent calculus [6]. It also explains how to interpret $\lambda\mathbf{J^m}$ in Herbelin's $\overline{\lambda}$-calculus [3], where the key ideia is to interpret a gm-application $t(u, l, (x)v)$ as the combination $v\{x := t(u :: l)\}$ of an head-cut and a mid-cut.

2.2 Reduction rules

Definition 5. *The* reduction rules *for* $\lambda\mathbf{J^m}$ *are as follows:*

$$
\begin{array}{lrcl}
(\beta_1) & (\lambda x.t)(u,[],(y)v) & \to & \mathbf{s}(\mathbf{s}(u,x,t),y,v) \\
(\beta_2) & (\lambda x.t)(u,v::l,(y)v') & \to & \mathbf{s}(u,x,t)(v,l,(y)v') \\
(\pi) & t(u,l,(x)v)(u',l',(y)v') & \to & t(u,l,(x)v(u',l',(y)v')) \\
(\mu) & t(u,l,(x)x(u',l',(y)v)) & \to & t(u,\mathbf{append}(l,u',l'),(y)v), \quad x \notin u',l',v
\end{array}
$$

$$
\begin{array}{lrcl}
\textit{where} & \mathbf{s}(t,x,x) & = & t \\
& \mathbf{s}(t,x,y) & = & y, \ y \neq x \\
& \mathbf{s}(t,x,\lambda y.u) & = & \lambda y.\mathbf{s}(t,x,u) \\
& \mathbf{s}(t,x,u(v,l,(y)v')) & = & \mathbf{s}(t,x,u)(\mathbf{s}(t,x,v),\mathbf{s}'(t,x,l),(y)\mathbf{s}(t,x,v')) \\
& & & \\
& \mathbf{s}'(t,x,[]) & = & [] \\
& \mathbf{s}'(t,x,v::l) & = & \mathbf{s}(t,x,v)::\mathbf{s}'(t,x,l) \\
& & & \\
& \mathbf{append}([],u,l) & = & u::l \\
& \mathbf{append}(u'::l',u,l) & = & u::\mathbf{append}(l',u,l)
\end{array}
$$

A detailed motivation for the reduction rules can be found in [2]. In brief, rules (β_1), (β_2) and (π) perform cut-elimination, *i.e.* they aim at reducing all gm-applications in a term to the form where the head is a variable. Reduction rule (μ) is structural and is used to eliminate gm-applications $t(u,l,(x)v)$ such that $\mathbf{mla}(x,v)$.

Consider the following grammar:

$$
\begin{array}{rcl}
t,u,v & ::= & x \mid \lambda x.t \mid t'(u,l,(y)v) \\
l & ::= & u::l \mid []
\end{array}
$$

The β,π-normal forms are generated by this grammar provided t' is a variable. The μ-normal forms are generated by this grammar provided that in the last production for terms, not $\mathbf{mla}(y,v)$, *i.e.* if v is of the form $y(u',l',(y')v')$, then y must occur either in u', l' or v'. Finally β,π,μ-normal forms are generated by this grammar provided the last production satisfies the two provisos above.

As observed in [2] subject reduction holds for $\to_{\beta,\pi,\mu}$.

2.3 λJ: the generalised λ-calculus

We now introduce the cons-free subsystem of $\lambda\mathbf{J^m}$, called $\lambda\mathbf{J}$.

Definition 6. *Terms and lists of* $\lambda\mathbf{J}$ *are as follows:*

$$
\begin{array}{lrcl}
(\lambda\mathbf{J}-\textit{terms}) & t,u,v & ::= & x \mid \lambda x.t \mid t(u,l,(x)v) \\
(\lambda\mathbf{J}-\textit{lists}) & l & ::= & []
\end{array}
$$

$\mathbf{\Lambda J}$ *is used to denote the set of* $\lambda\mathbf{J}$*-terms.*

Since there is only one form of lists in $\lambda\mathbf{J}$, every gm-application in $\lambda\mathbf{J}$ is of the form $t(u, [], (x)v)$, which we call a *generalised application* (or g-application, for short). $\lambda\mathbf{J}$-terms can simply be described as:

$$(\lambda\mathbf{J} - terms)\ t, u, v ::= x \mid \lambda x.t \mid t(u \cdot (x)v) \quad ,$$

where $t(u \cdot (x)v)$ is used as an abbreviation to $t(u, [], (x)v)$. This expression can be typed by the derived rule (called *generalised elimination*)

$$\frac{\Gamma; - \vdash t : A \supset B \quad \Gamma; - \vdash u : A \quad x : B, \Gamma; - \vdash v : C}{\Gamma; - \vdash t(u \cdot (x)v) : C}\ g - Elim \quad , \tag{1}$$

with proviso $x : B$ does not belong to Γ. Such rule corresponds to an instance of the rule $gm - Elim$ where the penultimate premiss is an instance of Ax.

Definition 7. *The* reduction rules *for* $\lambda\mathbf{J}$ *are as follows:*

$$\begin{array}{rl} (\beta_1) \quad (\lambda x.t)(u \cdot (y)v) & \to \mathbf{s}(\mathbf{s}(u, x, t), y, v) \\ (\pi)\ t(u \cdot (x)v)(u' \cdot (y)v') & \to t(u \cdot (x)v(u' \cdot (y)v')) \end{array}$$

$$\begin{array}{lrcl} \textit{where} & \mathbf{s}(t, x, x) & = & x \\ & \mathbf{s}(t, x, y) & = & y,\ y \neq x \\ & \mathbf{s}(t, x, \lambda y.u) & = & \lambda y.\mathbf{s}(t, x, u) \\ & \mathbf{s}(t, x, u(v \cdot (y)v')) & = & \mathbf{s}(t, x, u)(\mathbf{s}(t, x, v) \cdot (y)\mathbf{s}(t, x, v')) \end{array}$$

Comparatively to $\lambda\mathbf{J^m}$, $\lambda\mathbf{J}$ drops all rules and clauses involving cons. Since β_2-redexes and μ-*contracta* fall outside $\mathbf{\Lambda J}$ (notice that $\mathbf{append}([], u', l')$ is a cons-list), the rules (β_2) and (μ) are omitted.

