

# A calculus of multiary sequent terms

JOSÉ ESPÍRITO SANTO

and

LUÍS PINTO

Universidade do Minho

---

Multiary sequent terms were originally introduced as a tool for proving termination of permutative conversions in cut-free sequent calculus. This work develops the language of multiary sequent terms into a term calculus for the computational (Curry-Howard) interpretation of a fragment of sequent calculus with cuts and cut-elimination rules. The system, named *generalised multiary  $\lambda$ -calculus*, is a rich extension of the  $\lambda$ -calculus where the computational content of the sequent calculus format is explained through an enlarged form of the application constructor. Such constructor exhibits the features of multiarity (the ability of forming lists of arguments) and generality (the ability of prescribing a kind of continuation). The system integrates in a modular way the multiary  $\lambda$ -calculus and an isomorphic copy of the  $\lambda$ -calculus with generalised application  $\Lambda J$  (in particular, natural deduction is captured internally up to isomorphism). In addition, the system: (i) comes with permutative conversion rules, whose role is to eliminate the new features of application; (ii) is equipped with reduction rules — either the  $\mu$ -rule, typical of the multiary setting, or rules for cut-elimination, which enlarge the ordinary  $\beta$ -rule. This paper establishes the meta-theory of the system, with emphasis on the role of the  $\mu$ -rule, and including a study of the interaction of reduction and permutative conversions.

Categories and Subject Descriptors: F.4.1 [Mathematical Logic and Formal Languages]: Mathematical Logic—Proof theory; Lambda calculus and related systems

General Terms: Languages, Theory

Additional Key Words and Phrases: intuitionistic sequent calculus, lambda-calculus, Curry-Howard isomorphism, generalised application, multiary application, permutative conversions

---

## 1. INTRODUCTION

**Motivation.** It is well-known that two intuitionistic sequent calculus derivations determine the same natural deduction proof when they are inter-permutable [Zucker 1974; Pottinger 1977], that is, when they differ only by certain permutations in the order of application of inference rules. In [Dyckhoff and Pinto 1999] this idea is made precise for *cut-free* sequent calculus by the identification of a basic set of *permutative conversion* rules and the definition of a confluent and weakly normalising rewriting system whose normal forms are in 1-1 correspondence with the normal natural deductions.

Schwichtenberg proved in [Schwichtenberg 1999] that a variant of the rewriting system of [Dyckhoff and Pinto 1999] is strongly normalising. This variant has the characteristic of being “multiary”. *Multiarity* has a meaning at the level of the proof

---

Authors' address: Centro de Matemática, Universidade do Minho, 4710-057, Braga, Portugal.

©ACM (2011) This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in ACM Transactions on Computational Logic, Vol. 12, No. 3, Article 22 (May 2011) <http://doi.acm.org/10.1145/1929954.1929959>

system, and at the level of the language of proof annotations (the so-called *multiary sequent terms*). At the level of the proof system, it means that the left introduction rule allows in a single inference the introduction of  $A_1 \supset \dots \supset A_k \supset B$  from  $k + 1$  premises ( $k$  of them relative to the  $A_i$ 's and a last one relative to  $B$ )<sup>1</sup>. In fact, one has a family of rules, indexed by  $k \geq 1$ . At the level of annotations, it means that one needs an auxiliary class of expressions, named “paths” in [Schwichtenberg 1999], to collect the  $k$  annotations of the  $k$  premisses relative to the  $A_i$ 's mentioned before.

The starting point of this work is a new attitude towards the language of multiary sequent terms. While this language is regarded in [Schwichtenberg 1999] as a technical tool in order to achieve a certain termination result, here it is considered as the basis for a computational interpretation of (some suitable fragment) of sequent calculus, along the lines of the Curry-Howard isomorphism. A subsidiary motivation for this work is the possibility of an alternative implementation of the multiarity feature, in the style of Herbelin's proof system *LJT* and the corresponding  $\bar{\lambda}$ -calculus [Herbelin 1995]. In *LJT* one does not have a family of left introduction rules, but instead an auxiliary kind of sequents with a “stoup” formula; and in  $\bar{\lambda}$  one does not speak of “paths” but instead of *lists* of terms<sup>2</sup>. The sharing of syntactic ingredients between multiary sequent terms and  $\bar{\lambda}$ -terms gave an extra motivation for developing the former as some variant of the  $\lambda$ -calculus.

**The  $\lambda J^m$ -calculus.** In this paper we extend the proof system of [Schwichtenberg 1999] with a special form of cuts and cut-elimination rules, and, accordingly, we adapt the language of multiary sequent terms and equip it with reduction rules. The result, presented as a typing system for some extension of the  $\lambda$ -calculus, is as good as could be expected, from the point of view of both a smooth extension of the cut-free fragment, and of the obtention of a meaningful computational interpretation. Indeed, the cut-free multiary sequent terms consisted of variables,  $\lambda$ -abstractions, and a third construction, written here as  $y(u, l, (x)v)$  (with the *function expression*  $y$  a variable), corresponding to left introductions. Our extension simply means to enlarge this construction (the function expression is allowed to be an arbitrary term) so that cuts are encompassed, and to interpret the resulting construction as a form of application, named *generalised multiary application* (or gm-application), and written thus:

$$t(u, l, (x)v) . \tag{1}$$

Here  $t$  represents the left premiss of a cut, whereas  $(u, l, (x)v)$  represents its right premiss, which is necessarily a multiary left introduction. The cut-formula, necessarily an implication, is both the type of  $t$  and the formula left introduced by  $(u, l, (x)v)$ . Without surprise, cuts of this form resemble an *elimination rule* of natural deduction, and indeed our cuts may be seen as a multiary extension of von Plato's concept of general elimination [von Plato 2001].

<sup>1</sup>In this paper, contrary to [Schwichtenberg 1999], we only deal with implication  $\supset$ .

<sup>2</sup>Curiously, while in [Herbelin 1995] an interpretation for lists as “applicative contexts” is sketched, here we will regard lists as such (and more precisely, the possibility of forming functional application with lists of arguments) as a direct and literal manifestation, at the term level, of the multiarity feature of the proof-system.

In the computational reading,  $t$  is the function expression and  $(u, l, (x)v)$  is the *gm-argument* of the gm-application. This consists of the first argument  $u$ , a list  $l$  of extra arguments and the formal parameter  $x$  and the body  $v$  of an explicit substitution. The availability of list  $l$  qualifies the application as multiary; the availability of a kind of “continuation”  $(x)v$  qualifies the application as *generalised*, the terminology used in the  $\Lambda J$ -calculus [Joachimski and Matthes 2003]<sup>3</sup>.

Moreover, and this is the main point of the computational interpretation, the reading of  $t(u, l, (x)v)$  as an enlarged notion of application induces a transparent, coherent functional reading for the cut-elimination rules. For instance, there is a reduction rule for eliminating cuts as above where the left premiss  $t$  ends with a right introduction (is a  $\lambda$ -abstraction), which reads as a rich  $\beta$ -rule comprising the consumption of the first argument, the management of the list of extra arguments and, possibly, feeding the continuation with some result (in the latter case in accordance with the  $\beta$ -rule of the  $\Lambda J$ -calculus).

The purpose of this paper is to define, explain and develop the meta-theory of  $\lambda\mathbf{J}^{\mathbf{m}}$ , a rich system worthwhile studying because: (i)  $\lambda\mathbf{J}^{\mathbf{m}}$  comprises permutative conversions, allowing the extension of the studies [Dyckhoff and Pinto 1999; Schwichtenberg 1999] to a setting with cuts; (ii)  $\lambda\mathbf{J}^{\mathbf{m}}$  comprises reduction rules, corresponding to cut-elimination, which, when combined with permutative conversions, offer new strategies for the obtention of  $\lambda$ -terms in  $\beta$ -normal form; (iii)  $\lambda\mathbf{J}^{\mathbf{m}}$  gives a computational interpretation to sequent calculus based on an enriched concept of application, an interpretation that differs from the mainstream interpretations, usually given in terms of pattern matching [Cerrito and Kesner 2004] or explicit substitutions [Sørensen and Urzyczyn 2006]; (iv)  $\lambda\mathbf{J}^{\mathbf{m}}$  has several interesting subsystems, some of which capture internally natural deduction (up to isomorphism).

A summary of the contents of the paper follows.

**Subsystems and natural deduction.** One can isolate, inside our system of generalised multiary application, three classes of terms determined by imposing on the concept of application a trivial form to the feature of multiarity (which means imposing  $l = []$  in (1)) and/or to the feature of generality ( $v = x$  in (1)). These classes are (essentially) closed for reduction, determining the subsystems of *generalised application*  $\lambda\mathbf{J}$  (where multiarity is trivial), of *multiary application*  $\lambda^{\mathbf{m}}$  (where generality is trivial), and of *simple application*  $\lambda$  (where both features are trivial).

The multiary subsystem  $\lambda^{\mathbf{m}}$  corresponds to a system named  $\lambda\mathcal{P}\mathbf{h}$  in [Espírito Santo 2002a; 2002b]. The subsystems  $\lambda\mathbf{J}$  and  $\lambda$ , where multiarity is trivial, are isomorphic copies, inside the sequent calculus world, of systems of natural deduction - the  $\Lambda J$ -calculus and the ordinary  $\lambda$ -calculus (named  $\Lambda$  in this paper), respectively. The diagram in Figure 1 illustrates the situation<sup>4</sup>. The isomorphism  $\mathcal{G}$  (or its slight extension  $\mathcal{G}'$ ) is a mapping that: (i) at the logical level, translates between elimination and appropriate combinations of cut and left introduction (an idea going back to Gentzen); and (ii) at the term language level, makes a mere notational transliteration. Overall,  $\mathcal{G}$  and  $\mathcal{G}'$  go from a system of natural deduction, where one or

<sup>3</sup> $\Lambda J$  is the type-theoretic counterpart to von Plato’s natural deduction system.

<sup>4</sup>The dots symbolise the “frontier” between the sequent calculus and natural deduction worlds.



same name in  $\Lambda J$  - or rather its multiary extension; and the  $\mu$ -rule, corresponding to the rule with the same name in [Schwichtenberg 1999]- or rather its extension to the setting of  $\lambda\mathbf{J}^m$  where cuts are allowed. The  $\beta\pi$ -normal forms are those terms where applications have the form  $x(u, l, (y)v)$ , with function expression necessarily a variable, corresponding to the multiary sequent terms of [Schwichtenberg 1999]. By further imposing  $\mu$ -normality, one recovers what Schwichtenberg calls “multiary normal forms”.

Strong normalisation is proved, as well as confluence, the former for typable terms, the latter for the various combinations of rules. Conservativeness and preservation of strong normalisation over the subsystems also hold, obtained after studying how the mappings between  $\lambda\mathbf{J}^m$  and its subsystems preserve reduction. This subsumes a study of the correspondence between cut-elimination in  $\lambda\mathbf{J}^m$  and normalisation, ordinary or generalised, given that natural deduction may be seen as a subsystem of  $\lambda\mathbf{J}^m$ .

Particularly important is the connection between  $\lambda\mathbf{J}^m$  and its multiarity-free subsystem  $\lambda\mathbf{J}$ , through which  $\lambda\mathbf{J}^m$  benefits from the properties previously proved for  $\Lambda J$  [Joachimski and Matthes 2000; 2003]. But rarely the benefit is won through a routine extension of matters to the multiary setting, because of the presence of rule  $\mu$ , the rule typical of that setting, which poses new problems relatively to what is known of  $\Lambda J$ , but whose properties are part of the solutions.

**Organisation of the paper.** Section 2 introduces the system  $\lambda\mathbf{J}^m$ ; Section 3 studies the subsystems of  $\lambda\mathbf{J}^m$  and the relationship with natural deduction; Section 4 is devoted to the study of the rewriting properties of  $\lambda\mathbf{J}^m$ ; Section 5 concludes the paper. Appendix A gives the proof transformations associated with reduction and permutation rules. Appendix B compares  $\lambda\mathbf{J}^m$  with Herbelin’s  $\bar{\lambda}$ . Appendix C includes proofs of the main results on permutative conversions for rewriting system  $\rightarrow_{pq}$ , and complements these with results for  $\rightarrow_p$  and  $\rightarrow_q$ .

**Previous work on  $\lambda\mathbf{J}^m$ .** This paper is based on [Espírito Santo and Pinto 2003; 2004], where the development of the meta-theory of  $\lambda\mathbf{J}^m$  began. We present here both new results and new proofs of known results. We also state results whose only known proofs: have already been given in [Espírito Santo and Pinto 2003; 2004]; or are just small modifications of results in [Espírito Santo and Pinto 2003]. We provide, in the latter case, the adjusted proofs in appendix.

The presentation and development of  $\lambda\mathbf{J}^m$  is here more detailed in several aspects, particularly in the explanations of  $\lambda\mathbf{J}^m$  as a sequent calculus; of the proof transformations associated to reduction rules and permutative conversions; of the subsystems; of the connections with natural deduction. The new results are: the properties of  $\mu$ ; preservation of reduction by the mappings into the subsystems; conservativeness over subsystems; preservation of strong normalisation of reduction; the properties of the combined systems of reduction and permutations. Although not new, strong normalisation and confluence of reduction receive here new proofs that profit from the above new results; in particular, these new proofs depend only on the properties of the mappings into the subsystems, precisely the mappings that calculate the normal forms w.r.t. permutative conversions (whereas the original [Espírito Santo and Pinto 2004] proofs of strong normalisation and confluence of reduction depended on the properties of other mappings).

## 2. $\lambda\mathbf{J}^m$ : THE GENERALISED MULTIARY $\lambda$ -CALCULUS

In this section we introduce the expressions and typing system of  $\lambda\mathbf{J}^m$ , explaining why the latter is a sequent calculus. Next we introduce and discuss reduction and permutation rules.

### 2.1 Expressions and typing rules

The generalised multiary  $\lambda$ -calculus  $\lambda\mathbf{J}^m$  is a term calculus for intuitionistic implicational logic, corresponding to an extension with cuts of Schwichtenberg's multiary cut-free sequent calculus presented in [Schwichtenberg 1999]. In  $\lambda\mathbf{J}^m$ , *formulas* (=types)  $A, B, C, \dots$  are built up from *propositional variables* using just  $\supset$  (for implication). In the following,  $\mathbf{V}$  denotes the set of variables and  $x, y, w, z$  range over  $\mathbf{V}$ .

*Definition 2.1.* The terms of  $\lambda\mathbf{J}^m$  are described in the following grammar:

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

The sets of  $\lambda\mathbf{J}^m$ -terms and  $\lambda\mathbf{J}^m$ -lists are denoted by  $\mathcal{T}\mathbf{J}^m$  and  $\mathcal{L}\mathbf{J}^m$  respectively. A term of the form  $t(u, l, (x)v)$  is called a *generalised multiary application* (gm-application for short) and  $t$  is called the *head* of such term. The list  $[]$  is called the *empty list* and lists of the form  $t::l$  are called *cons-lists*. In terms  $\lambda x.v$  and  $t(u, l, (x)v)$ , occurrences of  $x$  in  $v$  are bound <sup>5</sup>.

Informally a generalised multiary application  $t(u, l, (x)v)$  can be thought of as the application of a function  $t$  to a list of arguments, whose head is  $u$  and tail is  $l$ , application which is the actual parameter for the explicit substitution for  $x$  in term  $v$ . Multiarity is the capability of applying a function to more than one argument and generality is the capability of specifying the term  $v$  where the result of applying  $t$  to its arguments is going to be used. Such reading of a generalised multiary application agrees with the typing and reduction rules established in the sequel for  $\lambda\mathbf{J}^m$ .

*Contexts*  $\Gamma$  are finite sets of *variable : formula* pairs, associating at most one formula to each variable.  $x \notin \Gamma$  means  $x : A \in \Gamma$  for no  $A$ . The notation  $\Gamma, x : A$  abbreviates  $\Gamma \cup \{x : A\}$ , a set which we always assume to be a context. *Sequents* of  $\lambda\mathbf{J}^m$  are of one of the following two forms  $\Gamma \vdash t : A$ , or  $\Gamma; B \vdash l : C$ , called *term sequents* and *list sequents* respectively. The distinguished position in the LHS of list sequents is called the *stoup*. Read a list sequent  $\Gamma; B \vdash l : C$  as “list  $l$  leads the formula  $B$  to its instance  $C$  in context  $\Gamma$ ”.  $C$  is an *instance* of  $B$  if  $B$  is of the form  $B_1 \supset \dots \supset B_k \supset C$ , for some  $k \geq 0$ .

<sup>5</sup>We adopt for  $\lambda\mathbf{J}^m$  and all the other term calculi defined in the paper the usual variable convention.

*Definition 2.2.* The typing rules of  $\lambda\mathbf{J}^m$  are as follows:

$$\begin{array}{c} \frac{}{x:A, \Gamma \vdash x:A} \textit{Axiom} \qquad \frac{x:A, \Gamma \vdash t:B}{\Gamma \vdash \lambda x.t:A \supset B} \textit{Right} \\ \\ \frac{\Gamma \vdash t:A \supset B \quad \Gamma \vdash u:A \quad \Gamma; B \vdash l:C \quad x:C, \Gamma \vdash v:D}{\Gamma \vdash t(u, l, (x)v):D} \textit{gm-Elim} \\ \\ \frac{\Gamma \vdash u:A \quad \Gamma; B \vdash l:C}{\Gamma; A \supset B \vdash u::l:C} \textit{Lft} \qquad \frac{}{\Gamma; C \vdash []:C} \textit{Ax} \end{array}$$

with the proviso that  $x \notin \Gamma$  in *Right* and in *gm-Elim*. An instance of rule *gm-Elim* is called a *generalised multiary elimination* (or *gm-elimination*, for short).

In a typing derivation, if a formula occurs in the stoup, then this occurrence is “main” and “linear”. This is so for two reasons. Firstly, both *Ax* and *Lft* — the only rules whose conclusion is a list sequent — “introduce” the formula in the stoup of their conclusions. Secondly, in the case of *Ax*, the formula in the stoup is not introduced through weakening; in the case of *Lft*, the formula in the stoup of the conclusion is introduced without contraction.

This typing system may be regarded as an extension of the simple typing system for the  $\lambda$ -calculus, with rules *Axiom* and *Right* for typing variables and  $\lambda$ -abstractions, and an elimination rule *gm-Elim* for typing applications, where the latter depends on auxiliary rules *Ax* and *Lft*. On the other hand, there is a logical view of the system as a sequent calculus. There are two axiom rules *Axiom* and *Ax* (this alternative is available because of the possibility of placing the left axiom formula in the stoup). *Right* is the usual right introduction rule. *Lft* is a constrained left introduction rule, where the main formula  $A \supset B$  is introduced without contraction and the right active formula  $B$  is required to be in the stoup. Finally, there is *gm-Elim*, which we regard as a combination of a form of cut with a form of left introduction more general than *Lft*. Actually, as a logical system,  $\lambda\mathbf{J}^m$  may be defined as an extension, with cuts of a certain form, of Schwichtenberg’s cut-free, multiary, sequent calculus of [Schwichtenberg 1999]. We make this claim more precise in the following subsection.

We end this subsection with two basic properties of the typing system.

**PROPOSITION 2.3** ADMISSIBILITY OF WEAKENING AND STRENGTHENING.

*The following rules are admissible in  $\lambda\mathbf{J}^m$ :*

$$\frac{\Gamma \vdash t:A}{\Gamma, x:B \vdash t:A} W ; \quad \frac{\Gamma, x:B \vdash t:A}{\Gamma \vdash t:A} S \textit{ if } x \notin t .$$

**PROOF.** Each of these is proved by routine simultaneous induction with an analogous property for list sequents.  $\square$

## 2.2 $\lambda\mathbf{J}^m$ as a multiary sequent calculus

**Multiariness.** The sequent calculus of [Schwichtenberg 1999] is *multiary* because it contains, for each  $k \geq 0$ , a rule

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B_1 \quad \dots \quad \Gamma \vdash B_k \quad \Gamma, x:C \vdash D}{\Gamma \vdash D} \textit{Left}_k , \quad (2)$$

with provisos  $y: A \supset B_1 \supset \dots \supset B_k \supset C \in \Gamma$  and  $x \notin \Gamma$ . Instances of this rule are named  $k+1$ -ary *left inferences*. The case  $k = 0$  gives the traditional unary left rule

$$\frac{\Gamma \vdash A \quad \Gamma, x: C \vdash D}{\Gamma \vdash D} \text{Left}_0 ,$$

where  $y: A \supset C \in \Gamma$ .

In a typing system like that of Definition 2.2, multiarity can be implemented by means of a single rule

$$\frac{\Gamma \vdash A \quad \Gamma; B \vdash C \quad \Gamma, x: C \vdash D}{\Gamma \vdash D} \text{m-Left} , \quad (3)$$

where  $y: A \supset B \in \Gamma$ . Instances of this rule are called *multiary left inferences*. This is so because derivability of a sequent  $\Gamma; B \vdash C$  forces either  $B = C$  or the existence of  $k > 0$  and of formulas  $B_1, \dots, B_k$  such that  $B = B_1 \supset \dots \supset B_k \supset C$  and each  $\Gamma \vdash B_i$  is derivable. Now (3) is of course derivable in  $\lambda\mathbf{J}^m$  as

$$\frac{\overline{\Gamma \vdash y: A \supset B} \text{Axiom} \quad \Gamma \vdash A \quad \Gamma; B \vdash C \quad \Gamma, x: C \vdash D}{\Gamma \vdash D} \text{gm-Elim} . \quad (4)$$

It is in this sense that  $\lambda\mathbf{J}^m$  is a multiary system.