The system thus obtained is no more than a notational variant of the ΛJ-calculus of Joachimski and Matthes.

3 Relating generality and multiarity

Generality can express multiarity and multiarity is a shorthand for certain forms of generality. In this section this idea is made precise and consequences of it are extracted.

3.1 The bijection between terms of λJ and μ-normal forms

We start by explaining how to express multiarity in terms of generality. The basic idea is to replace each cons by a g-application that introduces a fresh name. For instance,

$$t(u, [u_1, u_2], (x)v) \rightsquigarrow t(u \cdot (z_1)z_1(u_1 \cdot (z_2)z_2(u_2 \cdot (x)v))),$$

where z_1 and z_2 are fresh variables. This idea is embodied in the following type-preserving mapping.

Definition 8. *The mapping ν is as follows.*

$$\nu : \mathbf{\Lambda J^m} \longrightarrow \mathbf{\Lambda J}$$

$$\begin{aligned}
\nu(x) &= x \\
\nu(\lambda x.t) &= \lambda x.\nu(t) \\
\nu(t(u,l,(x)v)) &= \nu(t)(\nu(u)\cdot(z)\nu'(z,l,x,\nu(v))), \quad z \textit{ fresh}
\end{aligned}$$

$$\begin{aligned}
\nu'(z,[],x,v) &= \mathbf{s}(z,x,v) \\
\nu'(z,u::l,x,v) &= z(\nu(u)\cdot(w)\nu'(w,l,x,v)), \quad w \textit{ fresh}
\end{aligned}$$

Conversely, in $t(u,l,(x)v)$, if v is a gm-application $x(u',l',(y)v')$ such that $x \notin u',l',v'$, then v may be eliminated with the help of cons. In fact, the former term can be reduced to $t(u,\mathbf{append}(l,u',l'),(y)v')$, where the append operation generates $u'::l'$ and, if l is not empty, a further cons to concatenate l with $u'::l'$. This is precisely reduction rule μ. The following type-preserving mapping reduces the μ-redexes of a term in a innermost-first fashion.

Definition 9. *The mapping μ is as follows.*

$$\mu : \mathbf{\Lambda J^m} \longrightarrow \mathbf{\Lambda J^m}$$

$$\begin{aligned}
\mu(x) &= x \\
\mu(\lambda x.t) &= \lambda x.\mu(t) \\
\mu(t(u,l,(x)v)) &= \begin{cases} \mu(t)(\mu(u),\mathbf{append}(\mu'(l),u',l'),(y)v'), \\ \qquad\qquad \textit{if } \mu(v) = x(u',l',(y)v') \textit{ and } x \notin u',l',v' \\ \\ \mu(t)(\mu(u),\mu'(l),(x)\mu(v)), \quad \textit{otherwise} \end{cases}
\end{aligned}$$

$$\begin{aligned}
\mu'([]) &= [] \\
\mu'(u::l) &= \mu(u)::\mu'(l)
\end{aligned}$$

The results that follow show that the restriction of mapping μ to $\mathbf{\Lambda J}$ and the restriction of mapping ν to μ-normal forms are mutual inverses.

Lemma 1. *$t \to_\mu^* \mu(t)$, for all $t \in \mathbf{\Lambda J^m}$.*

Proof. Proved together with $l \to_\mu^* \mu'(l)$, for all $l \in \mathcal{L}\mathbf{J^m}$, by simultaneous induction on t and l. □

Lemma 2. *If $t \to_\mu t'$, then (i) $\mu(t) = \mu(t')$ and (ii) $\nu(t) = \nu(t')$, for all $t,t' \in \mathbf{\Lambda J^m}$.*

Proof. (i) is proved together with $l \to_\mu l'$ implies $\mu'(l) = \mu'(l')$, for all $l,l' \in \mathcal{L}\mathbf{J^m}$, by simultaneous induction on $t \to_\mu t'$ and $l \to_\mu l'$. (ii) is proved together with $l \to_\mu l'$ implies $\nu'(z,l,x,v) = \nu'(z,l',x,v)$, for all $l,l' \in \mathcal{L}\mathbf{J^m}$ and all $v \in \mathbf{\Lambda J}$, by simultaneous induction on $t \to_\mu t'$ and $l \to_\mu l'$. □

Lemma 3. *$\mu(t)$ is μ-normal, for all $t \in \mathbf{\Lambda J^m}$.*

Proof. Proved together with $\mu'(l)$ is μ-normal, for all $l \in \mathcal{L}\mathbf{J^m}$, by simultaneous induction on t and l. $\square$

Proposition 1. *(i)* $\to_\mu$ *is confluent.*
(ii) $\to_\mu$ *is strongly normalising.*
(iii) $\mu(t)$ *is the unique normal form of* t *w.r.t.* $\to_\mu$*,* *for all* $t \in \mathbf{\Lambda J^m}$*.*

Proof. (i) follows from lemmas 1 and 2. In order to guarantee (ii), observe that each μ-step reduces the number of μ-redexes. (iii) results from the combination of lemmas 1 and 3 and confluence of $\to_\mu$. $\square$

Lemma 4. $\nu(t) \to_\mu^* t$*, for all* $t \in \mathbf{\Lambda J^m}$*.*

Proof. Proved together with $t(u \cdot (z)\nu'(z,l,x,v)) \to_\mu^* t(u,l,(x)v)$, for all $t,u,v \in \mathbf{\Lambda J}$ and all $l \in \mathcal{L}\mathbf{J^m}$ s.t. $z \notin l,v$, by simultaneous induction on t and l. $\square$

Corollary 1. $t \to_\mu^* \mu(\nu(t))$*, for all* $t \in \mathbf{\Lambda J^m}$*.*

Proof. By Lemma 1, it suffices $\mu(\nu(t)) = \mu(t)$. From Lemma 1 (applied twice) and Lemma 4, $\nu(t)$ reduces both to $\mu(\nu(t))$ and $\mu(t)$, which are μ-normal. Thus by confluence, $\mu(\nu(t)) = \mu(t)$. $\square$

Proposition 2. *(i)* $\nu(t) = t$*, for all* $t \in \mathbf{\Lambda J}$*.*
(ii) $\mu(t) = t$*, for all* μ*-normal* $t \in \mathbf{\Lambda J^m}$*.*

Proof. (i) Follows by induction on t. (ii) Since t is μ-normal, Proposition 1 imposes $t = \mu(t)$. $\square$

Proposition 3. *(i)* $\nu(\mu(t)) = t$*, for all* $t \in \mathbf{\Lambda J}$*.*
(ii) $\mu(\nu(t)) = t$*, for all* μ*-normal* $t \in \mathbf{\Lambda J^m}$*.*

Proof. (i) From lemmas 1 and 2 we get $\nu(\mu(t)) = \nu(t)$, which is just t by the proposition above. (ii) Lemmas 1 and 4 imply reduction of $\nu(t)$ to t and $\mu(\nu(t))$ respectively. Thus t and $\mu(\nu(t))$ are two μ-normal forms of $\nu(t)$, which by confluence of $\to_\mu$ must be equal. $\square$

3.2 Preservation of reduction by mappings μ and ν

Preservation of reduction μ is considered in Lemma 2.

Lemma 5. *(i) If* $t \to_\beta t'$*, then* $\nu(t) \to_\beta \nu(t')$*, for all* $t,t' \in \mathbf{\Lambda J^m}$*.*
(ii) If $t \to_\beta t'$*, then* $\mu(t) \to_\beta \to_\mu^* \mu(t')$*, for all* $t,t' \in \mathbf{\Lambda J^m}$*.*

Proof. (i) is proved together with $l \to_\beta l'$ implies $\nu'(z,l,x,v) \to_\beta \nu'(z,l',x,v)$, for all $l,l' \in \mathcal{L}\mathbf{J^m}$ and all $v \in \mathbf{\Lambda J}$, by simultaneous induction on $t \to_\beta t'$ and $l \to_\beta l'$. (ii) follows from the commutation in $\lambda\mathbf{J^m}$ between $\to_\beta$ and $\to_\mu$: if $t \to_\beta t_1$ and $t \to_\mu t_2$, there exists t_3 such that $t_1 \to_\mu^* t_3$ and $t_2 \to_\beta t_3$. $\square$

In contrast to rule (β), one-to-one preservation of π-steps is problematic: mapping ν needs several steps in $\lambda\mathbf{J}$ to simulate a single step in $\lambda\mathbf{J^m}$ and mapping μ does not even preserve π-steps. These mismatches, between rule (π) and mappings ν and μ, are an obstacle to proving confluence of $\lambda\mathbf{J^m}$ along the lines of the proof of Theorem 5, where we lift confluence of $\lambda\mathbf{J}$. Such proof requires preservation of (π) (as well as (β)) by mapping μ. We illustrate these mismatches with an example.

Let t, u, u_1, u_2, u', v be μ-normal forms in $\lambda\mathbf{J}$, hence invariant both for μ and ν. Consider the following three terms in $\lambda\mathbf{J}$

$$
\begin{aligned}
t_0 &= t(u\cdot(z_1)z_1(u_1\cdot(z_2)z_2(u_2\cdot(x)x)))(u'\cdot(y)v) \ , \\
t_1 &= t(u\cdot(z_1)z_1(u_1\cdot(z_2)z_2(u_2\cdot(x)x))(u'\cdot(y)v)) \ , \\
t_2 &= t(u\cdot(z_1)z_1(u_1\cdot(z_2)z_2(u_2\cdot(x)x(u'\cdot(y)v)))) \ ,
\end{aligned}
$$

and the corresponding μ-normal forms

$$
\begin{aligned}
u_0 &= \mu(t_0) = t(u,[u_1,u_2],(x)x)(u'\cdot(y)v) \ , \\
u_1 &= \mu(t_1) = t(u\cdot(z_1)z_1(u_1,[u_2],(x)x)(u'\cdot(y)v)) \ , \\
u_2 &= \mu(t_2) = t(u,[u_1,u_2,u'],(y)v) \ .
\end{aligned}
$$

Consider also

$$
\begin{aligned}
v_1 &= t(u\cdot(z_1)z_1(u_1,[u_2],(x)x(u'\cdot(y)v))) \ , \\
v_2 &= t(u,[u_1,u_2],(x)x(u'\cdot(y)v)) \ .
\end{aligned}
$$

Observe that $\nu(u_0) = t_0$, $\nu(u_1) = t_1$ and $\nu(u_2) = \nu(v_1) = \nu(v_2) = t_2$. Observe also that there are the following reductions among these terms:

$$
\begin{tikzcd}
u_0 \arrow[ddrr, "\pi"] & & & t_0 \arrow[lll, two heads, "\mu"'] \arrow[d, "\pi"] \\
u_1 \arrow[dr, "\pi"'] & & & t_1 \arrow[lll, two heads, "\mu"'] \arrow[d, "\pi"] \\
& v_1 \arrow[r, "\mu"'] & v_2 \arrow[dll, "\mu"] & \cdot \arrow[d, "\pi"] \\
u_2 & & & t_2 \arrow[lll, two heads, "\mu"']
\end{tikzcd}
$$