**Cuts.** We now explain how we interpret gm-elimination as a specific form of cut. In order to do that, we present some inference rules that are missing as primitive in our system, but that make perfect sense in any system with an auxiliary sort of stoup sequents and expressions. Our gm-eliminations will then be interpreted as a combination of such inferences.

In a calculus featuring sequents with a stoup, a variant of (3) may be defined such that the formula  $A \supset B$  being introduced is placed in the stoup, therefore allowing for no contraction of the introduced formula. A linear variant of *m-Left* is thus obtained, whose instances are called *linear multiary left inferences*:

$$\frac{\Gamma \vdash A \quad \Gamma; B \vdash C \quad \Gamma, x: C \vdash D}{\Gamma; A \supset B \vdash D} \text{lm-Left} . \quad (5)$$

This rule may be seen as a generalisation of *Lft*, for the latter is obtained as the particular case of *lm-Left* where  $C = D$  and the rightmost premiss is an instance of *Axiom*.

The presence of stoup sequents allows also for variant formulations of the cut rule. Whether the cut-formula on the right premise is at the stoup or not distinguishes the following two variants:

$$\frac{\Gamma \vdash A \quad \Gamma, x: A \vdash B}{\Gamma \vdash B} \text{mid-cut} \quad \frac{\Gamma \vdash A \quad \Gamma; A \vdash B}{\Gamma \vdash B} \text{head-cut} .$$

After [Herbelin 1995] we call the former *mid-cut* (*m-cut* for short) and the latter *head-cut* (*h-cut* for short).

We interpret each gm-elimination as a combination of inferences of the following form:

$$\frac{\Gamma \vdash A \supset B \quad \frac{\Gamma \vdash A \quad \Gamma; B \vdash C \quad \Gamma, x: C \vdash D}{\Gamma; A \supset B \vdash D} \text{lm-Left}}{\Gamma \vdash D} \text{head-cut} \quad (6)$$



A gm-elimination is thus seen as a particular form of cut: a head-cut whose cut-formula in the right premiss is main in a linear, multiary left-introduction.

According to (4), gm-eliminations whose leftmost premiss is the conclusion of an axiom represent *m-Left* introductions. So, although we interpret gm-eliminations as cuts (6), gm-eliminations of the form (4) are not to be eliminated. In  $\lambda\mathbf{J}^m$ , at the level of derivations, a *cut* is a gm-elimination whose leftmost premiss is not the conclusion of an axiom. Accordingly, at the level of expressions, we distinguish between gm-applications  $t(u, l, (x)v)$  where the head term  $t$  is a variable, and call them *m-Left introductions*, and those where  $t$  is not a variable, calling them *cuts*. In  $\lambda\mathbf{J}^m$ , cut-elimination is about the elimination of cuts in this sense. A *cut-free* derivation or term is one without occurrences of cut. Therefore cut-free terms are generated by the following grammar.

$$\begin{aligned} t, u, v &::= x \mid \lambda x.t \mid x(u, l, (y)v) \\ l &::= u :: l \mid \square \end{aligned} \quad (7)$$

The cut-free fragment and the cut-free terms of  $\lambda\mathbf{J}^m$  correspond to multiary sequent calculus and multiary sequent terms of [Schwichtenberg 1999]. Bear in mind that  $y(u, l, (x)v)$  is written there as  $v_x\{y, \vec{L}\}$ , where  $\vec{L}$  is the list  $u :: l$ . Notice this  $\vec{L}$  is non-empty as required in [Schwichtenberg 1999]. The  $t(u, l, (x)v)$  notation emphasises the reading of this construction as an application, where  $t$  is the function and  $(u, l, (x)v)$  is the “gm-argument”. The  $v_x\{y, \vec{L}\}$  notation emphasises the explicit substitution reading, where  $v$ ,  $x$  and  $y, \vec{L}$  are respectively the scope, the formal parameter and the actual parameter of the substitution.

**Admissible rules.** In  $\lambda\mathbf{J}^m$  some (but not all) forms of cut are primitive. There is no primitive mid-cut construction - and this is what is missing for having a direct simulation of *LJ* with cuts inside  $\lambda\mathbf{J}^m$ . The mid-cut is admissible and, at the term level, corresponds to a meta-level operation of substitution  $\mathbf{s}(t, x, v)$ , called *generalised multiary substitution (gm-substitution for short)*, and defined as follows:

$$\begin{aligned} \mathbf{s}(t, x, x) &= t \\ \mathbf{s}(t, x, y) &= y, y \neq x \\ \mathbf{s}(t, x, \lambda y.u) &= \lambda y.\mathbf{s}(t, x, u) \\ \mathbf{s}(t, x, u(v, l, (y)v')) &= \mathbf{s}(t, x, u)(\mathbf{s}(t, x, v), \mathbf{s}'(t, x, l), (y)\mathbf{s}(t, x, v')) \\ \mathbf{s}'(t, x, \square) &= \square \\ \mathbf{s}'(t, x, v :: l) &= \mathbf{s}(t, x, v) :: \mathbf{s}'(t, x, l) \end{aligned}$$

**PROPOSITION 2.4 ADMISSIBILITY OF MID-CUT.** *The following rules are admissible in  $\lambda\mathbf{J}^m$ , with the proviso  $x \notin \Gamma$ .*

$$\frac{\Gamma \vdash t : A \quad x : A, \Gamma \vdash v : B}{\Gamma \vdash \mathbf{s}(t, x, v) : B} \quad \frac{\Gamma \vdash t : A \quad x : A, \Gamma; C \vdash l : B}{\Gamma; C \vdash \mathbf{s}'(t, x, l) : B} \quad (8)$$

**PROOF.** By simultaneous induction on  $v$  and  $l$ .  $\square$

The second rule corresponds to the first of the two following auxiliary kinds of cuts existing in the system of [Herbelin 1995]:

$$\frac{\Gamma \vdash A \quad x : A, \Gamma; C \vdash B}{\Gamma; C \vdash B} \text{aux-mid-cut} \quad \frac{\Gamma; C \vdash A \quad \Gamma; A \vdash B}{\Gamma; C \vdash B} \text{aux-head-cut}$$

On the other hand, a particular form of this auxiliary head cut, where the cut formula in the right premiss is main in a *Lft* inference, is admissible in  $\lambda\mathbf{J}^{\mathbf{m}}$ . This requires the *append* operation  $\mathbf{a}(l, u', l')$  defined thus:

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

PROPOSITION 2.5. *The following rule is admissible in  $\lambda\mathbf{J}^{\mathbf{m}}$ :*

$$\frac{\Gamma; C \vdash l : A_1 \supset A_2 \quad \Gamma \vdash u' : A_1 \quad \Gamma; A_2 \vdash l' : B}{\Gamma; C \vdash \mathbf{a}(l, u', l') : B} \quad (9)$$

PROOF. By induction on  $l$ .  $\square$

Think of  $\mathbf{a}(l, u', l')$  as the append of lists  $l$  and  $u' :: l'$ . We interpret (9) as a left-permutable cut of the form

$$\frac{\Gamma; C \vdash A_1 \supset A_2 \quad \frac{\Gamma \vdash A_1 \quad \Gamma; A_2 \vdash B}{\Gamma; A_1 \supset A_2 \vdash B} \text{Lft}}{\Gamma; C \vdash B} \text{aux-head-cut} \quad (10)$$

We end this subsection indicating some basic properties holding of the substitution and append operations and required throughout: (i)  $\mathbf{s}(t, x, u) = u$ , if  $x \notin u$ ; (ii)  $\mathbf{s}(t, x, \mathbf{s}(u, y, v)) = \mathbf{s}(\mathbf{s}(t, x, u), y, \mathbf{s}(t, x, v))$ , if  $y \notin t$ , and  $x \neq y$  (substitution lemma); and (iii)  $\mathbf{a}(\mathbf{a}(l, u, l'), u', l'') = \mathbf{a}(l, u, \mathbf{a}(l', u', l''))$  (append associativity).

### 2.3 Reduction rules

In  $\lambda\mathbf{J}^{\mathbf{m}}$  reduction rules do mostly cut-elimination and are introduced now.

*Definition 2.6.* The *reduction rules* for  $\lambda\mathbf{J}^{\mathbf{m}}$  are as follows<sup>6</sup>:

$$\begin{aligned} (\beta_1) \quad & (\lambda x.t)(u, [], (y)v) \rightarrow \mathbf{s}(\mathbf{s}(u, x, t), y, v) \\ (\beta_2) \quad & (\lambda x.t)(u, v :: l, (y)v') \rightarrow \mathbf{s}(u, x, t)(v, l, (y)v') \\ (\pi) \quad & t(u, l, (x)v)(u', l', (y)v') \rightarrow t(u, l, (x)v(u', l', (y)v')) \\ (\mu) \quad & t(u, l, (x)x(u', l', (y)v)) \rightarrow t(u, \mathbf{a}(l, u', l'), (y)v), \text{ if } x \notin u', l', v \end{aligned}$$

Let  $\beta = \beta_1 \cup \beta_2$ . The notation  $\rightarrow_{\beta\pi\mu}$  stands for the compatible closure of  $\beta \cup \pi \cup \mu$  and the notations  $\rightarrow_{\beta\pi\mu}^=$ ,  $\rightarrow_{\beta\pi\mu}^+$ ,  $\rightarrow_{\beta\pi\mu}^*$  and  $\leftrightarrow_{\beta\pi\mu}^*$  stand for the reflexive, the transitive, the reflexive-transitive and the equivalence closure of  $\rightarrow_{\beta\pi\mu}$  respectively. Notice that the compatible closure of  $\rightarrow_{\beta\pi\mu}$  defines two relations—one on terms and the other on lists—by simultaneous induction. The reflexive, the transitive and the reflexive-transitive closures apply then separately to each of these relations. Normal forms w.r.t.  $\rightarrow_{\beta\pi\mu}$  are called  *$\beta\pi\mu$ -normal forms* or  *$\beta\pi\mu$ -nfs* for short. If

<sup>6</sup>The names  $\pi$  and  $\mu$  come from [Joachimski and Matthes 2003] and [Schwichtenberg 1999], respectively.

$t$  has a unique  $\beta\pi\mu$ -nf, this is denoted  $\downarrow_{\beta\pi\mu}(t)$ . In the sequel we use similar conventions and notations for other reduction relations.

The  $\beta\pi$ -normal forms are exactly the cut-free terms given by grammar (7) above:  $t$  is a  $\beta\pi$ -normal form iff  $t \in (7)$ . The “if” statement follows by induction on  $t \in (7)$ , whereas the “only if” statement (*progress lemma*) follows by induction on  $t$ . A  $\beta\pi\mu$ -normal form  $t$  is a  $\beta\pi$ -normal form such that for each occurrence of  $x(u, l, (y)v)$  in  $t$ , if  $v = y(u', l', (z)v')$  then  $y$  must occur either in  $u'$ ,  $l'$  or  $v'$ . The  $\beta\pi\mu$ -normal forms correspond to Schwichtenberg’s “multiary normal forms”.

Here are some intuitions about the reduction rules. The reduction rules  $\beta_1$ ,  $\beta_2$ ,  $\pi$  aim at making the head of a gm-application a variable. The  $\beta$ -rules cover the case where the head of a gm-application is a lambda-abstraction; they perform function application to arguments. In  $\beta_2$ , since the top argument is consumed, a new argument occupies the top position; in  $\beta_1$  the last argument is consumed and the explicit substitution executed<sup>7</sup>. The  $\pi$ -rule has the effect of permuting gm-applications, simplifying the head of the outer application. Reduction rule  $\mu$  was already considered in [Schwichtenberg 1999]. In a  $\mu$ -redex  $t(u, l, (x)x(u', l', (y)v))$ , because of the proviso  $x \notin u', l', v$  we can anticipate that, after all arguments of  $u :: l$  are consumed and explicit substitution executed, computation will continue with application to a new list  $u' :: l'$  of arguments. The term is simplified by appending the two lists of arguments.

**PROPOSITION 2.7 SUBJECT REDUCTION.** *If  $t \rightarrow_{\beta\pi\mu} t'$  and  $\Gamma \vdash t : A$  is derivable, then so is  $\Gamma \vdash t' : A$ .*

**PROOF.** Proved together with the analogue property for list sequents (if  $l \rightarrow_{\beta\pi\mu} l'$  and  $\Gamma; B \vdash l : A$  is derivable, then so is  $\Gamma; B \vdash l' : A$ ), by simultaneous induction on  $t \rightarrow_{\beta\pi\mu} t'$  and  $l \rightarrow_{\beta\pi\mu} l'$ . The inductive cases are routine. The essence of the proof are the base cases, which associate a proof transformation to each reduction rule, showing how to map a derivation with endsequent  $\Gamma \vdash t : A$  to one with endsequent  $\Gamma \vdash t' : A$  (and similarly for lists). These proof transformations are described in Appendix A.  $\square$

The proof transformations referred to in the proof of subject reduction, and spelled out in the Appendix A, explain the reduction rules as steps of cut-elimination, according to the sequent calculus view of  $\lambda\mathbf{J}^m$  introduced in the previous subsection.

We end this subsection observing that reduction is compatible with the operations of substitution and append in the expected way, namely for  $R \in \{\beta, \pi, \mu\}$ : (i) if  $u \rightarrow_R u'$  then  $\mathbf{s}(u, x, v) \rightarrow_R^* \mathbf{s}(u', x, v)$  and  $\mathbf{s}(v, x, u) \rightarrow_R \mathbf{s}(v, x, u')$ ; (ii) if  $u \rightarrow_R u'$  then  $\mathbf{a}(l, u, l') \rightarrow_R \mathbf{a}(l, u', l')$  and if  $l \rightarrow_R l''$  then  $\mathbf{a}(l, u, l') \rightarrow_R \mathbf{a}(l'', u, l')$  and  $\mathbf{a}(l', u, l) \rightarrow_R \mathbf{a}(l', u, l'')$ .

<sup>7</sup>One might expect that the  $\beta$ -rule of a “multiary” system consumed an entire list of arguments in one single step, as in e.g.

$$(\lambda x_1. \lambda x_2. t)(u_1, u_2 :: u_3 :: l, (y)v) \rightarrow \mathbf{s}(u_2, x_2, \mathbf{s}(u_1, x_1, t))(u_3, l, (y)v) .$$

Let us call this latter kind of  $\beta$ -reduction *simultaneous*. We did not adopt this kind of  $\beta$ -rule because it does not express (as  $\beta_1$  and  $\beta_2$  do) the ordinary transformations in the process of cut-elimination. In addition, simultaneous  $\beta$ -reduction is derivable from  $\beta_1\beta_2$ -reduction.

## 2.4 Permutative conversions

In  $\lambda\mathbf{J}^{\mathbf{m}}$  one has permutative conversions, related to the fact that, in sequent calculus, one may sometimes permute the order of inferences and have essentially the “same” derivation. However, as in [Schwichtenberg 1999], “conversion” will never mean in the present paper a form of equality, but rather an oriented transformation on expressions or derivations, generated by permutative conversion rule(s), which induce rewriting systems. As in [Schwichtenberg 1999] (and also [Dyckhoff and Pinto 1999]), the reason for this approach is that these rewriting systems have good properties and interesting normal forms.

In  $\lambda\mathbf{J}^{\mathbf{m}}$  we have two kinds of permutative conversions (*permutations* for short):  $p$ -permutations and  $q$ -permutations. On the one hand,  $p$ -permutations aim at converting every gm-application to the form  $t(u, l, (x)x)$ , that is, a form that makes no real use of the generality feature. This form characterises the terms of  $\lambda^{\mathbf{m}}$ , a subsystem of  $\lambda\mathbf{J}^{\mathbf{m}}$  to be introduced in the next section. On the other hand,  $q$ -permutations eliminate all occurrences of the construction  $u :: l$  by converting every gm-application to the form  $t(u, [], (x)v)$ , that is, a form that makes no use of the multiarity feature. This form characterises the terms of another subsystem of  $\lambda\mathbf{J}^{\mathbf{m}}$  named  $\lambda\mathbf{J}$ . Only  $pq$ -nfs correspond to  $\lambda$ -terms: see Theorem 4.1.

The  $p$ -permutation rules are:

$$\begin{aligned} (p_1) \quad & t(u, l, (x)y) \rightarrow y, \quad x \neq y \\ (p_2) \quad & t(u, l, (x)\lambda y.v) \rightarrow \lambda y.t(u, l, (x)v) \\ (p_3) \quad & t_1(u_1, l_1, (x)t_2(u_2, l_2, (y)v)) \rightarrow \\ & t_1(u_1, l_1, (x)t_2)(t_1(u_1, l_1, (x)u_2), t_1(u_1, l_1, (x)l_2), (y)v) \quad \text{if } x \notin v, \end{aligned}$$

where:

$$\begin{aligned} t(u, l, (x)v) &= \begin{cases} t(u, l, (x)v) & \text{if } x \in v \\ v & \text{if } x \notin v \end{cases} \\ t(u, l, (x)[]) &= [] \\ t(u, l, (x)u' :: l') &= t(u, l, (x)u') :: t(u, l, (x)l') \end{aligned} \quad (11)$$

The unique  $q$ -permutation rule is

$$(q) \quad t(u, v :: l, (x)v') \rightarrow t(u, [], (y)y)(v, l, (x)v') .$$

Two basic properties of  $p$ -permutations needed in the sequel are:

LEMMA 2.8. *For all  $t, u, v \in \mathcal{T}\mathbf{J}^{\mathbf{m}}$  and  $l \in \mathcal{L}\mathbf{J}^{\mathbf{m}}$ :*

- (1) *if  $x \notin v$  then  $t(u, l, (x)v) \rightarrow_p v$ ;*
- (2)  *$t(u, l, (x)v) \rightarrow_p^- t(u, l, (x)v)$ .*

PROOF. Part 1 follows by case analysis on  $v$  and makes use of all  $p$ -permutations. Part 2 is a consequence of part 1.  $\square$

So, in particular, the transformation of  $t(u, l, (x)v)$  into  $v$ , when  $x \notin v$ , that we call a *garbage collection* step, is performed with one  $p$ -step; moreover  $t(u, l, (x)v)$  and  $t(u, l, (x)v)$  differ at most by a garbage collection step.

In the Appendix A we explain what are the proof transformations associated to permutations and how these transformations relate to traditional conversions for the exchange in the order of inferences in cut-free sequent calculus derivations.

PROPOSITION 2.9 SUBJECT PERMUTATION. *If  $\Gamma \vdash t : A$  and  $t \rightarrow_{pq} t'$  then  $\Gamma \vdash t' : A$ .*

PROOF. Proved together with the analogue property for list sequents (if  $l \rightarrow_{pq} l'$  and  $\Gamma; B \vdash l : A$  is derivable, then so is  $\Gamma; B \vdash l' : A$ ), by simultaneous induction on  $t \rightarrow_{pq} t'$  and  $l \rightarrow_{pq} l'$ . The essence of the proof is the transformation of derivations that can be associated with each permutation rule, as shown in the Appendix A.  $\square$

**Comparison with other systems of permutative conversions.** We considered in [Espírito Santo and Pinto 2003], in addition to the  $q$ -permutation rule of this paper, three slightly different  $p$ -permutation rules, that here we call  $p'_1$ ,  $p'_2$  and  $p'_3$ . The  $p'_1$ -permutation is exactly  $p_1$ , whereas  $p'_2$  and  $p'_3$  read as follows:

$$\begin{aligned} (p'_2) \quad & t(u, l, (x)\lambda y.v) \rightarrow \lambda y.t(u, l, (x)v) \\ (p'_3) \quad & t(u, l, (x)t'(u', l', (z)v)) \rightarrow \\ & t(u, l, (x)t')(t(u, l, (x)u'), t(u, l, (x)l'), (z)v) \text{ if } x \notin v, \end{aligned}$$

where the auxiliary operation  $t(u, l, (x)l')$  is given by

$$\begin{aligned} t(u, l, (x)\square) &= \square \\ t(u, l, (x)u' :: l') &= t(u, l, (x)u') :: t(u, l, (x)l') . \end{aligned}$$

Notice that, when comparing  $p'_2$ ,  $p'_3$  with  $p_2$ ,  $p_3$ , the “new” gm-applications and the calls to  $t(u, l, (x)l')$  generated in the contracta of the former are replaced, in the contracta of the latter, by calls to the operations (11). Thus, given Lemma 2.8, one immediately sees that, for  $i \in \{2, 3\}$ ,  $p_i$  is  $p'_i$  possibly followed by garbage collection steps. As garbage collection can be performed with  $p'$ -permutations<sup>8</sup>, we have  $\rightarrow_p \subseteq \rightarrow_{p'}^+$ . It was (resp. will be) proved that  $\rightarrow_{p'}$  (resp.  $\rightarrow_p$ ) terminates and is confluent. Hence,  $\rightarrow_{p'}$  and  $\rightarrow_p$  calculate the same normal form for each expression of  $\lambda\mathbf{J}^m$ .