Notice that $u_0 \to_\pi v_2$ whereas $\nu(u_0)$ requires three π-steps to reach $\nu(v_2)$. In general we have the following:

Lemma 6. *If $t \to_\pi t'$, then $\nu(t) \to_\pi^+ \nu(t')$, for all $t, t' \in \mathbf{\Lambda J^m}$.*

Proof. Proved together with $l \to_\pi l'$ implies $\nu'(z,l,x,v) \to_\pi^+ \nu'(z,l',x,v)$, for all $l, l' \in \mathcal{L}\mathbf{J^m}$ and all $v \in \mathbf{\Lambda J}$, by simultaneous induction on $t \to_\pi t'$ and $l \to_\pi l'$. $\square$

Going back to the example, observe that $t_0 \rightarrow_\pi t_1$ but $\mu(t_0)$ does not reduce to $\mu(t_1)$, it π-reduces to v_2. However, making enough π-reductions from t_1, one reaches a term (t_2 in the example) whose μ-normal form (u_2 in the example) is the same as the μ-normal form of v_2. Making enough π-reductions means to perform π-reductions as long as this generates π-redexes which hide μ-redexes. For instance, observe that the head of t_0 is a μ-redex. The reduction $t_0 \rightarrow_\pi t_1$ creates the π-redex $z_1(u_1 \cdot (z_2) z_2(u_2 \cdot (x)x))(u' \cdot (y)v)$ which hides in t_1 the mentioned μ-redex. Since the reduction of this π-redex causes a descendent of the original μ-redex to reappear, we perform it. Moreover, as another μ-redex becomes hidden, this process continues. We introduce a new reduction rule in $\lambda\mathbf{J}$ to perform such sequences of π-reductions in a single step.

Definition 10. *The rule* (π') *is the following:*

$$(\pi') \;\; t(u \cdot (x)v)(u' \cdot (y)v') \rightarrow t(u \cdot (x)@'(x, v, u', y, v'))$$

where

$$@'(x, t, u, y, v) \; = \; \begin{cases} x(u' \cdot (z)@'(z, v', u, y, v)), \\ \qquad\qquad \textit{if } t = x(u' \cdot (z)v') \textit{ and } x \notin u', v' \\ \\ t(u \cdot (y)v), \quad \textit{otherwise} \end{cases}$$

For instance, in the example before $t_1 \rightarrow_{\pi'} t_2$. Observe that $\rightarrow_{\pi'} \subseteq \rightarrow^+_\pi$ and that a term is π'-normal if and only if it is π-normal.

We now see how the situation improved w.r.t. the preservation of π-steps.

Lemma 7. *(i) If* $t \rightarrow_\pi t'$*, then* $\nu(t) \rightarrow_{\pi'} \nu(t')$*,*
for all $t, t' \in \mathbf{\Lambda J^m}$ *such that* t *is* μ*-normal.*
(ii) If $t \rightarrow_{\pi'} t'$*, then* $\mu(t) \rightarrow_\pi \rightarrow^*_\mu \mu(t')$*, for all* $t, t' \in \mathbf{\Lambda J}$*.*

Proof. (ii) is proved by induction on $t \rightarrow_{\pi'} t'$. (i) is proved together with $l \rightarrow_\pi l'$ implies $\nu'(z, l, x, v) \rightarrow_{\pi'} \nu'(z, l', x, v)$, for all $l, l' \in \mathcal{L}\mathbf{J^m}$ and all $v \in \mathbf{\Lambda J}$, by simultaneous induction on $t \rightarrow_\pi t'$ and $l \rightarrow_\pi l'$. $\square$

Now we turn to some basic results about rule (π'), leading to Corollary 2, which shows how to perform a sequence of (β) and (π) reductions by means of a sequence of (β) and (π') reductions. The proof of confluence of the relation $\rightarrow_{\beta,\pi,\mu}$ on $\lambda\mathbf{J^m}$-terms, given in Section 4, uses this transformation and the lemma above.

The mapping π in the definition below is considered in [4] and produces π-normal forms.

Definition 11. *The mapping π is as follows.*

$$\pi : \mathbf{\Lambda J} \longrightarrow \mathbf{\Lambda J}$$

$$\begin{aligned} \pi(x) &= x \\ \pi(\lambda x.t) &= \lambda x.\pi(t) \\ \pi(t(u{\cdot}(x)v)) &= @(\pi(t), \pi(u), x, \pi(v)) \end{aligned}$$

where

$$@(t, u, x, v) = \begin{cases} t'(u'{\cdot}(y)@(v', u, x, v)), & \textit{if } t = t'(u'{\cdot}(y)v') \\ \\ t(u{\cdot}(x)v), & \textit{otherwise} \end{cases}$$

[4] observes that (i) $t \rightarrow^*_\pi \pi(t)$, for all $t \in \mathbf{\Lambda J}$; (ii) if $t \rightarrow^*_\pi t'$ then $\pi(t) = \pi(t')$ for all $t, t' \in \mathbf{\Lambda J}$ (and from these two follows confluence of $\rightarrow_\pi$); (iii) $\rightarrow_\pi$ is strongly normalising for all terms of $\lambda\mathbf{J}$.

Next lemma establishes that rule (π') suffices to reduce a term to its π-normal form.