Tiny as it may seem, the difference between  $p$ -permutations and  $p'$ -permutations (some garbage collection steps are built in) has dramatic effects when combining permutations and reductions. Indeed, whereas  $\rightarrow_{\beta p'}$  is not strongly normalising, as shown in [Espírito Santo et al. 2006],  $\rightarrow_{\beta p}$  is strongly normalising (Section 4.6). This was the motivation for adopting in this paper the new formulation of  $p$ -permutations.

We consider now the permutations of [Schwichtenberg 1999]. Let us recall them<sup>9</sup>, more precisely rules (1), (2), (3), and (5) of Definition 3.1. in *op. cit.*<sup>10</sup>, which are as follows in our notation:

$$\begin{aligned} w(u, l, (x)v) &\rightarrow v & (1) \\ w(u, l, (x)y(u', l', (z)v)) &\rightarrow y(\mu[w(u, l, (x)u')], \mu[w(u, l, (x)l')], (z)v) & (2) \\ w(u, l, (x)x(u', l', (z)v)) &\rightarrow w(u, \mathbf{a}(l, \mu[w(u, l, (x)u')], \mu[w(u, l, (x)l')]), (z)v) & (3) \\ w(u, l, (x)\lambda y.v) &\rightarrow \lambda y.\mu[w(u, l, (x)v)] & (5) \end{aligned}$$

<sup>8</sup>One proves  $t(u, l, (x)v) \rightarrow_{p'}^+ v$  ( $x \notin v$ ) and  $t(u, l, (x)l') \rightarrow_{p'}^* l'$  ( $x \notin l'$ ) by simultaneous induction on  $v$  and  $l'$ .

<sup>9</sup>Notice that in [Schwichtenberg 1999]  $\pi$  is used for naming permutation relations, e.g.  $\rightarrow_\pi$ .

<sup>10</sup>Rule (4) of [Schwichtenberg 1999] deals with the pairing constructor and is therefore omitted. Also omitted is the case of rule (5) of [Schwichtenberg 1999] where  $\lambda$ -abstraction corresponds to introduction of universal quantification.

The provisos are: (1)  $x \notin v$ ; (2)  $x \neq y$ ,  $x \notin v$ ,  $x \in u'$  or  $x \in l'$ ; (3)  $x \notin v$ ,  $x \in u'$  or  $x \in l'$ . The notation  $\mu[\cdot \cdot \cdot]$  denotes  $\mu$ -normal form.

These rules relate to our  $p$  and  $p'$ -permutation rules. They also have the purpose of trivialising the use of the generality feature (see Lemma 3.5 in [Schwichtenberg 1999] on the characterisation of normal forms). We took from (2), (3) the idea of requiring  $x \notin v$  in rules  $p_3$  and  $p'_3$ , which is crucial in guaranteeing termination. The observation in [Schwichtenberg 1999] that these permutations correspond to steps for explicit execution of substitution also applies to  $p$  and  $p'$ -permutations insofar gm-applications are read as explicit substitutions.

The outstanding difference is that the permutations of [Schwichtenberg 1999] live in the cut-free fragment. For a more precise comparison, we spell out the particular cases of the  $p'$ -permutation rules when the redexes are cut-free:

$$\begin{aligned} (p''_1) \quad & w(u, l, (x)y) \rightarrow y \\ (p''_2) \quad & w(u, l, (x)\lambda y.v) \rightarrow \lambda y.w(u, l, (x)v) \\ (p''_3) \quad & w(u, l, (x)y(u', l', (z)v)) \rightarrow \\ & w(u, l, (x)y)(w(u, l, (x)u'), w(u, l, (x)l'), (z)v) \text{ if } x \notin v \end{aligned}$$

Permutation  $p''_1$  is a particular case of (1), while the latter, in turn, being a garbage collection rule, is derivable from  $p''$ -permutations. Permutation  $p''_2$  differs from (5) only because the contracta of the latter (as all contracta of Schwichtenberg's rules) is forced to be a  $\mu$ -normal form. As to  $p''_3$ , we do not have to require  $x \in u'$  or  $x \in l'$  from the LHS (as is required of (2) and (3) to avoid clash with (1)), since this LHS is never a  $p''_1$ -redex. For the purpose of comparison with (2) and (3), let us admit that the proviso on  $x$  holds. Depending on  $x \neq y$  or  $x = y$ , the LHS is either a (2)- or (3)-redex, respectively. The RHS of  $p''_3$  is a  $\pi$ -redex (hence a cut) and thus requires further treatment in the cut-free setting. If  $x \neq y$ , the RHS of  $p''_3$   $p''_1$ -permutes to

$$y(w(u, l, (x)u'), w(u, l, (x)l'), (z)v) ,$$

which is the (2)-contractum before  $\mu$ -normalisation. If  $x = y$ , we apply to the RHS of  $p''_3$  the auxiliary reduction rule  $h$ , defined in Figure 2, to get

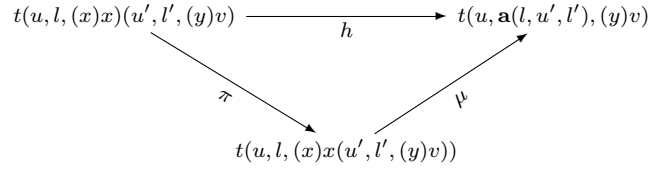
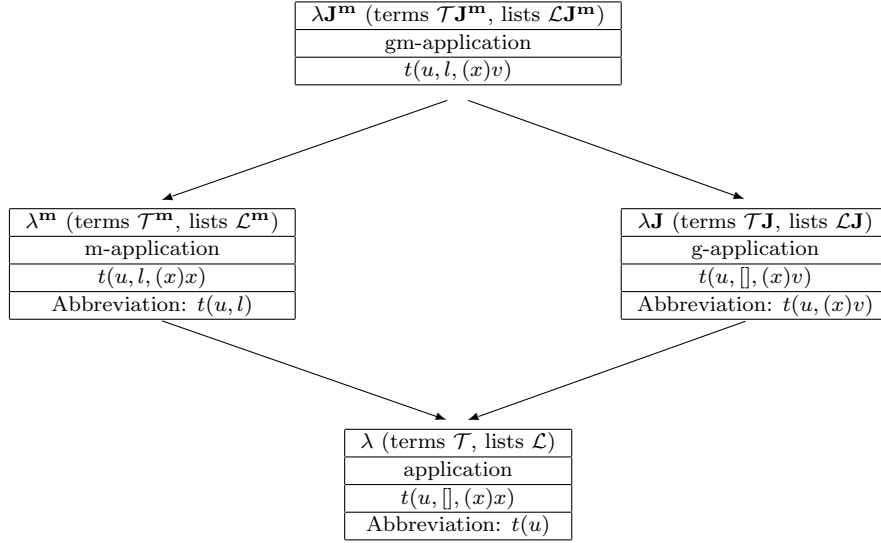
$$w(u, \mathbf{a}(l, w(u, l, (x)u'), w(u, l, (x)l')), (z)v) ,$$

which is the (3)-contractum before  $\mu$ -normalisation. This term may be seen as obtained from the RHS of  $p''_3$  by first contracting this RHS as a  $\pi$ -redex, and immediately contracting the  $\mu$ -redex thus generated, as illustrated in Figure 2<sup>11</sup>.

In [Schwichtenberg 1999] a permutation like  $q$  to deal with multiarity was not necessary because, in the cut-free fragment,  $p$ -permutations are enough in the sense that  $p$ -normal forms are in bijective correspondence with normal terms of the  $\lambda$ -calculus. In  $\lambda\mathbf{J}^{\mathbf{m}}$ , however, many  $p$ -normal forms correspond to the same  $\lambda$ -term<sup>12</sup>

<sup>11</sup>By the way, the triangle shown in this figure is interesting on its own. It shows the different ways available in  $\lambda\mathbf{J}^{\mathbf{m}}$  of expressing the application of  $t$  to  $u$  and to at least one more argument  $u'$ . This is explored in [Espírito Santo et al. 2006].

<sup>12</sup>A permutation similar to  $q$  is first pointed out in [Espírito Santo 2002a] in the context of the system  $\lambda Ph$ , a subsystem of  $\lambda\mathbf{J}^{\mathbf{m}}$  presented in Subsection 3.1 and renamed  $\lambda^{\mathbf{m}}$ .


Fig. 2. Auxiliary reduction rule  $h$ 

Fig. 3. Subsystems of  $\lambda\mathbf{J}^{\mathbf{m}}$  determined by restrictions of gm-application

### 3. SUBSYSTEMS AND NATURAL DEDUCTION

In the previous section, we showed how  $\lambda\mathbf{J}^{\mathbf{m}}$  extends the system in [Schwichtenberg 1999]. In this section we explain the links with three other systems (two of them systems of natural deduction), captured up to isomorphism as the subsystems  $\lambda^{\mathbf{m}}$ ,  $\lambda\mathbf{J}$ , and  $\lambda$  (recall Figure 1). In a final subsection we introduce mappings from  $\lambda\mathbf{J}^{\mathbf{m}}$  to its subsystems. These mappings will play a central role in the development of the meta-theory of  $\lambda\mathbf{J}^{\mathbf{m}}$ .

#### 3.1 Subsystems of $\lambda\mathbf{J}^{\mathbf{m}}$

Several subsystems of  $\lambda\mathbf{J}^{\mathbf{m}}$  are defined by constraining the construction  $t(u, l, (x)v)$ , as illustrated in Figure 3. Each subsystem consists of a subset of the expressions of  $\lambda\mathbf{J}^{\mathbf{m}}$ , over which we define a typing system and reduction and permutation rules.

Actually, we provide two definitions. First, the subsystem is defined officially, through the typing, reduction, and permutation rules of  $\lambda\mathbf{J}^{\mathbf{m}}$ . Second, an alternative, *direct* definition is proposed, where no mention of the rules of  $\lambda\mathbf{J}^{\mathbf{m}}$  is made. In addition, the use of the abbreviations introduced in Figure 3 makes *evident* how the subsystem, when defined in the direct way, is a transliteration of a known system.

The equivalence between the two definitions is given later in Proposition 3.8.

**$\lambda\mathbf{J}$ : the generalised  $\lambda$ -calculus.**

*Definition 3.1.* The terms of  $\lambda\mathbf{J}$  are described in the following grammar:

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

where  $x$  ranges over the set  $\mathbf{V}$  of *variables*.  $\mathcal{T}\mathbf{J}$  and  $\mathcal{L}\mathbf{J}$  are used to denote the sets of  $\lambda\mathbf{J}$ -terms and  $\lambda\mathbf{J}$ -lists respectively.

Hence, the sets  $\mathcal{T}\mathbf{J}$  and  $\mathcal{L}\mathbf{J}$  are the subsets of  $\mathcal{T}\mathbf{J}^{\mathbf{m}}$  and  $\mathcal{L}\mathbf{J}^{\mathbf{m}}$ , respectively, obtained by forbidding the construction  $u :: l$ . A gm-application of the form  $t(u, [], (x)v)$  is called a *generalised application* (or *g-application*, for short).

In the official definition of  $\lambda\mathbf{J}$ , the predicates “ $\Gamma \vdash t : A$  in  $\lambda\mathbf{J}$ ” and “ $t \rightarrow_R t'$  in  $\lambda\mathbf{J}$ ” are defined through  $\lambda\mathbf{J}^{\mathbf{m}}$ . Observe that  $\mathcal{T}\mathbf{J}$  is closed for  $R \in \{\beta_1, \pi, p_1, p_2, p_3\}$ , that is, if  $t \in \mathcal{T}\mathbf{J}$  and  $t \rightarrow_R t'$  in  $\lambda\mathbf{J}^{\mathbf{m}}$ , then  $t' \in \mathcal{T}\mathbf{J}$ . This follows from direct inspection of the rules. Additionally, the case  $R = \beta_1$  requires the fact that  $\mathcal{T}\mathbf{J}$  is closed for gm-substitution. For  $R \in \{\beta_2, \mu, q\}$ , observe that  $\beta_2$ -redexes,  $\mu$ -contracta and  $q$ -redexes fall outside  $\mathcal{T}\mathbf{J}$  (for  $R = \mu$  notice that  $\mathbf{a}([], u', l')$  is a cons-list).

*Definition 3.2.* (i) For  $t \in \mathcal{T}\mathbf{J}$ , we say  $\Gamma \vdash t : A$  in  $\lambda\mathbf{J}$  when  $\Gamma \vdash t : A$  in  $\lambda\mathbf{J}^{\mathbf{m}}$ . (ii) For  $R \in \{\beta_1, \pi, p_1, p_2, p_3\}$  and for  $t, t' \in \mathcal{T}\mathbf{J}$ , we say  $t \rightarrow_R t'$  in  $\lambda\mathbf{J}$  when  $t \rightarrow_R t'$  in  $\lambda\mathbf{J}^{\mathbf{m}}$ .

The alternative definition of the typing predicate and of the reduction and permutation relations of  $\lambda\mathbf{J}$  is as follows. We re-define  $\Gamma \vdash t : A$  and  $t \rightarrow_R t'$  directly, through  $\lambda\mathbf{J}$ 's own rules and not through  $\lambda\mathbf{J}^{\mathbf{m}}$ . We systematically use the abbreviation  $t(u, (x)v)$ , standing for  $t(u, [], (x)v)$ . The  $\lambda\mathbf{J}$ -terms can simply be given by:

$$t, u, v ::= x \mid \lambda x.t \mid t(u, (x)v) \ .$$

The typing rules for  $\lambda\mathbf{J}$  are the obvious *Axiom* and *Right* rules (analogous to those in Definition 2.2), plus the rule:

$$\frac{\Gamma \vdash t : A \supset B \quad \Gamma \vdash u : A \quad x : B, \Gamma \vdash v : C}{\Gamma \vdash t(u, (x)v) : C} \text{ } g\text{-Elim} \ , \quad (12)$$

with proviso  $x : B$  does not belong to  $\Gamma$ . The reduction rules for  $\lambda\mathbf{J}$  are

$$\begin{aligned} (\beta_1) \quad & (\lambda x.t)(u, (y)v) \rightarrow \mathbf{s}(\mathbf{s}(u, x, t), y, v) \\ (\pi) \quad & t(u, (x)v)(u', (y)v') \rightarrow t(u, (x)v(u', (y)v')) \end{aligned}$$

where rule  $\beta_1$  employs *generalised substitution* (*g-substitution* for short), defined as follows:

$$\begin{aligned} \mathbf{s}(t, x, x) &= x \\ \mathbf{s}(t, x, y) &= y, \quad y \neq x \\ \mathbf{s}(t, x, \lambda y.u) &= \lambda y.\mathbf{s}(t, x, u) \\ \mathbf{s}(t, x, u(v, (y)v')) &= \mathbf{s}(t, x, u)(\mathbf{s}(t, x, v), (y)\mathbf{s}(t, x, v')) \end{aligned}$$



We let  $\beta = \beta_1$ . The permutative conversion rules are

$$\begin{aligned} (p_1) \quad & t(u, (x)y) \rightarrow y, \quad x \neq y \\ (p_2) \quad & t(u, (x)\lambda y.v) \rightarrow \lambda y.t(u, (x)v) \\ (p_3) \quad & t_1(u_1, (x)t_2(u_2, (y)v)) \rightarrow t_1(u_1, (x)t_2)(t_1(u_1, (x)u_2), (y)v) \quad \text{if } x \notin v \end{aligned}$$

where the notation  $t(u, (x)v)$  abbreviates  $t(u, [], (x)v)$ . This ends the alternative definition of  $\lambda\mathbf{J}$ .

The system  $\lambda\mathbf{J}$  is isomorphic to the so-called  $\Lambda J$ -calculus of Joachimski and Matthes [Joachimski and Matthes 2003]. The isomorphism is evident, being just a transliteration (to be fully precise, the article just cited does not consider permutative conversion rules). Recall that  $\Lambda J$  is the Curry-Howard counterpart to a system of natural deduction due to von Plato [von Plato 2001], where the idea of generalised elimination rules originated. So, for us,  $\Lambda J$  is a natural deduction system, while  $\lambda\mathbf{J}$  is a sequent calculus fragment. We will return to the relationship with natural deduction in Subsection 3.2.

The set of  $\lambda\mathbf{J}$  normal forms w.r.t. reduction rules (i.e.  $\beta\pi$ -normal forms) is given by the following grammar:

$$t, u ::= x \mid \lambda x.t \mid x(u, (y)t) .$$

Because of the omission of the  $\mu$ -rule, there are  $\lambda\mathbf{J}$  normal forms which are not  $\lambda\mathbf{J}^{\mathbf{m}}$  normal forms.

**$\lambda^{\mathbf{m}}$ : the multiary  $\lambda$ -calculus.**

*Definition 3.3.* The terms of  $\lambda^{\mathbf{m}}$  are described in the following grammar:

$$\begin{aligned} (\lambda^{\mathbf{m}} - \text{terms}) \quad & t, u ::= x \mid \lambda x.t \mid t(u, l, (x)x) \\ (\lambda^{\mathbf{m}} - \text{lists}) \quad & l ::= t :: l \mid [] \end{aligned}$$

where  $x$  ranges over the set  $\mathbf{V}$  of *variables*.  $\mathcal{T}^{\mathbf{m}}$  and  $\mathcal{L}^{\mathbf{m}}$  are used to denote the sets of  $\lambda^{\mathbf{m}}$ -terms and  $\lambda^{\mathbf{m}}$ -lists respectively.

A gm-application of the form  $t(u, l, (x)x)$  is called a *multiary application* (or *m-application*, for short).

In the official definition of  $\lambda^{\mathbf{m}}$ , the predicates “ $\Gamma \vdash t : A$  in  $\lambda^{\mathbf{m}}$ ” and “ $t \rightarrow_R t'$  in  $\lambda^{\mathbf{m}}$ ” are defined through  $\lambda\mathbf{J}^{\mathbf{m}}$ . Observe that the subsets  $\mathcal{T}^{\mathbf{m}}$  and  $\mathcal{L}^{\mathbf{m}}$  are closed for  $R \in \{\beta_1, \beta_2, h, q\}$ <sup>13</sup>. This follows from direct inspection of the rules. Additionally, the cases  $R \in \{\beta_1, \beta_2\}$  require the fact that  $\mathcal{T}^{\mathbf{m}}$  and  $\mathcal{L}^{\mathbf{m}}$  are closed for gm-substitution and the case  $R = h$  requires the fact that  $\mathcal{T}^{\mathbf{m}}$  and  $\mathcal{L}^{\mathbf{m}}$  are closed for the append operator **a**. For  $R \in \{\pi, \mu, p_1, p_2, p_3\}$ , observe that  $\pi$ -contracta,  $\mu$ -redexes and  $p_i$ -redexes fall outside  $\mathcal{T}^{\mathbf{m}}$ . In particular,  $\lambda^{\mathbf{m}}$ -terms are  $\mu$ -nfs. So,  $h$  can be understood as  $\pi$  followed by reduction to  $\mu$ -normal form.

*Definition 3.4.* (i) For  $t \in \mathcal{T}^{\mathbf{m}}$  (resp.  $l \in \mathcal{L}^{\mathbf{m}}$ ), we say  $\Gamma \vdash t : A$  (resp.  $\Gamma ; B \vdash l : C$ ) in  $\lambda^{\mathbf{m}}$  when  $\Gamma \vdash t : A$  (resp.  $\Gamma ; B \vdash l : C$ ) in  $\lambda\mathbf{J}^{\mathbf{m}}$ . (ii) For  $R \in \{\beta_1, \beta_2, h, q\}$  and for  $t, t' \in \mathcal{T}^{\mathbf{m}}$  (resp.  $l, l' \in \mathcal{L}^{\mathbf{m}}$ ), we say  $t \rightarrow_R t'$  (resp.  $l \rightarrow_R l'$ ) in  $\lambda^{\mathbf{m}}$  when  $t \rightarrow_R t'$  (resp.  $l \rightarrow_R l'$ ) in  $\lambda\mathbf{J}^{\mathbf{m}}$ .

<sup>13</sup>Recall the definition of  $h$  in Figure 2.

The alternative definition of  $\lambda^{\mathbf{m}}$  is as follows. We re-define the calculus directly (without using the rules of  $\lambda\mathbf{J}^{\mathbf{m}}$ ) and employ systematically the abbreviation  $t(u, l)$ , standing for the  $m$ -application  $t(u, l, (x)x)$ . The terms and lists of  $\lambda^{\mathbf{m}}$  can simply be given by:

$$t, u, v ::= x \mid \lambda x.t \mid t(u, l) \quad l ::= t::l \mid [] .$$

The typing rules for  $\lambda^{\mathbf{m}}$  are the obvious *Axiom*, *Right*, *Ax* and *Lft* rules (analogous to those in Definition 2.2), plus the rule:

$$\frac{\Gamma \vdash t : A \supset B \quad \Gamma \vdash u : A \quad \Gamma; B \vdash l : C}{\Gamma \vdash t(u, l) : C} \text{ } m\text{-Elim} . \quad (13)$$

The reduction rules for  $\lambda^{\mathbf{m}}$  are

$$\begin{aligned} (\beta_1) \quad & (\lambda x.t)(u, []) \rightarrow \mathbf{s}(u, x, t) \\ (\beta_2) \quad & (\lambda x.t)(u, v::l) \rightarrow \mathbf{s}(u, x, t)(v, l) \\ (h) \quad & t(u, l)(u', l') \rightarrow t(u, \mathbf{a}(l, u', l')) \end{aligned}$$

where rules  $\beta_1$  and  $\beta_2$  employ *multiary substitution* (*m-substitution* for short) defined by

$$\begin{aligned} \mathbf{s}(t, x, x) &= x & \mathbf{s}'(t, x, []) &= [] \\ \mathbf{s}(t, x, y) &= y, \quad y \neq x & \mathbf{s}'(t, x, v::l) &= \mathbf{s}(t, x, v) :: \mathbf{s}'(t, x, l) \\ \mathbf{s}(t, x, \lambda y.u) &= \lambda y.\mathbf{s}(t, x, u) \\ \mathbf{s}(t, x, u(v, l)) &= \mathbf{s}(t, x, u)(\mathbf{s}(t, x, v), \mathbf{s}'(t, x, l)) \end{aligned}$$

and where rule  $h$  employs the append operator  $\mathbf{a}$  restricted to terms in  $\mathcal{T}^{\mathbf{m}}$  and lists in  $\mathcal{L}^{\mathbf{m}}$ . We let  $\beta = \beta_1 \cup \beta_2$ . The unique permutative conversion rule is

$$(q) \quad t(u, v::l) \rightarrow t(u, [])(v, l) .$$

This ends the alternative definition of  $\lambda^{\mathbf{m}}$ .

The system thus obtained is isomorphic to the  $\lambda\mathcal{P}\mathbf{h}$ -calculus defined in [Espírito Santo 2002a; 2002b] (except that in  $\lambda\mathcal{P}\mathbf{h}$  no  $q$ -permutation rule was considered). Originally this system was introduced essentially as the mid-cuts-free fragment of Herbelin's  $\bar{\lambda}$ , but it can also be seen as the multiary extension of the  $\lambda$ -calculus, where functions may be applied to lists of arguments.

The set of  $\lambda^{\mathbf{m}}$  normal forms w.r.t. reduction rules (i.e.  $\beta h$ -normal forms) is the restriction to  $\lambda^{\mathbf{m}}$  of the set of  $\lambda\mathbf{J}^{\mathbf{m}}$  normal forms and can thus be described by:

$$t, u ::= x \mid \lambda x.t \mid x(u, l) \quad l ::= u::l \mid []$$

### $\lambda$ -calculus.

*Definition 3.5.* The terms of  $\lambda$  are described in the following grammar:

$$\begin{aligned} (\lambda\text{-terms}) \quad & t, u ::= x \mid \lambda x.t \mid t(u, l, (x)x) \\ (\lambda\text{-lists}) \quad & l ::= [] \end{aligned}$$

where  $x$  ranges over the set  $\mathbf{V}$  of *variables*.  $\mathcal{T}$  and  $\mathcal{L}$  are used to denote the sets of  $\lambda$ -terms and  $\lambda$ -lists respectively.

Hence, the sets  $\mathcal{T}$  and  $\mathcal{L}$  are the subsets of  $\mathcal{TJ}^{\mathbf{m}}$  and  $\mathcal{LJ}^{\mathbf{m}}$  obtained by combining the constraints that define  $\lambda\mathbf{J}$  and  $\lambda^{\mathbf{m}}$ . We call (*simple*) *application* a gm-application of the form  $t(u, [], (x)x)$ .

In the official definition of  $\lambda$ , the predicates “ $\Gamma \vdash t : A$  in  $\lambda$ ” and “ $t \rightarrow_R t'$  in  $\lambda$ ” are defined through  $\lambda\mathbf{J}^{\mathbf{m}}$ . Observe that the subset of terms  $\mathcal{T}$  is closed for  $\beta_1$  because  $\mathcal{T}$  is closed for gm-substitution. As in  $\lambda\mathbf{J}$ , rules  $\beta_2$ ,  $\mu$  and  $q$  are dropped. As in  $\lambda^{\mathbf{m}}$ , rules  $\pi$  and  $p_i$  are also dropped, but, contrary to the latter system, we cannot replace rule  $\pi$  by rule  $h$  because  $\mu$ -*contracta* (hence  $h$ -*contracta*) fall outside  $\mathcal{T}$ .

*Definition 3.6.* (i) For  $t \in \mathcal{T}$ , we say  $\Gamma \vdash t : A$  in  $\lambda$  when  $\Gamma \vdash t : A$  in  $\lambda\mathbf{J}^{\mathbf{m}}$ . (ii) For  $t, t' \in \mathcal{T}$ , we say  $t \rightarrow_{\beta_1} t'$  in  $\lambda$  when  $t \rightarrow_{\beta_1} t'$  in  $\lambda\mathbf{J}^{\mathbf{m}}$ .

The alternative definition of  $\lambda$  re-defines the typing predicate and  $\beta_1$ -reduction directly, through  $\lambda$ 's own rules and not through  $\lambda\mathbf{J}^{\mathbf{m}}$ . The abbreviation  $t(u)$ , standing for  $t(u, [], (x)x)$ , is employed systematically.  $\lambda$ -terms can simply be described as:

$$t, u ::= x \mid \lambda x.t \mid t(u) .$$

The typing rules for  $\lambda$  are the obvious *Axiom* and *Right* rules (analogous to those in Definition 2.2), plus the rule:

$$\frac{\Gamma \vdash t : A \supset B \quad \Gamma \vdash u : A}{\Gamma \vdash t(u) : B} \textit{Elim} . \quad (14)$$

The unique reduction rule for  $\lambda$  is

$$(\beta_1) (\lambda x.t)(u) \rightarrow \mathbf{s}(u, x, t) ,$$

also called  $\beta$ . It uses (*simple*) *substitution*, defined as follows

$$\begin{aligned} \mathbf{s}(t, x, x) &= x \\ \mathbf{s}(t, x, y) &= y, y \neq x \\ \mathbf{s}(t, x, \lambda y.u) &= \lambda y.\mathbf{s}(t, x, u) \\ \mathbf{s}(t, x, u(v)) &= \mathbf{s}(t, x, u)(\mathbf{s}(t, x, v)) \end{aligned}$$

The terms of  $\lambda$  are *pq*-normal forms. Hence, there are no permutative conversions in  $\lambda$ . This completes the alternative definition of  $\lambda$ .

Again, the normal forms of  $\lambda$  (i.e.  $\beta_1$ -normal forms) are not necessarily normal forms of its extensions. For example,  $t(u)(u')$  is not a redex of  $\lambda$ , but it is a redex of its extensions.

The system  $\lambda$  is no more than an isomorphic copy of the ordinary  $\lambda$ -calculus. Again, this fact is an evidence, and has a meaning in terms of the relationship between natural deduction and sequent calculus, see Subsection 3.2.

**Equivalence of the alternative definitions of the subsystems.** We need some preliminary remarks. Each subsystem carries its own substitution operator. In the following, we use subscripts to stress that gm- or g- or m-substitution is intended (no subscript for substitution in  $\lambda$ ). Similarly for the append operator.

**LEMMA 3.7.** (1) For all  $t, v$  in  $\mathcal{TJ}$ ,  $\mathbf{s}_{gm}(t, x, v) = \mathbf{s}_g(t, x, v)$ . (2) For all  $t, u, v$  in  $\mathcal{T}^{\mathbf{m}}$ , all  $l, l'$  in  $\mathcal{L}^{\mathbf{m}}$ ,  $\mathbf{s}_{gm}(t, x, v) = \mathbf{s}_m(t, x, v)$  and  $\mathbf{a}_{gm}(l, u, l') = \mathbf{a}_m(l, u, l')$ . (3) For all  $t, v$  in  $\mathcal{T}$ ,  $\mathbf{s}_{gm}(t, x, v) = \mathbf{s}(t, x, v)$ .

PROOF. Routine.  $\square$

In the following proposition, the LHS of each of the six equivalences actually means  $\Gamma \vdash t : A$  or  $t \rightarrow_R t'$  according to the official Definitions 3.2, 3.4, and 3.6. So, in this proposition one establishes the equivalence between the official definitions of the subsystems and the corresponding direct ones.

PROPOSITION 3.8 SUBSYSTEMS OF  $\lambda\mathbf{J}^m$ .

- (1) (i) For all  $t$  in  $\mathcal{T}\mathbf{J}$ ,  $\Gamma \vdash t : A$  in  $\lambda\mathbf{J}^m$  iff  $\Gamma \vdash t : A$  in  $\lambda\mathbf{J}$  by the direct definition.  
(ii) Let  $R \in \{\beta_1, \pi, p_1, p_2, p_3\}$ . For all  $t, t'$  in  $\mathcal{T}\mathbf{J}$ ,  $t \rightarrow_R t'$  in  $\lambda\mathbf{J}^m$  iff  $t \rightarrow_R t'$  in  $\lambda\mathbf{J}$  by the direct definition.
- (2) (i) For all  $t$  in  $\mathcal{T}^m$ ,  $\Gamma \vdash t : A$  in  $\lambda\mathbf{J}^m$  iff  $\Gamma \vdash t : A$  in  $\lambda^m$  by the direct definition.  
(ii) Let  $R \in \{\beta_1, \beta_2, h, q\}$ . For all  $t, t'$  in  $\mathcal{T}^m$ ,  $t \rightarrow_R t'$  in  $\lambda\mathbf{J}^m$  iff  $t \rightarrow_R t'$  in  $\lambda^m$  by the direct definition.
- (3) (i) For all  $t$  in  $\mathcal{T}$ ,  $\Gamma \vdash t : A$  in  $\lambda\mathbf{J}^m$  iff  $\Gamma \vdash t : A$  in  $\lambda$  by the direct definition. (ii) Let  $R = \beta_1$ . For all  $t, t'$  in  $\mathcal{T}$ ,  $t \rightarrow_R t'$  in  $\lambda\mathbf{J}^m$  iff  $t \rightarrow_R t'$  in  $\lambda$  by the direct definition.

PROOF. Part (i) of statement 1 is proved by induction on  $t \in \lambda\mathbf{J}$ . The key observation is that a term  $t(u, (x)v)$  is typable in  $\lambda\mathbf{J}^m$  only with the particular case of the rule *gm-Elim* where the penultimate premiss is an instance of *Ax*, and such particular case corresponds exactly to the rule *g-Elim* (12).

Part (ii) of statement 1 is proved by two inductions. First, by induction on  $t \rightarrow_R t'$  in  $\lambda\mathbf{J}^m$ , one proves that if  $t$  is in  $\mathcal{T}\mathbf{J}$ , then  $t \rightarrow_R t'$  in  $\lambda\mathbf{J}$  by the direct definition. (Notice that  $t' \in \mathcal{T}\mathbf{J}$ , for  $\mathcal{T}\mathbf{J}$  is closed for  $\rightarrow_R$  of  $\lambda\mathbf{J}^m$ .) Second, by induction on  $t \rightarrow_R t'$  in  $\lambda\mathbf{J}$  by the direct definition, one proves that  $t \rightarrow_R t'$  in  $\lambda\mathbf{J}^m$ . In both proofs the inductive cases are routine. The base cases follow from the commutativity of the next two diagrams (where  $t, u, v$  are in  $\mathcal{T}\mathbf{J}$ ):

$$\begin{array}{ccc}
(\lambda x.t)(u, [], (y)v) \rightarrow_{\beta_1} \mathbf{s}_{gm}(\mathbf{s}_{gm}(u, x, t), y, v) & & \\
\parallel & & \parallel \\
(\lambda x.t)(u, (y)v) \rightarrow_{\beta_1} \mathbf{s}_g(\mathbf{s}_g(u, x, t), y, v) & & \\
\\
t(u, [], (x)v)(u', [], (y)v') \rightarrow_{\pi} t(u, [], (x)v(u', [], (y)v')) & & \\
\parallel & & \parallel \\
t(u, (x)v)(u', (y)v') \rightarrow_{\pi} t(u, (x)v(u', (y)v')) & & 
\end{array}$$

The first of the above diagrams uses part 1 of Lemma 3.7. The second just expands abbreviations. The same happens in the cases  $p_i$ , so these are omitted.

Statements 2 and 3 are proved in a similar way. Part (i) of statement 2 relies on the key observation that a term  $t(u, l)$  is typable in  $\lambda\mathbf{J}^m$  only with the particular case of the rule *gm-Elim* where the rightmost premiss is an instance of *Axiom*, and such particular case corresponds exactly to the rule *m-Elim* (13). Part (i) of statement 3 relies on the key observation that a term  $t(u)$  is typable in  $\lambda\mathbf{J}^m$  only with the particular case of the rule *gm-Elim* where the last two premisses are instances of *Ax* and *Axiom* respectively, and such particular case corresponds

exactly to the rule *Elim* (14). As to part (ii) of statements 2 and 3: 2 needs two simultaneous inductions, for proving similar claims for lists, and statement 2 (resp. 3) uses part 2 (resp. part 3) of Lemma 3.7.  $\square$

Parts (i) of Proposition 3.8 are conservativeness results for the typing systems of  $\lambda\mathbf{J}$ ,  $\lambda^{\mathbf{m}}$  and  $\lambda$ , given by the direct definitions (according to parts (i) of Definitions 3.2, 3.4, and 3.6, conservativeness holds by definition).

The same applies to parts (ii) of Definitions 3.2, 3.4, and 3.6 and Proposition 3.8, that concern reduction and permutation rules. However, parts (ii) of Proposition 3.8 only give conservativeness in a weak sense. For instance,  $\beta$ -reduction in  $\lambda$  is equivalent to  $\beta$ -reduction (not full reduction) in  $\lambda\mathbf{J}^{\mathbf{m}}$ . What we do in the next proposition is to establish this form of conservativeness for  $\lambda^{\mathbf{m}}$  and  $\lambda\mathbf{J}$  over  $\lambda$ . In all cases, full conservativeness will be proved later (Theorems 4.16 and 4.17 in Subsection 4.4).

It is easy to verify that the set of  $\lambda$ -terms is closed for  $\beta$ -reduction both in  $\lambda\mathbf{J}$  and  $\lambda^{\mathbf{m}}$ .

**PROPOSITION 3.9.** *For  $t, t' \in \mathcal{T}$ , the following is equivalent: (i)  $t \rightarrow_{\beta_1} t'$  in  $\lambda$ ; (ii)  $t \rightarrow_{\beta_1} t'$  in  $\lambda\mathbf{J}$ ; (iii)  $t \rightarrow_{\beta_1} t'$  in  $\lambda^{\mathbf{m}}$ .*

**PROOF.** Take for (i) the direct definition. Then, the proofs of (ii) iff (i) and (iii) iff (i) are similar to the proof of 3 in Proposition 3.8.  $\square$

### 3.2 $\lambda\mathbf{J}^{\mathbf{m}}$ and natural deduction

We observed that there are evident isomorphisms between  $\Lambda$  (resp.  $\Lambda\mathbf{J}$ ) and the subsystem  $\lambda$  (resp.  $\lambda\mathbf{J}$ ) of  $\lambda\mathbf{J}^{\mathbf{m}}$  (recall that we denote the ordinary  $\lambda$ -calculus by  $\Lambda$ ). We now analyse the meaning of this fact in terms of the relationship between the sequent calculus  $\lambda\mathbf{J}^{\mathbf{m}}$  and natural deduction.

The set of  $\Lambda$ -terms is given by:

$$M, N ::= x \mid \lambda x.M \mid MN$$

Then, the isomorphism  $\mathcal{G} : \Lambda \rightarrow \lambda$  is, to start with, a bijection between the set of  $\Lambda$ -terms and the set of  $\lambda$ -terms (i.e. the subset of  $\lambda\mathbf{J}^{\mathbf{m}}$ -terms where every gm-application has the form  $t(u, [], (x)x)$ ). If we let  $t = \mathcal{G}(M)$  and  $u = \mathcal{G}(N)$ , this bijection is defined by  $\mathcal{G}(x) = x$ ,  $\mathcal{G}(\lambda x.M) = \lambda x.t$  and  $\mathcal{G}(MN) = t(u)$ .  $\mathcal{G}$  is a mere transliteration that extends to a transliteration between ordinary  $\beta$ -reduction in  $\Lambda$  and  $\beta_1$ -reduction in  $\lambda$ .  $\mathcal{G}$  also extends to derivations and its effect on derivations is the replacement of each occurrence of the elimination rule

$$\frac{\Gamma \vdash M : A \supset B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B} \quad (15)$$

(that belongs to the natural deduction system associated with the  $\Lambda$ -calculus) by an occurrence of rule *Elim* (14).

Now, unfolding the abbreviation *Elim* (14) into a *gm-Elim*, as explained in the proof of Proposition 3.8, and applying our sequent calculus view (6) of the rule *gm-Elim*, we see that each occurrence of *Elim* corresponds to the following combination

of inferences

$$\frac{\Gamma \vdash A \supset B \quad \frac{\Gamma \vdash A \quad \overline{\Gamma; B \vdash B} \text{ Axiom}}{\Gamma; A \supset B \vdash B} \text{ lm-Left}}{\Gamma \vdash B} \text{ h-cut}$$

Since the middle premiss of the *lm-Left*-inference in this figure is the conclusion of an axiom, the mentioned *lm-Left*-inference is a unary left introduction, whose main formula is linear. We may simplify this figure thus:

$$\frac{\Gamma \vdash A \supset B \quad \frac{\Gamma \vdash A \quad \overline{\Gamma, x : B \vdash B} \text{ Axiom}}{\Gamma; A \supset B \vdash B} \text{ l-Left}}{\Gamma \vdash B} \text{ h-cut} \quad (16)$$

This is a familiar combination of inferences. Indeed, when  $\mathcal{G}$  is regarded as the replacement of (15) by (16), it becomes a variant of the translation of natural deduction to sequent calculus that goes back to Gentzen [1969]. This is why we refer to  $\mathcal{G}$  as Gentzen's mapping. One should stress, as was done in [Espírito Santo 2002a] in a different context, that, by establishing an isomorphism between  $\beta$ -reduction in  $\Lambda$  and  $\beta_1$ -reduction in  $\lambda$ ,  $\mathcal{G}$  becomes an isomorphism between normalisation for minimal logic and the elimination of cuts of the form (16) in the sequent calculus  $\lambda\mathbf{J}^m$ .

The analysis we have made of the relationship between the  $\Lambda$ -calculus and the subsystem  $\lambda$  can be adapted to the relationship between the  $\Lambda J$ -calculus and subsystem  $\lambda\mathbf{J}$ . The set of  $\Lambda J$ -terms is given by:

$$M, N, P ::= x \mid \lambda x.M \mid M(N, x.P)$$

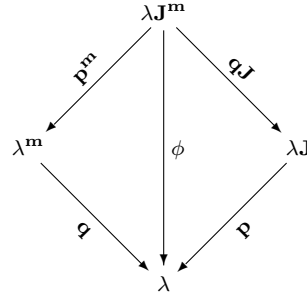
The isomorphism  $\mathcal{G}' : \Lambda J \rightarrow \lambda\mathbf{J}$  is, to start with, a bijection between the set of  $\Lambda J$ -terms and the set of  $\lambda\mathbf{J}$ -terms (i.e. the subset of  $\lambda\mathbf{J}^m$ -terms where every gm-application has the form  $t(u, [], (x)v)$ ). The bijection is defined by:  $\mathcal{G}'(x) = x$ ,  $\mathcal{G}'(\lambda x.M) = \lambda x.t$  and  $\mathcal{G}'(M(N, x.P)) = t(u, (x)v)$ , where  $t = \mathcal{G}'(M)$ ,  $u = \mathcal{G}'(N)$  and  $v = \mathcal{G}'(P)$ . If we define  $MN = M(N, x.x)$ ,  $\mathcal{G}'$  agrees with  $\mathcal{G}$  on  $\Lambda$ -terms.  $\mathcal{G}'$  is a mere transliteration, that extends to a transliteration between  $\beta\pi$ -reduction in  $\Lambda J$  and  $\beta_1\pi$ -reduction in  $\lambda\mathbf{J}$ .  $\mathcal{G}'$  also extends to derivations, and consists of the replacement of each occurrence of the general elimination rule

$$\frac{\Gamma \vdash M : A \supset B \quad \Gamma \vdash N : A \quad \Gamma, x : B \vdash P : C}{\Gamma \vdash M(N, x.P) : C} \quad (17)$$

(that belongs to the natural deduction system associated with  $\Lambda J$ ) by an occurrence of rule *g-Elim* (12). Unfolding the abbreviation *g-Elim* (12) into a *gm-Elim*, as explained in the proof of Proposition 3.8, and applying our sequent calculus view of the latter rule, *g-Elim* becomes, in turn, an abbreviation of the combination of inferences

$$\frac{\Gamma \vdash A \supset B \quad \frac{\Gamma \vdash A \quad \Gamma, x : B \vdash C}{\Gamma; A \supset B \vdash C} \text{ l-Left}}{\Gamma \vdash C} \text{ h-cut} \quad (18)$$

When regarded as the replacement of (17) by (18),  $\mathcal{G}'$  is an extension of Gentzen's idea of translating (15) by (16). In addition, by establishing an isomorphism be-


Fig. 4.  $\lambda\mathbf{J}^m$  and its subsystems

tween  $\beta\pi$ -reduction in  $\Lambda J$  and  $\beta_1\pi$ -reduction in  $\lambda\mathbf{J}$ ,  $\mathcal{G}'$  becomes an isomorphism between normalisation for minimal logic in the natural deduction system of [von Plato 2001]<sup>14</sup> and the elimination of cuts of the form (18) in the sequent calculus  $\lambda\mathbf{J}^m$ .