Lemma 8. *$t \rightarrow^*_{\pi'} \pi(t)$, for all $t \in \mathbf{\Lambda J}$.*

Proof. Because $\rightarrow_{\pi'} \subseteq \rightarrow^+_\pi$ and $\rightarrow_\pi$ is terminating, $\rightarrow_{\pi'}$ is also terminating. Let t' be a π'-normal form of t. Since t' is also a π-normal form, $t \rightarrow^*_\pi t'$ and $\rightarrow_\pi$ is confluent, it follows that $t' = \pi(t)$. Thus $t \rightarrow^*_{\pi'} \pi(t)$. $\square$

We establish now a kind of commutation between reduction $\rightarrow_\beta$ and mapping π, that uses next lemma.

Lemma 9. *$\mathbf{s}(\pi(t), x, \pi(u)) \rightarrow^*_\pi \pi(\mathbf{s}(t, x, u))$, for all $t, u \in \mathbf{\Lambda J}$.*

Proof. The proof is by induction on u. It uses the fact that, for all $t, t_0, u_0, v_0 \in \mathbf{\Lambda J}$, $\mathbf{s}(t, x, @(t_0, u_0, y, v_0)) \rightarrow^*_\pi @(\mathbf{s}(t, x, t_0), \mathbf{s}(t, x, u_0), y, \mathbf{s}(t, x, v_0))$, proved by induction on t_0 $\square$

Proposition 4. *If $t \rightarrow_\beta u$, then $\pi(t) \rightarrow^*_{\beta,\pi'} \pi(u)$, for all $t, u \in \mathbf{\Lambda J}$.*

Proof. By induction on $t \rightarrow_\beta u$. The base case uses the lemma before. $\square$

Corollary 2. *If $t \rightarrow^*_{\beta,\pi} u$, then $\pi(t) \rightarrow^*_{\beta,\pi'} \pi(u)$, for all $t, u \in \mathbf{\Lambda J}$.*

Proof. Follows by induction on the number of steps in the reduction sequence. The case corresponding to a β-step uses the proposition before and the case corresponding to a π-step uses invariance of $\rightarrow_\pi$ w.r.t. mapping π. $\square$

3.3 Two isomorphic subsystems of $\lambda\mathbf{J^m}$

Some of the preservation results obtained above can be put together so that the bijection between μ-normal forms and terms of $\lambda\mathbf{J}$ becomes an isomorphism, provided those two sets of terms are equipped with appropriate reduction relations.

Let $\lambda\mathbf{J}'$ denote the system obtained from $\lambda\mathbf{J}$ replacing rule (π) by rule (π'). Let $\lambda\mathbf{J}^{\mathbf{m}}_{\mu}$ denote the subsystem of μ-normal forms of $\lambda\mathbf{J}^{\mathbf{m}}$ obtained by closing relation $\rightarrow_{\beta,\pi}$ for mapping μ. More precisely, in $\lambda\mathbf{J}^{\mathbf{m}}_{\mu}$ the one step relations $\rightarrow_{\beta_\mu}$ and $\rightarrow_{\pi_\mu}$ are given by:

$$\begin{array}{lll} t \rightarrow_{\beta_\mu} t' & \text{if} \quad t \rightarrow_{\beta} t'' \text{ and } t' = \mu(t''), & \text{for some } t'' \in \lambda\mathbf{J}^{\mathbf{m}}; \\ t \rightarrow_{\pi_\mu} t' & \text{if} \quad t \rightarrow_{\pi} t'' \text{ and } t' = \mu(t''), & \text{for some } t'' \in \lambda\mathbf{J}^{\mathbf{m}}. \end{array}$$

Notice that in $\lambda\mathbf{J}^{\mathbf{m}}_{\mu}$ there is no need for a μ-reduction.

Theorem 1. *(i)* $t \rightarrow_{\beta_\mu} t'$ *iff* $\nu(t) \rightarrow_{\beta} \nu(t')$, *for all μ-normal forms* t, t'.
(ii) $t \rightarrow_{\pi_\mu} t'$ *iff* $\nu(t) \rightarrow_{\pi'} \nu(t')$, *for all μ-normal forms* t, t'.
(iii) $t \rightarrow_{\beta} t'$ *iff* $\mu(t) \rightarrow_{\beta_\mu} \mu(t')$, *for all* $t, t' \in \mathbf{\Lambda J}$.
(iv) $t \rightarrow_{\pi'} t'$ *iff* $\mu(t) \rightarrow_{\pi_\mu} \mu(t')$, *for all* $t, t' \in \mathbf{\Lambda J}$.

Proof. We just show the "only if" statements since the "if" statements follow from these and the fact that ν and μ are mutual inverses. (i) follows from lemmas 1, 2 and 5. (ii) follows from lemmas 1, 2 and 7. (iii) and (iv) hold by lemmas 5 and 7 respectively. □

Now confluence and strong normalisation of relation $\rightarrow_{\beta,\pi}$ on $\lambda\mathbf{J}$ are used to obtain corresponding properties for $\lambda\mathbf{J}'$ and thus for its isomorphic system $\lambda\mathbf{J}^{\mathbf{m}}_{\mu}$.

Theorem 2. $\rightarrow_{\beta,\pi'}$ *in* $\lambda\mathbf{J}'$ *is confluent.*

Proof. Assume $t \rightarrow^*_{\beta,\pi'} t_1$ and $t \rightarrow^*_{\beta,\pi'} t_2$. Then, since $\rightarrow_{\pi'} \subseteq \rightarrow^+_{\pi}$, also $t \rightarrow^*_{\beta,\pi} t_1$ and $t \rightarrow^*_{\beta,\pi} t_2$. Using confluence of $\rightarrow^*_{\beta,\pi}$ for $\lambda\mathbf{J}$, there exists t_3 such that $t_1 \rightarrow^*_{\beta,\pi} t_3$ and $t_2 \rightarrow^*_{\beta,\pi} t_3$. So, using Corollary 2 followed by Lemma 8, one obtains $t_1 \rightarrow^*_{\beta,\pi'} \pi(t_3)$ and $t_2 \rightarrow^*_{\beta,\pi'} \pi(t_3)$. □

Theorem 3. *There is no infinite* $\rightarrow_{\beta,\pi'}$*-reduction starting at a typable term of* $\lambda\mathbf{J}'$.