### 3.3 Mappings into subsystems

The mappings from  $\lambda\mathbf{J}^m$  into its subsystems are summarised in Figure 4<sup>15</sup>. They show how to translate to systems where one or both of the features of multiarity and generality are missing. As described in Theorem 4.1 of next section and in the analogue Theorem C.14 of Appendix C, the mappings from  $\lambda\mathbf{J}^m$  into its subsystems are also the normal-form mappings relative to the terminating and confluent rewriting systems of  $p$ ,  $q$ , and  $pq$ -permutations.

Mapping  $\phi$  gives a direct interpretation of  $\lambda\mathbf{J}^m$  into  $\lambda$  and is defined in Figure 5. It translates a gm-application  $u_0(u_1, [u_2, u_3], (x)u_4)$ , say, by  $\mathbf{s}(u'_0(u'_1)(u'_2)(u'_3), x, u'_4)$  where  $u'_i$  denotes the recursive translation of  $u_i$ . Mappings from  $\lambda\mathbf{J}^m$  into the subsystems  $\lambda^m$  and  $\lambda\mathbf{J}$  and from these into  $\lambda$  are also defined in Figure 5. A mapping corresponding to  $\mathbf{p}$  from  $\lambda\mathbf{J}$  into the  $\lambda$ -calculus is given in [Joachimski and Matthes 2003]. In [Espírito Santo 2002a] an interpretation  $\mathcal{Q}$  of  $\lambda^m$  (or rather  $\lambda\mathcal{P}\mathbf{h}$ ) into the  $\lambda$ -calculus corresponding to  $\mathbf{q}$  is studied. A simple inspection of the definitions shows that  $\mathbf{p}$  and  $\mathbf{q}$  are the restrictions of  $\mathbf{p}^m$  and  $\mathbf{q}\mathbf{J}$  to  $\mathcal{T}\mathbf{J}$  and  $\mathcal{T}^m$ . We note also that  $\mathbf{p}^m$  and  $\mathbf{q}\mathbf{J}$  restricted to  $\mathcal{T}^m$  and  $\mathcal{T}\mathbf{J}$ -terms are identity mappings and, in virtue of the next proposition,  $\mathbf{p}$  and  $\mathbf{q}$  are also the restrictions of  $\phi$  to these two classes of terms.

PROPOSITION 3.10  $\phi$  DECOMPOSITION.  $\mathbf{p} \circ \mathbf{q}\mathbf{J} = \phi = \mathbf{q} \circ \mathbf{p}^m$ .

PROOF. These properties are proved together with the following properties (a) and (b), respectively, using simultaneous induction.

- (a)  $\phi'(\phi(t), \phi(u), l, x, \phi(v)) = \mathbf{p}(\mathbf{q}\mathbf{J}'(\mathbf{q}(t), \mathbf{q}(u), l, x, \mathbf{q}(v)))$
- (b)  $\phi'(\phi(t), \phi(u), l, x, \phi(v)) = \mathbf{q}(\mathbf{s}(\mathbf{p}^m(t)(\mathbf{p}^m(u), \mathbf{p}^m(l)), x, \mathbf{p}^m(v)))$

<sup>14</sup>This includes the “hidden convertibility” that comes with the general elimination rule for implication, corresponding to the  $\pi$ -reduction rule of  $\Lambda J$ :  $M'(N', y.P')(N, x.P) \rightarrow M'(N', y.P'(N, x.P))$ .

<sup>15</sup>Regrettably, the same diagram in [Espírito Santo and Pinto 2003] mixed up  $\mathbf{p}$  and  $\mathbf{q}$  in the names of the mappings.

$$\begin{array}{ll}
\phi : \mathcal{T}\mathbf{J}^{\mathbf{m}} \longrightarrow \mathcal{T} & \phi' : \mathcal{T} \times \mathcal{T} \times \mathcal{L}\mathbf{J}^{\mathbf{m}} \times \mathbf{V} \times \mathcal{T} \longrightarrow \mathcal{T} \\
\phi(x) = x & \phi'(t, u, [], x, v) = \mathbf{s}(t(u), x, v) \\
\phi(\lambda x.t) = \lambda x.\phi(t) & \phi'(t, u, v :: l, x, v') = \phi'(t(u), \phi(v), l, x, v') \\
\phi(t(u, l, (x)v)) = \phi'(\phi(t), \phi(u), l, x, \phi(v)) & \\
\\
\mathbf{p}^{\mathbf{m}} : \mathcal{T}\mathbf{J}^{\mathbf{m}} \longrightarrow \mathcal{T}^{\mathbf{m}} & \mathbf{p}^{\mathbf{m}'} : \mathcal{L}\mathbf{J}^{\mathbf{m}} \longrightarrow \mathcal{L}^{\mathbf{m}} \\
\mathbf{p}^{\mathbf{m}}(x) = x & \mathbf{p}^{\mathbf{m}'}([]) = [] \\
\mathbf{p}^{\mathbf{m}}(\lambda x.t) = \lambda x.\mathbf{p}^{\mathbf{m}}(t) & \mathbf{p}^{\mathbf{m}'}(u :: l) = \mathbf{p}^{\mathbf{m}}(u) :: \mathbf{p}^{\mathbf{m}'}(l) \\
\mathbf{p}^{\mathbf{m}}(t(u, l, (x)v)) = \mathbf{s}(\mathbf{p}^{\mathbf{m}}(t)(\mathbf{p}^{\mathbf{m}}(u), \mathbf{p}^{\mathbf{m}'}(l)), x, \mathbf{p}^{\mathbf{m}}(v)) & \\
\\
\mathbf{q}\mathbf{J} : \mathcal{T}\mathbf{J}^{\mathbf{m}} \longrightarrow \mathcal{T}\mathbf{J} & \mathbf{q}\mathbf{J}' : \mathcal{T}\mathbf{J} \times \mathcal{T}\mathbf{J} \times \mathcal{L}\mathbf{J}^{\mathbf{m}} \times \mathbf{V} \times \mathcal{T}\mathbf{J} \longrightarrow \mathcal{T}\mathbf{J} \\
\mathbf{q}\mathbf{J}(x) = x & \mathbf{q}\mathbf{J}'(t, u, v :: l, x, v') = \mathbf{q}\mathbf{J}'(t(u), \mathbf{q}\mathbf{J}(v), l, x, v') \\
\mathbf{q}\mathbf{J}(\lambda x.t) = \lambda x.\mathbf{q}\mathbf{J}(t) & \mathbf{q}\mathbf{J}'(t, u, [], x, v) = t(u, (x)v) \\
\mathbf{q}\mathbf{J}(t(u, l, (x)v)) = \mathbf{q}\mathbf{J}'(\mathbf{q}\mathbf{J}(t), \mathbf{q}\mathbf{J}(u), l, x, \mathbf{q}\mathbf{J}(v)) & \\
\\
\mathbf{p} : \mathcal{T}\mathbf{J} \longrightarrow \mathcal{T} & \\
\mathbf{p}(x) = x & \\
\mathbf{p}(\lambda x.t) = \lambda x.\mathbf{p}(t) & \\
\mathbf{p}(t(u, (x)v)) = \mathbf{s}(\mathbf{p}(t)(\mathbf{p}(u)), x, \mathbf{p}(v)) & \\
\\
\mathbf{q} : \mathcal{T}^{\mathbf{m}} \longrightarrow \mathcal{T} & \mathbf{q}' : \mathcal{T} \times \mathcal{T} \times \mathcal{L}^{\mathbf{m}} \longrightarrow \mathcal{T} \\
\mathbf{q}(x) = x & \mathbf{q}'(t, u, v :: l) = \mathbf{q}'(t(u), \mathbf{q}(v), l) \\
\mathbf{q}(\lambda x.t) = \lambda x.\mathbf{q}(t) & \mathbf{q}'(t, u, []) = t(u) \\
\mathbf{q}(t(u, l)) = \mathbf{q}'(\mathbf{q}(t), \mathbf{q}(u), l) &
\end{array}$$

Fig. 5. The mappings  $\phi$ ,  $\mathbf{p}^{\mathbf{m}}$ ,  $\mathbf{q}\mathbf{J}$ ,  $\mathbf{p}$  and  $\mathbf{q}$ 

□

All the mappings considered above preserve typability in the expected way.

**PROPOSITION 3.11 PRESERVATION OF TYPABILITY.** *Let  $F$  be  $\phi$  (resp.  $\mathbf{q}\mathbf{J}$ ,  $\mathbf{p}^{\mathbf{m}}$ ,  $\mathbf{q}$  and  $\mathbf{p}$ ), let  $S$  be  $\lambda\mathbf{J}^{\mathbf{m}}$  (resp.  $\lambda\mathbf{J}^{\mathbf{m}}$ ,  $\lambda\mathbf{J}^{\mathbf{m}}$ ,  $\lambda^{\mathbf{m}}$  and  $\lambda\mathbf{J}$ ) and let  $S'$  be  $\lambda$  (resp.  $\lambda\mathbf{J}$ ,  $\lambda^{\mathbf{m}}$ ,  $\lambda$  and  $\lambda$ ). If  $\Gamma \vdash t : A$  in  $S$  then  $\Gamma \vdash F(t) : A$  in  $S'$ .*

**PROOF.** In hold of part 4 of Theorem 4.1 (next section) and part 3 of Theorem C.14 (Appendix C), this result is a consequence of subject permutation (Proposition 2.9). □

Other properties of these mappings are detailed as needed in Subsection 4.1.

#### 4. REWRITING PROPERTIES OF $\lambda\mathbf{J}^{\mathbf{m}}$

Here we establish the main rewriting properties of  $\lambda\mathbf{J}^{\mathbf{m}}$ . We start by stating the main results about permutative conversions, which are as in [Espírito Santo and Pinto 2003], but recall  $p$ -permutations are now slightly different. The adjusted proofs of these results are presented in Appendix C. Then we establish new properties of reduction (namely: properties of  $\mu$ -reduction, preservation of reduction by the various mappings of Section 3, conservativeness over the subsystems, and preservation of strong normalisation) and based on them we give new proofs of



strong normalisation and confluence of reduction. Last we address the interplay between reduction and permutative conversions. In particular, for combinations of rules whose normal forms are the  $\beta$ -normal forms of  $\lambda$ , we prove confluence and strong normalisation.

#### 4.1 Properties of permutative conversions

From the viewpoint of  $\lambda$ -calculi, the goal of permutations is to reduce gm-applications to simple applications. Taking this set of conversions as defining a rewriting system, one obtains a strongly normalising and confluent rewriting system such that the normal form of a  $\lambda\mathbf{J}^{\mathbf{m}}$ -term is its image under the mapping  $\phi : \lambda\mathbf{J}^{\mathbf{m}} \rightarrow \lambda$ . Moreover we get a permutability theorem: two  $\lambda\mathbf{J}^{\mathbf{m}}$ -terms have the same  $\phi$ -image iff they are inter-permutable.

**THEOREM 4.1 MAIN RESULTS ON PERMUTATIVE CONVERSIONS.**

- (1) *Permutability.* For all  $t, t' \in \mathcal{T}\mathbf{J}^{\mathbf{m}}$ ,  $\phi(t) = \phi(t')$  iff  $t \leftrightarrow_{pq}^* t'$ .
- (2) *Characterisation of pq-normal forms.* For all  $t \in \mathcal{T}\mathbf{J}^{\mathbf{m}}$ ,  $t$  is pq-normal iff  $t \in \mathcal{T}$ .
- (3) *Confluence and termination.*  $\rightarrow_{pq}$  is confluent and terminating.
- (4) *Representation of  $\phi$ .* For all  $t \in \mathcal{T}\mathbf{J}^{\mathbf{m}}$ ,  $\phi(t) = \downarrow_{pq}(t)$ .

**PROOF.** See Appendix C for detailed proofs. Two crucial properties used in these proofs are, for all  $t, t' \in \mathcal{T}\mathbf{J}^{\mathbf{m}}$ , (1) if  $t \rightarrow_{pq}^* t'$  then  $\phi(t) = \phi(t')$ , and (2)  $t \rightarrow_{pq}^* \phi(t)$ .  $\square$

In Appendix C we show that this sequence of results can be analogously established for mappings  $\mathbf{p}^{\mathbf{m}}$  and  $\mathbf{p}$  w.r.t.  $p$ -permutations and also for mappings  $\mathbf{q}\mathbf{J}$  and  $\mathbf{q}$  w.r.t. the permutation  $q$ .

#### 4.2 Properties of $\mu$

We prove for  $\mu$  properties of confluence, termination, postponement, and commutation.

##### **Confluence and termination.**

**THEOREM 4.2.**  $\rightarrow_{\mu}$  is confluent and terminating.

**PROOF.** Confluence of  $\rightarrow_{\mu}$  is a consequence of the diamond property: in  $\lambda\mathbf{J}^{\mathbf{m}}$ , if  $t \rightarrow_{\mu} t_1$  and  $t \rightarrow_{\mu} t_2$  then there is  $t_3$  such that  $t_1 \rightarrow_{\mu} t_3$  and  $t_2 \rightarrow_{\mu} t_3$ . The latter, in turn, follows from associativity of operation  $\mathbf{a}$ . Termination of  $\rightarrow_{\mu}$  follows from the fact that each  $\mu$ -reduction step decreases by one the number of  $\mu$ -redexes.  $\square$

**Postponement.** In  $\lambda\mathbf{J}^{\mathbf{m}}$  any  $\rightarrow_{\beta\pi\mu}$ -reduction sequence can be transformed into a reduction sequence consisting of two subsequences so that the first contains only  $\beta$  and  $\pi$ -steps and the second contains only  $\mu$ -steps. This property (Theorem 4.5 below) is used in particular to guarantee confluence of  $\rightarrow_{\beta\pi\mu}$ ; its proof uses a new form of reduction, denoted by  $\rightarrow_{\pi+}$ , which generalises  $\rightarrow_{\pi}$ .

The difficulty in arguing about  $\mu$ -postponement directly in terms of  $\rightarrow_{\pi}$  has to do with the fact that, when performing a  $\pi$ -reduction, a new  $\pi$ -redex can be generated

and a  $\mu$ -redex may become hidden, as in the following example:

$$\begin{aligned} t_0(u_0, l_0, (y)y(u_1, l_1, (z)v_1))(u, l, (x)v) &=: t_1 \\ \rightarrow_\pi t_0(u_0, l_0, (y)y(u_1, l_1, (z)v_1))(u, l, (x)v) &=: t'_1 \\ \rightarrow_\pi t_0(u_0, l_0, (y)y(u_1, l_1, (z)v_1(u, l, (x)v))) &=: t_4 \end{aligned} \quad (19)$$

Observe that the head of  $t_1$  is a  $\mu$ -redex which becomes hidden in the  $\pi$ -contractum  $t'_1$ ; the  $\mu$ -redex is recovered by performing another  $\pi$ -reduction.

The mentioned difficulty occurs in a situation like this:

$$\begin{aligned} t_0(u_0, l_0, (y)y(u_1, l_1, (z)v_1))(u, l, (x)v) &=: t_1 \\ \rightarrow_\mu t_0(u_0, \mathbf{a}(l_0, u_1, l_1), (z)v_1)(u, l, (x)v) &=: t_2 \\ \rightarrow_\pi t_0(u_0, \mathbf{a}(l_0, u_1, l_1), (z)v_1(u, l, (x)v)) &=: t_3 \end{aligned} \quad (20)$$

Indeed  $t_4 \rightarrow_\mu t_3$ , but the recovery of the  $\mu$ -redex in  $t_4$  costed two  $\pi$ -steps. Now the proof of part 2 of Lemma 4.4 below demands exactly one reduction step from  $t_1$  to  $t_4$ , and it is precisely for collecting such sequences of  $\pi$ -steps in a single step of reduction that the new form of reduction  $\rightarrow_{\pi_+}$  is introduced.

*Definition 4.3.* Let the rule  $\pi_n$ , for each natural  $n$ , be as follows:

$$(\pi_n) \quad t(u, l, (x)v)(u', l', (y)v') \rightarrow t(u, l, (x)\textcircled{y}_n^x(v, u', l', y, v'))$$

where

$$\begin{aligned} \textcircled{y}_1^y(t, u, l, x, v) &= t(u, l, (x)v) \\ \textcircled{y}_{n+1}^y(t, u, l, x, v) &= \begin{cases} y(u', l', (z)\textcircled{z}_n^z(v', u, l, x, v)), \\ \quad \text{if } t = y(u', l', (z)v') \text{ and } y \notin u', v', l' \\ t(u, l, (x)v), \quad \text{otherwise} \end{cases} \end{aligned}$$

Let  $\pi_+ = \bigcup_{n \geq 1} \pi_n$  and, as before, let  $\rightarrow_{\pi_+}$  be the compatible closure of  $\pi_+$ .

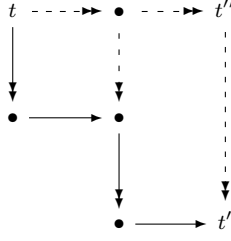
Observe that: (i)  $\pi_1 = \pi$ ; (ii)  $\pi_n \subseteq \bigcup_{i=1}^n \rightarrow_{\pi}^i \subseteq \rightarrow_{\pi}^+$ ; (iii)  $\rightarrow_{\pi_+} \subseteq \rightarrow_{\pi}^+$ .

LEMMA 4.4. *In  $\lambda\mathbf{J}^m$ :*

- (1) *If  $t_1 \rightarrow_\mu t_2$  and  $t_2 \rightarrow_\beta t_3$  (resp.  $t_2 \rightarrow_{\pi_+} t_3$ ) then there is  $t_4$  such that  $t_1 \rightarrow_\beta t_4$  (resp.  $t_1 \rightarrow_{\pi_+} t_4$ ) and  $t_4 \rightarrow_\mu^* t_3$ .*
- (2) *If  $t_1 \rightarrow_\mu^* t_2$  and  $t_2 \rightarrow_{\beta, \pi_+} t_3$  then there is  $t_4$  such that  $t_1 \rightarrow_{\beta, \pi_+} t_4$  and  $t_4 \rightarrow_\mu^* t_3$ .*
- (3) *If  $t_1 \rightarrow_\mu^* t_2$  and  $t_2 \rightarrow_{\beta\pi} t_3$  then there is  $t_4$  such that  $t_1 \rightarrow_{\beta\pi}^+ t_4$  and  $t_4 \rightarrow_\mu^* t_3$ .*

PROOF. 1. Let  $\rho$  be the  $\mu$ -redex contracted in  $t_1 \rightarrow_\mu t_2$ , and  $\sigma'$  be the  $\beta$ -redex (resp.  $\pi_+$ -redex) contracted in  $t_2$ . Then  $\sigma'$  is the residual of a  $\beta$ -redex (resp.  $\pi_+$ -redex)  $\sigma$  in  $t_1$ . The proof proceeds by the analysis of the relative positions of  $\sigma$  and  $\rho$  in  $t_1$ . If there is no overlap between  $\sigma$  and  $\rho$ , the result is trivial. If  $\sigma$  is a strict subterm of  $\rho$ , the analysis is easy, with some subcases using the fact that  $\rightarrow_\mu$  is compatible with operation  $\mathbf{a}$ . If  $\sigma = \rho$ , there are two interesting subcases. In one of these subcases,  $\sigma$  is a  $\pi_+$ -redex and the result follows from

$$\text{if } y \notin u', l', v' \text{ then } \textcircled{x}_n^x(t, u, l, y, y(u', l', (z)v')) \rightarrow_\mu \textcircled{x}_n^x(t, u, \mathbf{a}(l, u', l'), z, v').$$


Fig. 6.  $\mu$ -postponement

This is readily proved by induction on  $n$ . The other subcase is as follows:

$$\begin{aligned} t_1 &= (\lambda z.t_0)(u_1, [], (x)x(u_2, l_2, (y)v)), \\ \rightarrow_{\mu} t_2 &= (\lambda z.t_0)(u_1, u_2 :: l_2, (y)v), \\ \rightarrow_{\beta_2} t_3 &= \mathbf{s}(u_1, z, t_0)(u_2, l_2, (y)v), \end{aligned}$$

where  $x \notin u_2, l_2, v$ . In this case,  $t_1 \rightarrow_{\beta_1} t_3$ . Finally, if  $\rho$  is a strict subterm of  $\sigma$ , there are several subcases that follow from compatibility of  $\rightarrow_{\mu}$  with operations  $\mathbf{s}$  and  $@_n^x$ , but the only interesting subcase relates to the discussion anteceding Definition 4.3.

$$\begin{aligned} t_1 &= t_0(u_0, l_0, (y)y(v_2, l_2, (w)v_3))(u_1, l_1, (z)v_1), \\ \rightarrow_{\mu} t_2 &= t_0(u_0, \mathbf{a}(l_0, v_2, l_2), (w)v_3)(u_1, l_1, (z)v_1), \\ \rightarrow_{\pi_n} t_3 &= t_0(u_0, \mathbf{a}(l_0, v_2, l_2), (w)@_n^w(v_3, u_1, l_1, z, v_1)), \end{aligned}$$

where  $y \notin v_2, l_2, v_3$ . In this case,

$$\begin{aligned} t_1 &\rightarrow_{\pi_{n+1}} t_0(u_0, l_0, (y)@_{n+1}^y(y(v_2, l_2, (w)v_3), u_1, l_1, z, v_1)) \\ &= t_0(u_0, l_0, (y)y(v_2, l_2, (w)@_n^w(v_3, u_1, l_1, z, v_1))) \\ &\rightarrow_{\mu} t_3. \end{aligned}$$

2. By induction on the length of the reduction sequence  $t_1 \rightarrow_{\mu}^* t_2$ . For length zero the result is trivial. For length greater than zero the result follows from the induction hypothesis and from 1 (Here we see that, in order to compose the induction hypothesis with statement 1, it is crucial that this statement asserts the existence of exactly one reduction step from  $t_1$  to  $t_4$ .)