Proof. If there was, since $\rightarrow_{\pi'} \subseteq \rightarrow^+_{\pi}$, one could build an infinite sequence of β, π-steps, starting at a typable term of $\lambda\mathbf{J}$, contradicting strong normalisation of $\lambda\mathbf{J}$. □

4 Results of confluence and strong normalisation for $\lambda\mathbf{J}^{\mathbf{m}}$

This section studies confluence and strong normalisation for the notions of reduction in $\lambda\mathbf{J}^{\mathbf{m}}$ resulting from all possible combinations of rules (β), (π) and (μ). The proofs of confluence presented here follow one of two directions: (i) for notions of reduction involving only rules (β) and (π), arguments are simple extensions of those used in [4]; (ii) for notions of reduction including μ, arguments are built in a modular way, using essentially properties of presevation of reduction by mappings μ and ν, together with confluence results for $\lambda\mathbf{J}$. Strong normalisation of $\rightarrow_{\beta,\pi,\mu}$ for all typable terms of $\lambda\mathbf{J^m}$ is obtained from the strong normalisation of $\rightarrow_{\beta,\pi}$ for $\lambda\mathbf{J}$'s typable terms, with the help of results of preservation of reduction by mapping ν. Strong normalisation of typable terms for all the other relations follows, since they are included in $\rightarrow_{\beta,\pi,\mu}$. In fact, for relations not involving rule (β), strong normalisation holds for all terms.

4.1 Confluence

Firstly we tackle confluence of relations $\rightarrow_{\pi}$, $\rightarrow_{\beta}$ and $\rightarrow_{\beta,\pi}$ in $\lambda\mathbf{J^m}$. The following definition extends Definition 11.

Definition 12. *The mapping π is as follows.*

$$\pi : \mathbf{\Lambda J^m} \longrightarrow \mathbf{\Lambda J^m}$$

$$\begin{aligned}
\pi(x) &= x \\
\pi(\lambda x.t) &= \lambda x.\pi(t) \\
\pi(t(u,l,(x)v)) &= @(\pi(t),\pi(u),\pi'(l),x,\pi(v)) \\
\\
\pi'([]) &= [] \\
\pi'(u::l) &= \pi(u)::\pi'(l)
\end{aligned}$$

where

$$@(t,u,l,x,v) = \begin{cases} t'(u',l',(y)@(v',u,l,x,v)), & \text{if } t = t'(u',l',(y)v') \\ \\ t(u,l,(x)v), & \text{otherwise} \end{cases}$$

Lemma 10. *$\pi(t)$ is π-normal, for all $t \in \mathbf{\Lambda J^m}$.*

Proof. Proved together with $\pi'(l)$ is π-normal, for all $l \in \mathcal{L}\mathbf{J^m}$, by simultaneous induction on t and l. ☐

Lemma 11. *$t \rightarrow_{\pi}^{*} \pi(t)$, for all $t \in \mathbf{\Lambda J^m}$.*

Proof. Proved together with $l \rightarrow_{\pi}^{*} \pi'(l)$, for all $l \in \mathcal{L}\mathbf{J^m}$, by simultaneous induction on t and l. ☐

Lemma 12. *If $t_1 \rightarrow_{\pi}^{*} t_2$, then $\pi(t_1) = \pi(t_2)$, for all $t_1, t_2 \in \mathbf{\Lambda J^m}$.*

Proof. Proved together with the fact that $l_1 \rightarrow_{\pi}^{*} l_2$ implies $\pi'(l_1) = \pi'(l_2)$, for all $l_1, l_2 \in \mathcal{L}\mathbf{J^m}$, by simultaneous induction on $t \rightarrow_{\pi}^{*} t'$ and $l \rightarrow_{\pi}^{*} l'$. ☐

Proposition 5. *$\rightarrow_{\pi}^{*}$ has the triangle property w.r.t. mapping π.*[1]

Proof. If $t_1 \rightarrow_{\pi}^{*} t_2$, from the two lemmas above, $t_2 \rightarrow_{\pi}^{*} \pi(t_2) = \pi(t_1)$. ☐

Definition 13. *Reduction $\Rightarrow_{\beta}$ is inductively defined on terms of $\lambda\mathbf{J^m}$ as follows:*

[1] A relation $\rightarrow$ has the triangle property w.r.t. a function f if $a \rightarrow b$ implies $b \rightarrow f(a)$

$x \Rightarrow_\beta x$;
$\lambda x.t \Rightarrow_\beta \lambda x.t'$ *if* $t \Rightarrow_\beta t'$;
$t(u, l, (x)v) \Rightarrow_\beta t'(u', l', (x)v')$ *if* $t \Rightarrow_\beta t', u \Rightarrow_\beta u', l \Rightarrow_\beta l', v \Rightarrow_\beta v'$;
$(\lambda y.t)(u, [], (x)v) \Rightarrow_\beta \mathbf{s}(\mathbf{s}(u', y, t'), x, v')$ *if* $t \Rightarrow_\beta t', u \Rightarrow_\beta u', v \Rightarrow_\beta v'$;
$(\lambda y.t)(u, u_0 :: l, (x)v) \Rightarrow_\beta \mathbf{s}(u', y, t')(u_0', l', (x)v')$ *if*
$t \Rightarrow_\beta t', u \Rightarrow_\beta u', u_0 \Rightarrow_\beta u_0', l \Rightarrow_\beta l', v \Rightarrow_\beta v'$;
$[] \Rightarrow_\beta []$;
$u :: l \Rightarrow_\beta u' :: l'$ *if* $u \Rightarrow_\beta u', l \Rightarrow_\beta l'$.