3. Immediate from 2,  $\pi \subseteq \pi_+$ , and  $\rightarrow_{\pi_+} \subseteq \rightarrow_{\pi}^+$ .  $\square$

The  $\mu$ -postponement theorem is now a consequence of part 3 of this lemma.

**THEOREM 4.5 POSTPONEMENT OF  $\mu$ .** *If  $t \rightarrow_{\beta\pi\mu}^* t'$  in  $\lambda\mathbf{J}^m$  then there is  $t''$  such that  $t \rightarrow_{\beta\pi}^* t''$  and  $t'' \rightarrow_{\mu}^* t'$  in  $\lambda\mathbf{J}^m$ .*

**PROOF.** The proof can be read without loss of generality from the diagram in Figure 6, where reduction  $t \rightarrow_{\beta\pi\mu}^* t'$  is represented by the solid arrows. Vertical (resp. horizontal) arrows represent  $\mu$ -steps (resp.  $\beta\pi$ -steps). Each rectangle holds by an application of part 3 of Lemma 4.4. The required reduction  $t \rightarrow_{\beta\pi}^* t''$  is obtained from the topmost sequence of horizontal arrows (each containing at least a  $\beta\pi$ -step) and the required reduction  $t'' \rightarrow_{\mu}^* t'$  corresponds to the rightmost arrow.  $\square$

Statement 3 of Lemma 4.4 is also used later in proving termination of reduction. In such application it will be crucial that each of the  $\beta$  or  $\pi$ -steps succeeding a  $\mu$ -step originates at least one  $\beta$  or  $\pi$ -step after postponement.

**Commutation.** In order to obtain confluence of  $\rightarrow_{\beta\pi\mu}$  from confluence of  $\rightarrow_{\beta\pi}$  and  $\rightarrow_{\mu}$  as in Subsection 4.5, besides  $\mu$ -postponement, some form of commutation of  $\mu$  with  $\beta$  and  $\pi$  is also required. Although  $\mu$  commutes with  $\beta$  (Theorem 4.7 below), it does not commute with  $\pi$ .

Again the problem is that, when performing a  $\pi$ -reduction, newly generated  $\pi$ -redexes may hide  $\mu$ -redexes. Recalling the terms in (19) and (20), there is no common  $\pi$ -reduct of  $t_2$  and  $\mu$ -reduct of  $t'_1$ . In particular, no  $\pi_+$ -reduct of  $t_2$  would help. So, given that  $t_1 \rightarrow t'_1$  is a  $\pi_+$ -step, rule  $\pi_+$  does not commute with  $\mu$  either.

Instead of  $\pi_+$  we use the “eager” version  $\pi'$  of rule  $\pi$ , already considered in [Espírito Santo and Pinto 2004] in the context of  $\lambda\mathbf{J}$ . This rule also performs in a single step sequences of  $\pi$ -reductions, but contrary to  $\pi_+$ ,  $\pi$ -reduction *must* carry on while newly generated  $\pi$ -redexes hide  $\mu$ -redexes. For  $\pi'$  commutation with  $\mu$  will hold (Theorem 4.7).

*Definition 4.6.* The rule  $\pi'$  is the following:

$$(\pi') \quad t(u, l, (x)v)(u', l', (y)v') \rightarrow t(u, l, (x)@'(x, v, u', l', y, v'))$$

where

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

Observe that  $\rightarrow_{\pi'} \subseteq \rightarrow_{\pi}^+$  and that a term is  $\pi'$ -normal if and only if it is  $\pi$ -normal.

**THEOREM 4.7 COMMUTATION WITH  $\mu$ .** *In  $\lambda\mathbf{J}^m$ , if  $t_1 \rightarrow_{\mu} t_2$  and  $t_1 \rightarrow_{\beta} t_3$  (resp.  $t_1 \rightarrow_{\pi'} t_3$ ), then there is  $t_4$  such that  $t_1 \rightarrow_{\mu}^* t_4$  and  $t_2 \rightarrow_{\beta}^* t_4$  (resp.  $t_2 \rightarrow_{\pi'}^* t_4$ ).*

**PROOF.** Very similar to the proof of part 1 of Lemma 4.4. Let  $\rho$  be the  $\mu$ -redex contracted in  $t_1 \rightarrow_{\mu} t_2$ , and  $\sigma$  be the  $\beta$ - or  $\pi'$ -redex contracted in  $t_1 \rightarrow t_3$ . Again, the proof proceeds by the analysis of the relative positions of  $\sigma$  and  $\rho$  in  $t_1$ . The fact that  $\rightarrow_{\mu}$  is compatible with operation  $@'$  is used. Additionally, the only novelty is in one of the subcases of  $\sigma = \rho$ , more precisely when  $\sigma$  is a  $\pi'$ -redex. There one needs the fact

$$\text{if } y \notin u', l', v' \text{ then } @'(x, t, u, l, y, y(u', l', (z)v')) \rightarrow_{\mu} @'(x, t, u, \mathbf{a}(l, u', l'), z, v'),$$

proved by a straightforward induction on  $t$ .  $\square$

### 4.3 Preservation of reduction

Now we offer a detailed study of how the various mappings of  $\lambda\mathbf{J}^m$  into subsystems preserve reduction steps. The properties obtained are used not only to achieve conservativeness of a system over its subsystems but also to infer confluence and normalisation properties of  $\lambda\mathbf{J}^m$ , as well as preservation of strong normalisation.

In order to prove results of preservation of strong normalisation, we introduce the notion of garbage-free terms. An expression of  $\lambda\mathbf{J}^{\mathbf{m}}$  (resp. term of  $\lambda\mathbf{J}$ ) is said *garbage-free* when, for every gm-application  $t(u, l, (x)v)$  (resp. g-application  $t(u, (x)v)$ ) occurring as a subterm of it,  $x \in v$ <sup>16</sup>.

LEMMA 4.8.

- (1) For  $t, u$  in  $\lambda\mathbf{J}^{\mathbf{m}}$  (resp.  $\lambda\mathbf{J}$ ), if  $t$  is garbage-free and  $t \rightarrow_{\beta\pi\mu} u$  in  $\lambda\mathbf{J}^{\mathbf{m}}$  (resp.  $t \rightarrow_{\beta\pi} u$  in  $\lambda\mathbf{J}$ ), then  $u$  is garbage-free.
- (2) For  $t$  in  $\lambda\mathbf{J}^{\mathbf{m}}$  (resp.  $\lambda\mathbf{J}$ ), if  $x \in t$  and  $t$  is garbage-free,  $x \in \mathbf{p}^{\mathbf{m}}(t)$  (resp.  $x \in \mathbf{p}(t)$ ).

PROOF. The statements about  $\lambda\mathbf{J}$  are proved by easy inductions. The statements about  $\lambda\mathbf{J}^{\mathbf{m}}$  are proved together with analogous statements for lists by easy simultaneous inductions.  $\square$

Now we tackle preservation of reduction by the mappings  $\mathbf{p}^{\mathbf{m}}$  and  $\mathbf{p}$ , the results to the latter being corollaries of the results to the former.

PROPOSITION 4.9 PRESERVATION OF REDUCTION BY  $\mathbf{p}^{\mathbf{m}}$ .

- (1) If  $t \rightarrow_{\beta_i} t'$  in  $\lambda\mathbf{J}^{\mathbf{m}}$  then (i)  $\mathbf{p}^{\mathbf{m}}(t) \rightarrow_{\beta_i}^* \mathbf{p}^{\mathbf{m}}(t')$  in  $\lambda^{\mathbf{m}}$ , and (ii) if  $t$  is garbage-free then  $\mathbf{p}^{\mathbf{m}}(t) \rightarrow_{\beta_i}^+ \mathbf{p}^{\mathbf{m}}(t')$  in  $\lambda^{\mathbf{m}}$ .
- (2) If  $t \rightarrow_{\pi} t'$  in  $\lambda\mathbf{J}^{\mathbf{m}}$  then  $\mathbf{p}^{\mathbf{m}}(t) = \mathbf{p}^{\mathbf{m}}(t')$ .
- (3) If  $t \rightarrow_{\mu} t'$  in  $\lambda\mathbf{J}^{\mathbf{m}}$  then (i)  $\mathbf{p}^{\mathbf{m}}(t) \rightarrow_h^* \mathbf{p}^{\mathbf{m}}(t')$  in  $\lambda^{\mathbf{m}}$ , and (ii) if  $t$  is garbage-free then  $\mathbf{p}^{\mathbf{m}}(t) \rightarrow_h^+ \mathbf{p}^{\mathbf{m}}(t')$  in  $\lambda^{\mathbf{m}}$ .
- (4) If  $t \rightarrow_h t'$  in  $\lambda\mathbf{J}^{\mathbf{m}}$  then (i)  $\mathbf{p}^{\mathbf{m}}(t) \rightarrow_h^* \mathbf{p}^{\mathbf{m}}(t')$  in  $\lambda^{\mathbf{m}}$ , and (ii) if  $t$  is garbage-free then  $\mathbf{p}^{\mathbf{m}}(t) \rightarrow_h^+ \mathbf{p}^{\mathbf{m}}(t')$  in  $\lambda^{\mathbf{m}}$ .

PROOF. Notice that statement 4 is a corollary of statements 2 and 3 (since  $t \rightarrow_h t'$  implies that there exists  $t''$  such that  $t \rightarrow_{\pi} t''$  and  $t'' \rightarrow_{\mu} t'$ , and  $t''$  is garbage-free when  $t$  is). It remains a statement for each  $R \in \{\beta_1, \beta_2, \pi, \mu\}$ . Each of these is proved together with a similar statement for lists by simultaneous induction on  $t \rightarrow_R t'$  and  $l \rightarrow_R l'$ . The inductive cases are routine and use the fact that  $\rightarrow_R^*$ ,  $\rightarrow_R^+$  and  $=$  are compatible. The proof of the base cases requires the substitution lemma for m-substitution, compatibility of  $\rightarrow_{\beta, h}$  with m-substitution, specifically

- (1) for all  $t, u, v$  in  $\lambda^{\mathbf{m}}$  and  $R \in \{\beta_1, \beta_2, h\}$ , if  $t \rightarrow_R u$  then  $\mathbf{s}(v, x, t) \rightarrow_R \mathbf{s}(v, x, u)$ ,  $\mathbf{s}(t, x, v) \rightarrow_R^* \mathbf{s}(u, x, v)$ , and moreover, if  $x \in v$  then  $\mathbf{s}(t, x, v) \rightarrow_R^+ \mathbf{s}(u, x, v)$ ,

and requires also the following two commutation properties of  $\mathbf{p}^{\mathbf{m}}$ :

- (1) for all  $t, u$  in  $\lambda\mathbf{J}^{\mathbf{m}}$ ,  $\mathbf{p}^{\mathbf{m}}(\mathbf{s}(u, x, t)) = \mathbf{s}(\mathbf{p}^{\mathbf{m}}(u), x, \mathbf{p}^{\mathbf{m}}(t))$ ;
- (2) for all  $l, u, l'$  in  $\lambda\mathbf{J}^{\mathbf{m}}$ ,  $\mathbf{p}^{\mathbf{m}'}(\mathbf{a}(l, u, l')) = \mathbf{a}(\mathbf{p}^{\mathbf{m}'}(l), \mathbf{p}^{\mathbf{m}}(u), \mathbf{p}^{\mathbf{m}'}(l'))$ .

We just check the base cases. Below we generally abbreviate  $\mathbf{p}^{\mathbf{m}}$  and  $\mathbf{p}^{\mathbf{m}'}$  by  $p$  and omit some parentheses.

<sup>16</sup>If the application  $t(u, l, (x)v)$  is thought of as the substitution of  $t(u, l)$  for  $x$  in  $v$ , then, in the case  $x \notin v$ , it is natural to regard it as equal to  $v$  and to regard the components  $t, u, l$  as *garbage*. In explicit substitutions calculi the reduction to  $v$  in such case is called *garbage collection*.

$$\begin{aligned}
\text{Case } \beta_1. \quad p((\lambda x.t)(u, [], (y)v)) &= \mathbf{s}((\lambda x.pt)(pu, [], y, pv)) \\
&\rightarrow_{\beta_1}^* \mathbf{s}(\mathbf{s}(pu, x, pt), y, pv) \\
&= \mathbf{s}(p(\mathbf{s}(u, x, t)), y, pv) \\
&= p(\mathbf{s}(\mathbf{s}(u, x, t), y, v))
\end{aligned}$$

The first equality is by definition of  $\mathbf{p}^{\mathbf{m}}$ , the reduction uses compatibility of  $\mathbf{s}$  with  $\rightarrow_{\beta_1}$  and the other equalities follow by commutation of  $\mathbf{s}$  with  $\mathbf{p}^{\mathbf{m}}$ . For (ii), observe additionally that compatibility of  $\mathbf{s}$  with  $\rightarrow_{\beta}$  guarantees one or more  $\beta_1$ -steps, because, when  $(\lambda x.t)(u, [], (y)v)$  is garbage-free,  $y \in v$  and  $v$  is also garbage-free, hence, by Lemma 4.8,  $y \in pv$ .

Case  $\beta_2$ : similar to case  $\beta_1$ .

$$\begin{aligned}
\text{Case } \pi. \quad p(t(u, l, (x)v)(u', l', (y)v')) &= \mathbf{s}(\mathbf{s}(pt)(pu, pl, x, pv)(pu', pl', y, pv')) \\
&= \mathbf{s}(\mathbf{s}(pt)(pu, pl, x, (pv)(pu', pl')), y, \mathbf{s}(pt)(pu, pl, x, pv')) \\
&= \mathbf{s}(pt)(pu, pl, x, \mathbf{s}(pv)(pu', pl', y, pv')) \\
&= p(t(u, l, (x)v)(u', l', (y)v'))
\end{aligned}$$

Here the first and last equalities are by definition of  $\mathbf{p}^{\mathbf{m}}$ , the second by  $x \notin u', l', v'$ , and the third by the substitution lemma.

$$\begin{aligned}
\text{Case } \mu. \quad p(t(u, l, (x)x)(u', l', (y)v)) &= \mathbf{s}((pt)(pu, pl, x, \mathbf{s}(x)(pu', pl', y, pv))) \\
&= \mathbf{s}(\mathbf{s}((pt)(pu, pl, x, x)(pu', pl')), y, \mathbf{s}((pt)(pu, pl, x, pv))) \\
&= \mathbf{s}((pt)(pu, pl)(pu', pl', y, pv)) \\
&\rightarrow_h^* \mathbf{s}((pt)(pu, \mathbf{a}(pl, pu', pl')), y, pv) \\
&= \mathbf{s}((pt)(pu, p(\mathbf{a}(l, u', l'))), y, pv) \\
&= p(t(u, \mathbf{a}(l, u', l'), (y)v))
\end{aligned}$$

Here the first and last equalities are by definition of  $\mathbf{p}^{\mathbf{m}}$ , the second by the substitution lemma, the third by  $x \notin u', l', v'$ , and the penultimate by commutation of  $\mathbf{a}$  with  $\mathbf{p}^{\mathbf{m}}$ . The reduction holds by compatibility of  $\mathbf{s}$  with  $\rightarrow_h$ . For (ii), as in the case of  $\beta_1$ , observe that  $\rightarrow_h^*$  can be replaced by  $\rightarrow_h^+$ , because, when  $t(u, l, (x)x)(u', l', (y)v)$  is garbage-free,  $y \in pv$  holds.  $\square$

**COROLLARY 4.10 PRESERVATION OF REDUCTION BY  $\mathbf{p}$ .**

- (1) If  $t \rightarrow_{\beta_1} t'$  in  $\lambda\mathbf{J}$  then (i)  $\mathbf{p}(t) \rightarrow_{\beta_1}^* \mathbf{p}(t')$  in  $\lambda$ , and (ii) if  $t$  is garbage-free then  $\mathbf{p}(t) \rightarrow_{\beta_1}^+ \mathbf{p}(t')$  in  $\lambda$ .
- (2) If  $t \rightarrow_{\pi} t'$  in  $\lambda\mathbf{J}$  then  $\mathbf{p}(t) = \mathbf{p}(t')$ .

**PROOF.** As to statement 1, suppose  $t \rightarrow_{\beta_1} t'$  in  $\lambda\mathbf{J}$ . From part 1 of Proposition 3.8 we get  $t \rightarrow_{\beta_1} t'$  in  $\lambda\mathbf{J}^{\mathbf{m}}$ , hence, by Proposition 4.9: (a)  $\mathbf{p}^{\mathbf{m}}(t) \rightarrow_{\beta_1}^* \mathbf{p}^{\mathbf{m}}(t')$  in  $\lambda^{\mathbf{m}}$ ; (b) if  $t$  is garbage-free then  $\mathbf{p}^{\mathbf{m}}(t) \rightarrow_{\beta_1}^+ \mathbf{p}^{\mathbf{m}}(t')$  in  $\lambda^{\mathbf{m}}$ . From (a), because  $\mathbf{p}$  is the restriction of  $\mathbf{p}^{\mathbf{m}}$  to  $\lambda\mathbf{J}$ -terms,  $\mathbf{p}(t) \rightarrow_{\beta_1}^* \mathbf{p}(t')$  in  $\lambda^{\mathbf{m}}$ . Therefore, since  $\mathbf{p}(t)$  and  $\mathbf{p}(t')$  are in  $\lambda$ , it follows from Proposition 3.9 that  $\mathbf{p}(t) \rightarrow_{\beta_1}^* \mathbf{p}(t')$  in  $\lambda$ , thus proving (i). (ii) is analogously obtained from (b).

Statement 2 is similar but simpler to 1 (no need to use Proposition 3.9).  $\square$

We now move to preservation of reduction by  $\mathbf{qJ}$  and  $\mathbf{q}$ . Contrary to  $\mathbf{p}^{\mathbf{m}}$ , each reduction step in  $\lambda\mathbf{J}^{\mathbf{m}}$  is mapped by  $\mathbf{qJ}$  into at least a reduction step. This is a crucial property when inferring termination of the source system from that of the target, as we do in the proof of Theorem 4.20. The following lemma collects some of the properties required for the results of preservation of reduction by  $\mathbf{qJ}$ .

LEMMA 4.11. For all  $t, u, l, v, u', l', v'$  in  $\lambda\mathbf{J}^m$ :

- (1)  $\mathbf{qJ}(t(u, (x)v)(u', l', (y)v')) \rightarrow_{\pi}^+ \mathbf{qJ}(t(u, (x)v(u', l', (y)v')))$ .
- (2)  $\mathbf{qJ}(t(u, l, (x)v)(u', l', (y)v')) \rightarrow_{\pi}^+ \mathbf{qJ}(t(u, l, (x)v(u', l', (y)v')))$ .
- (3)  $\mathbf{qJ}(t(u, l, (x)x)(u', l', (y)v')) = \mathbf{qJ}(t(u, \mathbf{a}(l, u', l'), (y)v'))$ .

PROOF. 1. One proves, by induction on  $l'$  that, for all  $t_0, u_0, v_0, u'_0, v'_0$  in  $\lambda\mathbf{J}$ , and all  $l'$  in  $\lambda\mathbf{J}^m$ , one has

$$\mathbf{qJ}'(t_0(u_0, (x)v_0), u'_0, l', y, v'_0) \rightarrow_{\pi}^+ t_0(u_0, (x)\mathbf{qJ}'(v_0, u'_0, l', y, v'_0)) .$$

We need the following property of  $\mathbf{qJ}'$ : if  $t \rightarrow_R t'$  in  $\lambda\mathbf{J}$ , then also  $\mathbf{qJ}'(t, u, l, x, v) \rightarrow_R \mathbf{qJ}'(t', u, l, x, v)$  in  $\lambda\mathbf{J}$ . The proof of this property is a routine induction on  $l$ .

2. One proves, by induction on  $l$  that, for all  $t_0, u_0, v_0, u'_0, v'_0$  in  $\lambda\mathbf{J}$ , and all  $l, l'$  in  $\lambda\mathbf{J}^m$ , one has

$$\mathbf{qJ}'(\mathbf{qJ}'(t_0, u_0, l, x, v_0), u'_0, l', y, v'_0) \rightarrow_{\pi}^+ \mathbf{qJ}'(t_0, u_0, l, x, \mathbf{qJ}'(v_0, u'_0, l', y, v'_0)) .$$

The base case follows by 1.

3. One proves, by induction on  $l$  that, for all  $t_0, u_0, v_0, v'_0$  in  $\lambda\mathbf{J}$ , and all  $u', l, l'$  in  $\lambda\mathbf{J}^m$ , one has

$$\mathbf{qJ}'(\mathbf{qJ}'(t_0, u_0, l, x, x), \mathbf{qJ}(u'), l', y, v'_0) = \mathbf{qJ}'(t_0, u_0, \mathbf{a}(l, u', l'), y, v'_0) .$$

□

PROPOSITION 4.12 (CO-)PRESERVATION OF REDUCTION BY  $\mathbf{qJ}$ .