Observe that $\Rightarrow_\beta$ is reflexive and $\rightarrow_\beta \subseteq \Rightarrow_\beta \subseteq \rightarrow_\beta^*$.

Definition 14. *The mapping β is as follows.*

$$\beta : \mathbf{\Lambda J^m} \longrightarrow \mathbf{\Lambda J^m}$$
$$\begin{aligned} x^\beta &= x \\ (\lambda x.t)^\beta &= \lambda x.t^\beta \\ t(u, l, (x)v)^\beta &= \begin{cases} \mathbf{s}(\mathbf{s}(u^\beta, y, t_1^\beta), x, v^\beta), \\ \qquad\qquad \textit{if } t = \lambda y.t_1 \textit{ and } l = [] \\ \mathbf{s}(u^\beta, y, t_1^\beta)(u_1^\beta, l_1^{\beta'}, (x)v^\beta), \\ \qquad\qquad \textit{if } t = \lambda y.t_1 \textit{ and } l = u_1 :: l_1 \\ t^\beta(u^\beta, l^{\beta'}, (x)v^\beta), \quad \textit{otherwise} \end{cases} \end{aligned}$$

$$\begin{aligned} []^{\beta'} &= [] \\ (u :: l)^{\beta'} &= u^\beta :: l^{\beta'} \end{aligned}$$

Proposition 6. $\Rightarrow_\beta$ *has the triangle property w.r.t.* β.

Proof. By induction on $\Rightarrow_\beta$. It uses parallelism of $\Rightarrow_\beta$, *i.e.* the fact that if $t \Rightarrow_\beta t'$ and $u \Rightarrow_\beta u'$ then $\mathbf{s}(t, x, u) \Rightarrow_\beta \mathbf{s}(t', x, u')$, as well as simple inversion principles for $\Rightarrow_\beta$. □

Lemma 13. *If* $t \Rightarrow_\beta t_1$ *and* $t \rightarrow_\pi t_2$, *then* $t_1 \rightarrow_\pi^* t_3$ *and* $t_2 \Rightarrow_\beta t_3$, *for some* $t_3 \in \mathbf{\Lambda J^m}$.

Proof. Proved together with the fact that if $l \Rightarrow_\beta l_1$ and $l \rightarrow_\pi l_2$, then there exists $l_3 \in \mathcal{L}\mathbf{J^m}$ such that $l_1 \rightarrow_\pi^* l_3$ and $l_2 \Rightarrow_\beta l_3$, for all $l, l_1, l_2 \in \mathcal{L}\mathbf{J^m}$, by simultaneous induction on $t \Rightarrow_\beta t_1$ and $l \Rightarrow_\beta l_1$. This proof uses parallelism of $\rightarrow_\pi^*$. □

Corollary 3. $\Rightarrow_\beta$ *and* $\rightarrow_\pi^*$ *commute.*

Proof. Follows from the previous lemma by a simple diagram chase. □

Proposition 7. $\Rightarrow_\beta \rightarrow_\pi^*$ *has the triangle property w.r.t.* $\pi \circ \beta$.

Proof. Follows from the triangle properties of $\rightarrow_\pi^*$ and $\Rightarrow_\beta$ w.r.t. π and β, together with commutativity between the two relations. □

Theorem 4. $\rightarrow_\pi$, $\rightarrow_\beta$ *and* $\rightarrow_{\beta,\pi}$ *are confluent.*

Proof. Confluence of a relation can be obtained from a triangle property, as shown in Lemma 1 of [4]. (Confluence of $\to_\pi$ can also be obtained immediately from lemmas 11 and 12.) As to confluence of $\to_{\beta,\pi}$, observe that $\to^*_{\beta,\pi}$ is confluent and that the reflexive and transitive closure of $\Rightarrow_\beta\to^*_\pi$ is equal to $\to^*_{\beta,\pi}$, since $\to_{\beta,\pi}\subseteq\Rightarrow_\beta\to^*_\pi\subseteq\to^*_{\beta,\pi}$. □

Now we consider confluence in the presence of rule μ. The method used before still works when one adjoins rule μ, because: (i) $\to^*_{\pi,\mu}$ has a triangle property (w.r.t. $\mu\circ\pi$); and (ii) $\to^*_\mu$ commutes with $\Rightarrow_\beta$. However, in the presence of rule μ, one can lift confluence results of $\lambda\mathbf{J}$.

Theorem 5. *$\to_{\beta,\pi,\mu}$, $\to_{\beta,\mu}$ and $\to_{\pi,\mu}$ are confluent.*

Proof. Let R be relation β (resp. π or $\beta\cup\pi$) and let R' be β (resp. π' or $\beta\cup\pi'$). Assume $t\to^*_{R,\mu} t_1$ and $t\to^*_{R,\mu} t_2$. Then, by lemmas 2, 5 and 6 it follows that $\nu(t)\to^*_R\nu(t_1)$ and $\nu(t)\to^*_R\nu(t_2)$. Now confluence of R in $\lambda\mathbf{J}$ guarantees the existence of t_3 such that $\nu(t_1)\to^*_R t_3$ and $\nu(t_2)\to^*_R t_3$. So, using Lemma 8 and Corollary 2, $\nu(t_1)\to^*_{R'}\pi(t_3)$ and $\nu(t_2)\to^*_{R'}\pi(t_3)$, which in turn, by lemmas 5 and 7, implies $\mu(\nu(t_1))\to^*_{R,\mu}\mu(\pi(t_3))$ and $\mu(\nu(t_2))\to^*_{R,\mu}\mu(\pi(t_3))$. Then, from Corollary 1, it follows $t_1\to^*_\mu\mu(\nu(t_1))$ and $t_2\to^*_\mu\mu(\nu(t_2))$ and thus t_1 and t_2 have $\mu(\pi(t_3))$ as common reduct. □