- (1) If  $t \rightarrow_{\beta_1} t'$  in  $\lambda\mathbf{J}^m$  then  $\mathbf{qJ}(t) \rightarrow_{\beta_1} \mathbf{qJ}(t')$  in  $\lambda\mathbf{J}$ .
- (2) If  $t \rightarrow_{\pi} t'$  in  $\lambda\mathbf{J}^m$  then  $\mathbf{qJ}(t) \rightarrow_{\pi}^+ \mathbf{qJ}(t')$  in  $\lambda\mathbf{J}$ .
- (3) If  $t \rightarrow_{\mu} t'$  in  $\lambda\mathbf{J}^m$  then  $\mathbf{qJ}(t') \rightarrow_{\pi}^+ \mathbf{qJ}(t)$  in  $\lambda\mathbf{J}$ .
- (4) If  $t \rightarrow_h t'$  in  $\lambda\mathbf{J}^m$  then  $\mathbf{qJ}(t) = \mathbf{qJ}(t')$ .

PROOF. Notice that in statement 3 reduction in the target goes in the opposite direction (hence “co-preservation”) and moreover it is a corollary of statements 2 and 4 (since  $\mu \subseteq h \circ \pi^{-1}$ , one has  $\rightarrow_{\mu} \subseteq \rightarrow_{\pi^{-1}, h}^+$ ). It remains a statement for each  $R \in \{\beta_1, \beta_2, \pi, h\}$ . Each of these is proved together with a similar statement for lists by simultaneous induction on  $t \rightarrow_R t'$  and  $l \rightarrow_R l'$ . The inductive cases are routine and use the fact that  $\rightarrow_R^*$  and  $=$  are compatible. We check the base cases. They require the previous lemma and commutation of  $\mathbf{s}$  with  $\mathbf{qJ}$  as follows:  $\mathbf{qJ}(\mathbf{s}(u, x, t)) = \mathbf{s}(\mathbf{qJ}(u), x, \mathbf{qJ}(t))$ , for all  $t, u$  in  $\lambda\mathbf{J}^m$ . Below we generally abbreviate  $\mathbf{qJ}$  and  $\mathbf{qJ}'$  by  $q$  and omit some parentheses.

$$\begin{aligned} \text{Case } \beta_1. \quad q((\lambda x.t)(u, [], (y)v)) &= (\lambda x.qt)(qu, (y)qv) \\ &\rightarrow_{\beta_1} \mathbf{s}(\mathbf{s}(qu, x, qt), y, qv) \\ &= \mathbf{s}(q(\mathbf{s}(u, x, t)), y, qv) \\ &= q(\mathbf{s}(\mathbf{s}(u, x, t), y, v)) \end{aligned}$$

The last two equalities are by commutation of  $\mathbf{s}$  with  $\mathbf{qJ}$  and the first by definition of  $\mathbf{qJ}$ .

$$\begin{aligned}
\text{Case } \beta_2. \quad q((\lambda x.t)(u, u' :: l, (y)v)) &= q(\lambda x.qt, qu, u' :: l, y, qv) \\
&= q(\lambda x.qt)[qu], qu', l, y, qv) \\
&\rightarrow_{\beta_1} q(\mathbf{s}(qu, x, qt), qu', l, y, qv) \\
&= q(q(\mathbf{s}(u, x, t)), qu', l, y, qv) \\
&= q(\mathbf{s}(u, x, t)(u', l, (y)v))
\end{aligned}$$

The reduction holds by the property of  $\mathbf{qJ}'$  mentioned in the proof of part 1 of the previous lemma. The penultimate equality is by commutation of  $\mathbf{s}$  with  $\mathbf{qJ}$  and the others by definition of  $\mathbf{qJ}$  or  $\mathbf{qJ}'$ .

Cases  $\pi$  and  $h$  are respectively parts 2 and 3 of the previous lemma.  $\square$

**COROLLARY 4.13 PRESERVATION OF REDUCTION BY  $\mathbf{q}$ .**<sup>17</sup>

- (1) If  $t \rightarrow_{\beta_i} t'$  in  $\lambda^{\mathbf{m}}$  then  $\mathbf{q}(t) \rightarrow_{\beta_1} \mathbf{q}(t')$  in  $\lambda$ .
- (2) If  $t \rightarrow_h t'$  in  $\lambda^{\mathbf{m}}$  then  $\mathbf{q}(t) = \mathbf{q}(t')$ .

**PROOF.** The proof can be obtained as in Corollary 4.10, replacing  $\mathbf{p}^{\mathbf{m}}$  and  $\mathbf{p}$  by  $\mathbf{qJ}$  and  $\mathbf{q}$  resp.,  $\lambda^{\mathbf{m}}$  by  $\lambda\mathbf{J}$ , Proposition 4.9 by Proposition 4.12 and part 1 of Proposition 3.8 by part 2 of the same proposition.  $\square$

Preservation of reduction by  $\phi$  is now a simple corollary of the preservation properties before.

**PROPOSITION 4.14 PRESERVATION OF REDUCTION BY  $\phi$ .**

- (1) If  $t \rightarrow_{\beta_i} t'$  in  $\lambda\mathbf{J}^{\mathbf{m}}$  then  $\phi(t) \rightarrow_{\beta_1}^* \phi(t')$  in  $\lambda$ .
- (2) If  $t \rightarrow_{\pi, \mu, h} t'$  in  $\lambda\mathbf{J}^{\mathbf{m}}$  then  $\phi(t) = \phi(t')$ .

**PROOF.** It follows from  $\phi = \mathbf{q} \circ \mathbf{p}^{\mathbf{m}}$ , Proposition 4.9 and Corollary 4.13 (or from  $\phi = \mathbf{p} \circ \mathbf{qJ}$ , Proposition 4.12 and Corollary 4.10).  $\square$

The proof-theoretic reading of the last preservation result says how mapping  $\phi$  gives an interpretation of cut-elimination by means of normalisation, and therefore relates the result with the work of Zucker [1974] and Pottinger [1977].

#### 4.4 Conservativeness and preservation of strong normalisation

**Conservativeness.** With the help of the reduction preservation properties just established, here we strengthen the conservativeness results for reduction of Subsection 3.1.

The following result, which is a corollary of the  $\mu$ -postponement theorem (Theorem 4.5), plays a key role in proving conservativeness.

**LEMMA 4.15.** *If  $t \rightarrow_{\beta\pi\mu}^* t'$  in  $\lambda\mathbf{J}^{\mathbf{m}}$  and  $t' \in \lambda\mathbf{J}$ , then  $t \rightarrow_{\beta\pi}^* t'$  in  $\lambda\mathbf{J}^{\mathbf{m}}$ .*

**PROOF.** By Theorem 4.5 one has

$$\begin{array}{ccc}
t & \xrightarrow{\beta\pi} & t'' \\
& \searrow & \downarrow \mu \\
& & t'
\end{array}$$

<sup>17</sup>This result is implicit in [Espírito Santo 2002a].



From  $t'' \rightarrow_{\mu}^* t'$  and  $t' \in \lambda\mathbf{J}$ , it follows  $t'' = t'$ , for  $t'$  cannot contain a  $\mu$ -contractum. Hence  $t \rightarrow_{\beta\pi}^* t'$ .  $\square$

**THEOREM 4.16 CONSERVATIVENESS.**  $\lambda\mathbf{J}^{\mathbf{m}}$  is conservative over its subsystems, that is:

- (1) For all  $t, t' \in \lambda\mathbf{J}$ ,  $t \rightarrow_{\beta\pi\mu}^* t'$  in  $\lambda\mathbf{J}^{\mathbf{m}}$  iff  $t \rightarrow_{\beta\pi}^* t'$  in  $\lambda\mathbf{J}$ .
- (2) For all  $t, t' \in \lambda^{\mathbf{m}}$ ,  $t \rightarrow_{\beta\pi\mu}^* t'$  in  $\lambda\mathbf{J}^{\mathbf{m}}$  iff  $t \rightarrow_{\beta h}^* t'$  in  $\lambda^{\mathbf{m}}$ .
- (3) For all  $t, t' \in \lambda$ ,  $t \rightarrow_{\beta\pi\mu}^* t'$  in  $\lambda\mathbf{J}^{\mathbf{m}}$  iff  $t \rightarrow_{\beta}^* t'$  in  $\lambda$ .

**PROOF.** The “if” statements follow from the Definitions 3.2, 3.4, and 3.6 of  $\lambda\mathbf{J}$ ,  $\lambda^{\mathbf{m}}$ , and  $\lambda$ . As to the “only if” statements, we start by showing that of 3. Let  $t, t' \in \lambda$  and suppose  $t \rightarrow_{\beta\pi\mu}^* t'$  in  $\lambda\mathbf{J}^{\mathbf{m}}$ . From the fact that  $\phi$  preserves reduction (Proposition 4.14), it follows that  $\phi(t) \rightarrow_{\beta}^* \phi(t')$  in  $\lambda$ . But  $\phi(t) = t$  and  $\phi(t') = t'$ , as  $t, t' \in \lambda$ . The “only if” statement of 2 is proved in a similar fashion, using preservation of reduction by mapping  $\mathbf{p}^{\mathbf{m}}$  (Proposition 4.9). As to 1, the proof needs a slight adjustment, as mapping  $\mathbf{q}\mathbf{J}$  does not preserve  $\mu$ -steps. So, we start by using Lemma 4.15, and only then apply Proposition 4.12 in order to get  $\mathbf{q}\mathbf{J}(t) \rightarrow_{\beta\pi}^* \mathbf{q}\mathbf{J}(t')$  in  $\lambda\mathbf{J}$ . But  $\mathbf{q}\mathbf{J}(t) = t$  and  $\mathbf{q}\mathbf{J}(t') = t'$ , as  $t, t' \in \lambda\mathbf{J}$ .  $\square$

The following result, together with part 3 of the previous theorem, says that  $\lambda$  is conservatively extended by its supersystems.

**THEOREM 4.17 CONSERVATIVENESS.** <sup>18</sup> For all  $t, t' \in \lambda$ :

- (1)  $t \rightarrow_{\beta\pi}^* t'$  in  $\lambda\mathbf{J}$  iff  $t \rightarrow_{\beta}^* t'$  in  $\lambda$ .
- (2)  $t \rightarrow_{\beta h}^* t'$  in  $\lambda^{\mathbf{m}}$  iff  $t \rightarrow_{\beta}^* t'$  in  $\lambda$ .

**PROOF.** Analogous to the proof of the previous theorem, using Proposition 3.9 to justify the “if” statements, and the reduction preservation properties of  $\mathbf{p}$  and  $\mathbf{q}$  (resp.) to justify the “only if” statements.  $\square$

Part 1 of this theorem has the following proof-theoretic reading: normalisation in natural deduction with general elimination rules is conservative over normalisation in the sense of Prawitz.

**Preservation of strong normalisation.** We show that any term which is strongly normalising w.r.t. reduction in a given system is also strongly normalising w.r.t. reduction in each of the system’s supersystems. Such results are obtained with the help of the results on preservation of reduction by the respective mapping from the supersystem to the system. The following lemma is also needed.

**LEMMA 4.18.** (1)  $\rightarrow_{\pi\mu}$  in  $\lambda\mathbf{J}^{\mathbf{m}}$  is SN. (2)  $\rightarrow_{\pi}$  and  $\rightarrow_{\mu}$  and  $\rightarrow_h$  in  $\lambda\mathbf{J}^{\mathbf{m}}$  are SN. (3)  $\rightarrow_{\pi}$  in  $\lambda\mathbf{J}$  is SN. (4)  $\rightarrow_h$  in  $\lambda^{\mathbf{m}}$  is SN.

**PROOF.** 1. Proved in [Espírito Santo and Pinto 2004]. 2. Immediate from 1 and  $h \subseteq \mu \circ \pi$ . 3. Follows from 2 and the fact that an infinite  $\pi$ -reduction sequence in  $\lambda\mathbf{J}$  originates, by definition of  $\lambda\mathbf{J}$ , an infinite  $\pi$ -reduction sequence in  $\lambda\mathbf{J}^{\mathbf{m}}$ . 4. *mutatis mutandis*.  $\square$

<sup>18</sup>Part 2 of this theorem is implicit in [Espírito Santo 2002a].

## THEOREM 4.19 PRESERVATION OF STRONG NORMALISATION.

- (1) For all  $t \in \lambda^{\mathbf{m}}$ ,  $t$  is SN in  $\lambda^{\mathbf{m}}$  iff  $t$  is SN in  $\lambda\mathbf{J}^{\mathbf{m}}$ .
- (2) For all  $t \in \lambda\mathbf{J}$ ,  $t$  is SN in  $\lambda\mathbf{J}$  iff  $t$  is SN in  $\lambda\mathbf{J}^{\mathbf{m}}$ .
- (3) For all  $t \in \lambda$ ,  $t$  is SN in  $\lambda$  iff  $t$  is SN in  $\lambda^{\mathbf{m}}$ .
- (4) For all  $t \in \lambda$ ,  $t$  is SN in  $\lambda$  iff  $t$  is SN in  $\lambda\mathbf{J}$ .
- (5) For all  $t \in \lambda$ ,  $t$  is SN in  $\lambda$  iff  $t$  is SN in  $\lambda\mathbf{J}^{\mathbf{m}}$ .

PROOF. Part 5 follows immediately by combining either 1 and 3 or 2 and 4. As to the first four properties, for the “if” statements, observe that if there is an infinite reduction sequence in the subsystem, the same infinite reduction sequence exists in the supersystem. This is so by the definitions of the subsystems, in the case of 1 and 2; and by virtue of Proposition 3.9, in the case of 3 and 4. Now we address the “only if” statements.

Statement 1. Suppose there is an infinite reduction sequence,  $S$  say, in  $\lambda\mathbf{J}^{\mathbf{m}}$ , starting at  $t \in \lambda^{\mathbf{m}}$ . Because  $\rightarrow_{\pi}$  is terminating (Lemma 4.18), there are infinitely many  $\beta\mu$ -steps in  $S$ . Using map  $\mathbf{p}^{\mathbf{m}}$ , we obtain from  $S$  an infinite reduction sequence in  $\lambda\mathbf{J}$  starting at  $t = \mathbf{p}^{\mathbf{m}}(t)$ , since, by Proposition 4.9, each  $\beta\mu$ -step is mapped into one or more  $\beta h$ -steps (notice that  $t$ , being a  $\lambda^{\mathbf{m}}$ -term, is garbage-free and thus, by Lemma 4.8, all terms in  $S$  are garbage-free).

Statement 2. If there was an infinite reduction sequence in  $\lambda\mathbf{J}^{\mathbf{m}}$  starting at  $t \in \lambda\mathbf{J}$ , using the facts that  $\rightarrow_{\mu}$  is terminating (Lemma 4.18) and that  $\mu$ -steps can be postponed over  $\beta\pi$ -steps (see the argument in the proof of Theorem 4.5), there would be arbitrarily long reduction sequences of  $\beta\pi$ -steps in  $\lambda\mathbf{J}^{\mathbf{m}}$  starting at  $t$ . Thus, as each  $\beta\pi$ -step in  $\lambda\mathbf{J}^{\mathbf{m}}$  under  $\mathbf{q}\mathbf{J}$  gives rise to at least one  $\beta\pi$ -step in  $\lambda\mathbf{J}$  (Proposition 4.12), there would be arbitrarily long reduction sequences in  $\lambda\mathbf{J}$  starting at  $t = \mathbf{q}\mathbf{J}(t)$ .

Statements 3 and 4. They follow by similar arguments. In particular, to prove 3 we use the properties of map  $\mathbf{q}$  in Corollary 4.13 and the fact that  $\rightarrow_h$  is terminating in  $\lambda^{\mathbf{m}}$  (Lemma 4.18) and to prove 4 we use the properties of map  $\mathbf{p}$  in Corollary 4.10 and the facts that  $\rightarrow_{\pi}$  is terminating in  $\lambda\mathbf{J}$  (Lemma 4.18) and that garbage-free terms are closed for reduction in  $\lambda\mathbf{J}$  (Lemma 4.8).  $\square$

## 4.5 Strong normalisation and confluence of reduction

**Strong normalisation.** Strict preservation of  $\beta\pi$ -reduction steps by mapping  $\mathbf{q}\mathbf{J}$ , combined with the properties established for  $\mu$ , allow the lifting to  $\lambda\mathbf{J}^{\mathbf{m}}$  of the strong normalisation property of  $\Lambda J$  [Joachimski and Matthes 2003].

THEOREM 4.20 STRONG NORMALISATION. *There is no infinite  $\rightarrow_{\beta\pi\mu}$ -reduction sequence starting at a typable term of  $\lambda\mathbf{J}^{\mathbf{m}}$ .*

PROOF. Suppose there is such an infinite reduction sequence  $S$ , starting at typable term  $t \in \lambda\mathbf{J}^{\mathbf{m}}$ . Since  $\rightarrow_{\mu}$  is terminating,  $S$  contains infinitely many  $\beta\pi$ -reduction steps. As in the proof of statement 2 of Theorem 4.19, we can build an arbitrarily long reduction sequence in  $\lambda\mathbf{J}$  starting from  $\mathbf{q}\mathbf{J}(t)$ . But this is a typable term, since  $\mathbf{q}\mathbf{J}$  preserves typability (Proposition 3.11), contradicting strong normalisation of  $\lambda\mathbf{J}$ .  $\square$

$$\begin{array}{ll}
 \pi : \mathcal{TJ}^{\mathbf{m}} \longrightarrow \mathcal{TJ}^{\mathbf{m}} & \pi' : \mathcal{LJ}^{\mathbf{m}} \longrightarrow \mathcal{LJ}^{\mathbf{m}} \\
 \pi(x) = x & \pi'(\Box) = \Box \\
 \pi(\lambda x.t) = \lambda x.\pi(t) & \pi'(u::l) = \pi(u)::\pi'(l) \\
 \pi(t(u, l, (x)v)) = @(\pi(t), \pi(u), \pi'(l), x, \pi(v)) & \\
 \\
 \text{where} & \\
 @ (t, u, l, x, v) = \begin{cases} t'(u', l', (y)@(v', u, l, x, v)), & \text{if } t = t'(u', l', (y)v') \\ t(u, l, (x)v), & \text{otherwise} \end{cases}
 \end{array}$$

 Fig. 7. The mapping  $\pi$ 

This strong normalisation result contains a strong cut-elimination result, since it says, in particular, that from a term representing a sequent calculus derivation there cannot be an infinite sequence of cut-elimination (i.e.  $\beta\pi$ ) steps.

**Confluence.** Firstly we recall from [Espírito Santo and Pinto 2004] that relations  $\rightarrow_{\pi}$ ,  $\rightarrow_{\beta}$  and  $\rightarrow_{\beta\pi}$  in  $\lambda\mathbf{J}^{\mathbf{m}}$  are confluent.

**THEOREM 4.21 CONFLUENCE.**  $\rightarrow_{\pi}$ ,  $\rightarrow_{\beta}$  and  $\rightarrow_{\beta\pi}$  are confluent.

**PROOF.** By extending the confluence proofs for  $\Lambda J$  presented in [Joachimski and Matthes 2000]. See [Espírito Santo and Pinto 2004] for details.  $\square$

Now we consider confluence in the presence of rule  $\mu$ . The proof relies on the properties of rule  $\mu$  proved above, but also in the fact that, in certain cases,  $\pi$ -reductions can be replaced by  $\pi'$ -reductions (Lemma 4.22(4)). To establish the latter we make use of the mapping  $\pi$  of Figure 7 which calculates the  $\pi$ -normal form of a  $\lambda\mathbf{J}^{\mathbf{m}}$ -term.

**LEMMA 4.22.** For all  $t, u \in \mathcal{TJ}^{\mathbf{m}}$ ,

- (1)  $\pi(t)$  is  $\pi$ -normal.
- (2)  $t \rightarrow_{\pi'}^* \pi(t)$ .
- (3) if  $t \rightarrow_{\beta\pi} u$ , then  $\pi(t) \rightarrow_{\beta\pi'}^* \pi(u)$ .
- (4) if  $t \rightarrow_{\pi}^* u$  (resp.  $t \rightarrow_{\beta\pi}^* u$ ) and  $u$  is a  $\pi$ -normal form, then  $t \rightarrow_{\pi'}^* u$  (resp.  $t \rightarrow_{\beta\pi'}^* u$ ).

**PROOF.** 1. Proved together with the corresponding statement for  $l \in \mathcal{LJ}^{\mathbf{m}}$  by simultaneous induction on  $t$  and  $l$ .

2. Because  $\rightarrow_{\pi'} \subseteq \rightarrow_{\pi}^{\dagger}$  and  $\rightarrow_{\pi}$  is terminating,  $\rightarrow_{\pi'}$  is also terminating. Let  $t'$  be a  $\pi'$ -normal form of  $t$ . Since  $t'$  is also a  $\pi$ -normal form,  $t \rightarrow_{\pi}^* t'$ . Now, by 1 and confluence of  $\rightarrow_{\pi}$ ,  $t' = \pi(t)$  and thus  $t \rightarrow_{\pi'}^* \pi(t)$ .