4.2 Strong Normalisation

Theorem 6. *There is no infinite $\to_{\beta,\pi,\mu}$-reduction sequence starting at a typable term of $\lambda\mathbf{J^m}$.*

Proof. Suppose there is such an infinite reduction sequence S. It cannot contain infinitely many β,π-steps. Otherwise, since (i) μ-reduction is invariant under ν (Lemma 2), (ii) each β,π-step in $\lambda\mathbf{J^m}$ originates under ν one or more β,π-steps in $\lambda\mathbf{J}$ (lemmas 5 and 6) and (iii) ν preserves typability, one could build in $\lambda\mathbf{J}$ an infinite sequence of β,π-steps starting at a typable term, contradicting strong normalisation of $\lambda\mathbf{J}$. Therefore beyond a certain point in sequence S there are solely μ-steps, necessarily in infinite number, which is also impossible due to strong normalisation of $\to_\mu$ (Proposition 1). □

Theorem 7. *There is no infinite $\to_{\pi,\mu}$-reduction sequence in $\lambda\mathbf{J^m}$.*

Proof. Similar to the one above showing strong normalisation of $\to_{\beta,\pi,\mu}$. Additionally, one just needs to observe that $\to_\pi$ in $\lambda\mathbf{J}$ is strongly normalising. □

5 Conclusion

This work shows that the reduction relations of $\lambda\mathbf{J^m}$ enjoy strong normalisation of typable terms and confluence. As such $\lambda\mathbf{J^m}$ is a well-behaved extension of the λ-calculus and we intend to explore its potential in functional programming. On the other hand, as shown in [2], $\lambda\mathbf{J^m}$ captures as subsystems, not only the system ΛJ of Joachimski and Matthes, but also the multiary λ-calculus $\lambda\mathcal{P}\mathbf{h}$ [1], as well as a notational variant of λ-calculus. So, we consider $\lambda\mathbf{J^m}$ a useful tool for the computational interpretation of successively stronger fragments of sequent calculus, deserving further study in this direction.

Our investigations of the relationship between generality and multiarity identify two isomorphic subsystems of $\lambda\mathbf{J^m}$: (i) a variant of $\lambda\mathbf{J}$, which is the subsystem with minimal use of multiarity (*i.e.* no use); (ii) the subsystem of μ-normal forms, which is the subsystem with maximal use of multiarity (*i.e.* uses cons for expressing generality whenever possible). Think of $t \in \lambda\mathbf{J}$ and of all its μ-reduction sequences, leading to $\mu(t)$. In a sense, all the terms involved in these reduction sequences are representations of the same term, ranging from the term t with minimal use of multiarity to the term $\mu(t)$ with maximal use of multiarity, going through intermediate terms that do not belong to the subsystems: t and $\mu(t)$ are canonical representations whereas the intermediate terms are a redundancy allowed in $\lambda\mathbf{J^m}$. Thus the two isomorphic subsystems are non-redundant opposite extremes w.r.t. the use of multiarity.

However both subsystems have shortcomings because of this extreme nature. In the former, multiarity is not available as a shorthand. In the latter, it is a simple definition of expressions and reduction that is not available, because unconstrained gm-application, as well as β- and π-reduction, can create μ-redexes, *i.e.* do not preserve maximal multiarity. Although exhibiting some redundancy, $\lambda\mathbf{J^m}$ does not suffer from the drawbacks of these subsystems. Therefore it seems to be the system with the right use of multiarity.

Acknowledgment: Diagram in Subsection 3.2 was produced with Paul Taylor's macros.

References

1. J. Espírito Santo, *An isomorphism between a fragment of sequent calculus and an extension of natural deduction*, *in: M. Baaz and A. Voronkov (Eds.), Proc. of LPAR'02*, 2002, Springer-Verlag, LNAI vol. **2514**, 354–366.
2. J. Espírito Santo and L. Pinto, *Permutative conversions in intuitionistic multiary sequent calculus with cuts*, *in: M. Hoffman (Ed.), Proc. of TLCA*, 2003, Springer-Verlag, LNCS vol. **2701**, 286–300.
3. H. Herbelin, *A λ-calculus structure isomorphic to a Gentzen-style sequent calculus structure*, *in: L. Pacholski and J. Tiuryn (Eds.), Proceedings of CSL'94*, 1995, Springer-Verlag, LNCS vol. **933**, 61–75.
4. F. Joachimski and R. Matthes, *Standardization and confluence for a Lambda Calculus with generalized applications*, *in: L. Bachmair (Ed.), Proc. of RTA*, 2000, Springer-Verlag, LNCS vol. **1833**, 141–155.
5. F. Joachimski and R. Matthes, *Short proofs of normalisation for the simply typed λ-calculus, permutative conversions and Gödel's* **T**, *Archive for Mathematical Logic*, **42** (2003) 59–87.
6. H. Schwichtenberg, *Termination of permutative conversions in intuitionistic Gentzen calculi*, *Theoretical Computer Science*, **212** (1999) 247–260.
7. J. von Plato, *Natural deduction with general elimination rules*, *Archive for Mathematical Logic*, **40** (2001) 541–567.