3. By induction on  $t \rightarrow_{\beta\pi} u$ . The base case for  $\pi$  is trivial since  $\rightarrow_{\pi}$  is invariant w.r.t. mapping  $\pi$ . The base case for  $\beta$  uses the fact that, for all  $t, u \in \mathcal{TJ}^{\mathbf{m}}$ ,

$$\mathbf{s}(\pi(t), x, \pi(u)) \rightarrow_{\pi}^* \pi(\mathbf{s}(t, x, u)).$$

which can in be proved by induction on  $u$  with the help of another fact:

$$\mathbf{s}(t, x, @(t_0, u_0, l_0, y, v_0)) \rightarrow_{\pi}^* @(\mathbf{s}(t, x, t_0), \mathbf{s}(t, x, u_0), \mathbf{s}'(t, x, l_0), y, \mathbf{s}(t, x, v_0)),$$

for all  $t, t_0, u_0, v_0 \in \mathcal{TJ}^{\mathbf{m}}$ ,  $l_0 \in \mathcal{LJ}^{\mathbf{m}}$ . This latter fact, in turn, can be proved by induction on  $t_0$ .

4. The result for  $\rightarrow_{\pi}^*$  follows from 2 and from the fact that  $\pi$ -normal forms are unique, so  $u = \pi(t)$ . For the other result, observe firstly that 3 implies that  $\pi(t) \rightarrow_{\beta\pi'}^* \pi(u)$ . Observe then that  $\pi$ -normality of  $u$  guarantees  $\pi(u) = u$  and 1 guarantees  $t \rightarrow_{\pi'}^* \pi(t)$ .  $\square$

**THEOREM 4.23 CONFLUENCE.**  $\rightarrow_{\beta\pi\mu}$ ,  $\rightarrow_{\pi\mu}$  and  $\rightarrow_{\beta\mu}$  are confluent.

**PROOF.** Let  $R = \pi$  (resp.  $R = \beta\pi$ ) and let  $R' = \pi'$  (resp.  $R' = \beta\pi'$ ). Suppose  $t \rightarrow_{R\mu} t_1$  and  $t \rightarrow_{R\mu} t_2$ . This fork is closed as shown in the diagram of Figure 8.  $\boxed{1}$  holds by postponement of  $\mu$  (Theorem 4.5),  $\boxed{2}$  holds by confluence (Theorem 4.21),  $\boxed{3}$  holds by Lemma 4.22(3) ( $t_6$  being the  $\pi$ -normal form of  $t_5$ ),  $\boxed{4}$  holds by commutation with  $\mu$  (Theorem 4.7) and  $\boxed{5}$  holds by confluence of  $\rightarrow_{\mu}$  (Theorem 4.2). The proof of the first two statements of this theorem concludes by observing that each  $\pi'$ -reduction step in the reduction sequences from  $t_1$  to  $t_7$  and from  $t_2$  to  $t_8$  may be replaced by a sequence of  $\pi$ -reduction steps.

As to the last statement of this theorem, let  $R = \beta$  and  $R' = R$  and consider the same diagram, except that  $t_6 = t_5$ . The proof is complete after observing that  $\boxed{1}$  to  $\boxed{5}$  hold for the same reasons, except that  $\boxed{3}$  now holds trivially.  $\square$

Other proofs of confluence and strong normalisation of relation  $\rightarrow_{\beta\pi\mu}$  in  $\lambda\mathbf{J}^{\mathbf{m}}$  are given in [Espírito Santo and Pinto 2004]. There the lifting of the results from  $\Lambda\mathbf{J}$  is achieved through the mapping  $\nu : \lambda\mathbf{J}^{\mathbf{m}} \rightarrow \lambda\mathbf{J}$  introduced in that paper. The main problem in using the original mapping  $\mathbf{qJ} : \lambda\mathbf{J}^{\mathbf{m}} \rightarrow \lambda\mathbf{J}$  (introduced in [Espírito Santo and Pinto 2003] under name  $\mathbf{q}^{\mathbf{m}}$ ) is that this mapping does not preserve  $\mu$ -reduction. This was an obstacle to the lifting not overcome at the time of the writing of [Espírito Santo and Pinto 2004], that we remove here through the deeper study of the properties of  $\mu$ .

#### 4.6 Interaction of reduction and permutative conversions

One of the novelties of  $\lambda\mathbf{J}^{\mathbf{m}}$  is the co-existence of reduction and permutation rules. If one wants to obtain, from a  $\lambda\mathbf{J}^{\mathbf{m}}$ -term  $t$ , not only a  $\lambda$ -term, but also one in  $\beta$ -normal form, then one has to apply to  $t$  both reduction and permutation rules. This reason alone justifies the interest in combining both kinds of transformations. Below we present a complete study of the confluence and normalisation properties of the combined systems having the  $\beta$ -normal  $\lambda$ -terms as normal forms. The proofs of these properties benefit greatly of the results established before in this section.

Firstly observe that we cannot freely combine reduction and permutation rules and, simultaneously, keep strong normalisation. Two direct illustrations of this are the following cycles<sup>19</sup>.

$$\begin{aligned}
t(u, u' :: l, (x)v) &\rightarrow_q t(u)(u', l, (x)v) \\
&= t(u, [], (y)y)(u', l, (x)v) \\
&\rightarrow_{\pi} t(u, [], (y)y(u', l, (x)v)) \\
&\rightarrow_{\mu} t(u, \mathbf{a}([], u', l), (x)v) \\
&= t(u, u' :: l, (x)v) .
\end{aligned} \tag{21}$$

<sup>19</sup>Notice how in (21) we move around the particular case of the triangle in Figure 2 where  $l = []$ .

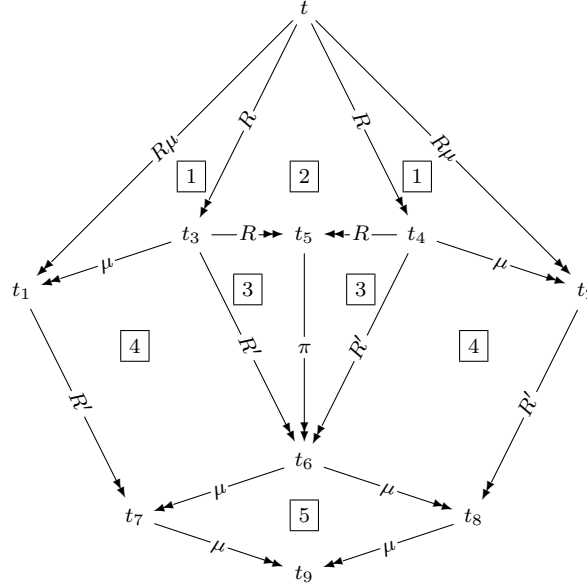


Fig. 8. Proof of Theorem 4.23

$$\begin{aligned}
 t(u, l, (x)v)(u', l', (y)v') &\rightarrow_{\pi} t(u, l, (x)v(u', l', (y)v')) \\
 &\rightarrow_{p_3} t(u, l, (x)v)(u', l', (y)v')
 \end{aligned} \tag{22}$$

In the latter example, we assume  $x \in v$ , whereas  $x \notin u', l', v'$  is guaranteed.

Calculation (22) shows  $\pi^{-1} \subseteq \rightarrow_p$ . So we omit  $\pi$  from the combinations of reduction rules with permutative conversions. Since we are interested in getting  $\lambda$ -terms, this is no handicap. In fact:

**PROPOSITION 4.24 COMPLETE COMBINATIONS.** *Let  $t$  be  $\lambda\mathbf{J}^m$ -term. Then the following are equivalent: (1)  $t$  is a  $\lambda$ -term in  $\beta$ -nf; (2)  $t$  is a  $\beta\mu pq$ -nf; (3)  $t$  is a  $\beta pq$ -nf; (4)  $t$  is a  $\beta_1 pq$ -nf.*

**PROOF.** A simple induction on  $t$  proves the first equivalence. The other two equivalences hold respectively because a  $\mu$ -redex is also a  $p$ -redex and a  $\beta_2$ -redex is also a  $q$ -redex.  $\square$

For all the combinations of rules capturing the class of  $\lambda$ -terms in  $\beta$ -normal form, confluence follows easily from previous results.

**THEOREM 4.25 CONFLUENCE.**  $\rightarrow_{\beta\mu pq}$ ,  $\rightarrow_{\beta pq}$ , and  $\rightarrow_{\beta_1 pq}$  are confluent.

**PROOF.** Let  $S$  be one of the combinations  $\beta\mu pq$ ,  $\beta pq$  or  $\beta_1 pq$ . Suppose  $t \rightarrow_S^* t_1$  and  $t \rightarrow_S^* t_2$ . This fork is closed as shown in the diagram in Figure 9, where  $\boxed{1}$  holds by Proposition 4.14 and the properties (1) and (2) in the proof of Theorem 4.1 and  $\boxed{2}$  holds by confluence of  $\rightarrow_{\beta}$  in the  $\lambda$ -calculus.  $\square$

Notice that  $\rightarrow_{\beta\mu pq}$ ,  $\rightarrow_{\beta pq}$ , and  $\rightarrow_{\beta_1 pq}$  all determine the same normal form, if any, for a given  $t \in \lambda\mathbf{J}^m$ , since  $\rightarrow_{\beta\mu pq} \supseteq \rightarrow_{\beta pq} \supseteq \rightarrow_{\beta_1 pq}$ . So, we are guaranteed to have

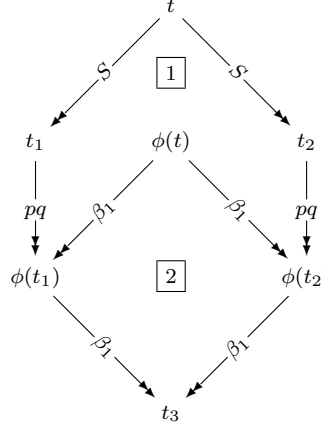


Fig. 9. Proof of Theorem 4.25

at most one  $\lambda$ -term in  $\beta$ -normal form corresponding to  $t$ . The weak normalisation theorem below shows that, for typable  $t$ , there is always one such  $\lambda$ -term, and two strategies to arrive at it. The theorem needs the following remark.

LEMMA 4.26. *If  $u \in \lambda\mathbf{J}^{\mathbf{m}}$  is a  $\beta\pi$ -normal form, then  $\phi(u)$  is a  $\beta$ -normal  $\lambda$ -term.*

PROOF. That the assignment of  $\lambda$ -terms to sequent calculus derivations relates normal terms with cut-free derivations is not surprising<sup>20</sup>. We give a new proof, where instead of the assignment  $\phi$  we analyse  $pq$ -reduction.

The proof needs a new subclass  $T$  of  $\lambda\mathbf{J}^{\mathbf{m}}$ -terms. We define the classes  $T$ ,  $A$ , and  $L$  simultaneously as follows:

$$\begin{aligned} (T) \quad t, u, v &::= x \mid \lambda x.t \mid a(u, l, (x)v) \\ (A) \quad a, b &::= x \mid a(u, l, (x)b) \\ (L) \quad l &::= t::l \mid \square \end{aligned}$$

$T$  is in between the classes of  $\beta\pi$ -normal terms and  $\beta$ -normal terms. The former (resp. the latter) is obtained by forbidding (resp. extending as  $a(u, l, (x)v)$ ) the second clause in the definition of  $A$ .

The point is that  $T$  is closed for  $pq$ -reduction (which is the case neither for  $\beta\pi$ -normal forms, nor for  $\beta$ -normal forms<sup>21</sup>). Specifically, one proves simultaneously, by induction on  $t$ ,  $a$ , and  $l$ : (a) if  $t \in T$  and  $t \rightarrow_{pq} t'$ , then  $t' \in T$ ; (b) if  $a \in A$  and  $a \rightarrow_{pq} a'$ , then  $a' \in A$ ; (c) if  $l \in L$  and  $l \rightarrow_{pq} l'$ , then  $l' \in L$ . Now since  $u$  is a  $\beta\pi$ -normal form,  $u \in T$ . From  $u \rightarrow_{pq}^* \phi(u)$ , and  $T$  closed for  $pq$ -reduction, it follows  $\phi(u) \in T$ . Hence  $\phi(u)$  is a  $\beta$ -normal form.  $\square$

<sup>20</sup>A direct proof is as follows. One proves:

- (1) For all  $u \in \lambda\mathbf{J}^{\mathbf{m}}$ , if  $u$  is a  $\beta\pi$ -normal form, then  $\phi(u)$  is a  $\beta$ -normal  $\lambda$ -term;
- (2) For all  $l \in \lambda\mathbf{J}^{\mathbf{m}}$ , for all  $n \geq 0$ , for all  $t_1, \dots, t_n, u, v$   $\beta$ -normal  $\lambda$ -terms, if  $l$  is a  $\beta\pi$ -normal form, then  $\phi'(x[t_1] \dots [t_n], u, l, y, v)$  is a  $\beta$ -normal  $\lambda$ -term;

by simultaneous induction on  $u$  and  $l$ .

<sup>21</sup>Obviously,  $p$ -reduction generates  $\pi$ -redexes. On the other hand, a  $\beta$ -normal form with the shape  $x(u, l, (y)\lambda z.v)(u', l', (y')v')$  immediately becomes a  $\beta$ -redex after one  $p_2$ -step.

**THEOREM 4.27 WEAK NORMALISATION.**  $\rightarrow_{\beta\mu pq}$  is weakly normalising on typable terms.

**PROOF.** Let  $t$  be a typable  $\lambda\mathbf{J}^m$ -term. The first strategy is rather obvious. From the property (2) in the proof of Theorem 4.1,  $t \rightarrow_{pq}^* \phi(t)$ . Now  $\phi(t)$  is a typable  $\lambda$ -term, so we reduce  $\phi(t)$  in  $\lambda$  to its  $\beta$ -normal form  $t'$ .

So the sought  $\beta$ -normal  $\lambda$ -term  $t'$  is obtained by reduction to  $pq$ -normal form, followed by normalisation. There is a second, “dual”, strategy that proceeds by cut-elimination first, followed by reduction to  $pq$ -normal form. It is not obvious that these two stages are enough. Let  $u$  be the  $\beta\pi$ -normal form of  $t$ , that exists since  $t$  is typable. We can build the following diagram:

$$\begin{array}{ccc}
t & \xrightarrow{pq} & \phi(t) \\
\beta\pi \downarrow & & \searrow \beta \\
u & \xrightarrow{pq} & \phi(u) \\
& & \searrow \beta \\
& & t'
\end{array}$$

The reduction from  $\phi(t)$  to  $\phi(u)$  is by Proposition 4.14. The triangle holds by confluence of  $\beta$ -reduction in the  $\lambda$ -calculus. That indeed  $\phi(u) = t'$  follows from Lemma 4.26.  $\square$

Computationally, whereas with the first strategy  $\beta$ -reduction happens only on  $\lambda$ -terms, the second strategy opens up alternatives for  $\beta$ -reduction, including the possibility of sharing of computations due to the explicit substitution facility offered by gm-application.

So, essentially, the meta-theory of  $\lambda\mathbf{J}^m$  developed so far was enough to prove the previous weak normalisation result. Now, *strong* normalisation for the combined relation  $\rightarrow_{\beta\mu pq}$  actually holds, but the proof needs a new tool: the  $\lambda\mathbf{ex}$ -calculus [Kesner 2009]. This is a calculus of explicit substitutions with variable names enjoying strong normalisation. Our proof of strong normalisation is based (as other strong normalisation proofs before) on a mapping that preserves infinite reduction sequences. Let us recall the  $\lambda\mathbf{ex}$ -calculus. Its terms are given by

$$M, N ::= x \mid \lambda x.M \mid MN \mid \langle N/x \rangle M .$$

We note that [Kesner 2009] uses  $M[x/N]$  for explicit substitution, instead of  $\langle N/x \rangle M$ . In  $\langle N/x \rangle M$  the occurrences of  $x$  in  $M$  are considered to be bound. We use  $\mathcal{T}\lambda\mathbf{ex}$  to denote the set of  $\lambda\mathbf{ex}$ -terms.

The typing of  $\lambda\mathbf{ex}$ -terms is as usual in simply-typed  $\lambda$ -calculus, with the following extra rule for typing explicit substitutions:

$$\frac{\Gamma \vdash M : A \quad \Gamma, x : A \vdash N : B}{\Gamma \vdash \langle M/x \rangle N : B} .$$

The calculus  $\lambda\mathbf{ex}$  is equipped with six reduction rules:

$$\begin{array}{ll}
\text{(B)} \quad (\lambda x.M)N \rightarrow \langle N/x \rangle M & \text{(Lamb)} \quad \langle N/x \rangle (\lambda y.M) \rightarrow \lambda y. \langle N/x \rangle M \\
\text{(Var)} \quad \langle N/x \rangle x \rightarrow N & \text{(App)} \quad \langle N/x \rangle (MM') \rightarrow (\langle N/x \rangle M) \langle N/x \rangle M' \\
\text{(Gc)} \quad \langle N/x \rangle M \rightarrow M, x \notin M & \text{(Comp)} \quad \langle N/x \rangle \langle N'/y \rangle M \rightarrow \langle \langle N/x \rangle N' / y \rangle \langle N/x \rangle M, x \in N'
\end{array}$$

$$\begin{array}{ll}
\circ : \mathcal{T}\mathbf{J}^{\mathbf{m}} \longrightarrow \mathcal{T}\lambda\mathbf{ex} & \circ : \mathcal{T}\lambda\mathbf{ex} \times \mathcal{L}\mathbf{J}^{\mathbf{m}} \longrightarrow \mathcal{T}\lambda\mathbf{ex} \\
x^\circ = x & (M, \square)^\circ = M \\
(\lambda x.t)^\circ = \lambda x.t^\circ & (M, u::l)^\circ = (Mu^\circ, l)^\circ \\
t(u, l, (x)v)^\circ = \begin{cases} (t^\circ u^\circ, l)^\circ & \text{if } v = x \\ \langle (t^\circ u^\circ, l)^\circ / x \rangle v^\circ & \text{if } v \neq x \end{cases} & 
\end{array}$$

Fig. 10. The mapping  $\circ$ 

The compatible closure of all rules but **B** is denoted by  $\rightarrow_x$  and the compatible closure of all the six rules by  $\rightarrow_{\mathbf{Bx}}$ . The calculus  $\lambda\mathbf{ex}$  is also equipped with the equation:

$$\langle N/x \rangle \langle N'/y \rangle M = \langle N'/y \rangle \langle N/x \rangle M, \quad x \notin N', y \notin N.$$

The relation  $=_e$  is the equivalence relation generated from the equation above and renaming of bound variables ( $\alpha$ -equivalence). The relations  $\rightarrow_{\mathbf{ex}}$  and  $\rightarrow_{\lambda\mathbf{ex}}$  result then from taking  $\rightarrow_x$  and  $\rightarrow_{\mathbf{Bx}}$  modulo  $=_e$ , i.e.:

$$\begin{array}{l}
M \rightarrow_{\mathbf{ex}} N \text{ iff exists } M', N' \in \mathcal{T}\lambda\mathbf{ex} \text{ s.t. } M =_e M' \rightarrow_x N' =_e N; \\
M \rightarrow_{\lambda\mathbf{ex}} N \text{ iff exists } M', N' \in \mathcal{T}\lambda\mathbf{ex} \text{ s.t. } M =_e M' \rightarrow_{\mathbf{Bx}} N' =_e N.
\end{array}$$

The relation  $\rightarrow_{\lambda\mathbf{ex}}$  is strongly normalising on typable terms. Another fundamental property of  $\lambda\mathbf{ex}$  in what follows is *full composition*:

$$\text{for all } M, N \in \mathcal{T}\lambda\mathbf{ex}, \langle M/x \rangle N \rightarrow_{\mathbf{ex}}^+ \mathbf{s}(M, x, N),$$

where  $\mathbf{s}(M, x, N)$  denotes implicit substitution in  $\lambda\mathbf{ex}$  (written  $N\{x/M\}$  in [Kesner 2009]), defined in the obvious way. We point to [Kesner 2009] for proofs of these two properties.

The mapping of  $\lambda\mathbf{J}^{\mathbf{m}}$ -terms into  $\lambda\mathbf{ex}$ -terms is denoted by  $\circ$  and given in Figure 10. This mapping preserves typing.

PROPOSITION 4.28 PRESERVATION OF TYPING BY  $\circ$ .

- (1) If  $\Gamma \vdash t : A$  in  $\lambda\mathbf{J}^{\mathbf{m}}$  then  $\Gamma \vdash t^\circ : A$  in  $\lambda\mathbf{ex}$ .
- (2) If  $\Gamma \vdash M : A$  in  $\lambda\mathbf{ex}$  and  $\Gamma; A \vdash l : B$  in  $\lambda\mathbf{J}^{\mathbf{m}}$  then  $\Gamma \vdash (M, l)^\circ : B$  in  $\lambda\mathbf{ex}$ .

PROOF. Simple simultaneous induction on  $t$  and  $l$ .  $\square$

Other fundamental properties of mapping  $\circ$  are presented in Proposition 4.30 and they are proved with the help of the collection of results in the following lemma.

LEMMA 4.29. For all  $t, u, v, l, l_0 \in \lambda\mathbf{J}^{\mathbf{m}}$  and for all  $M, N \in \lambda\mathbf{ex}$ :

- (1)  $x \in t$  iff  $x \in t^\circ$ ;  $x \in (M, l)^\circ$  iff  $x \in M$  or  $x \in l$ ;
- (2)  $\mathbf{s}(t^\circ, x, u^\circ) = \mathbf{s}(t, x, u)^\circ$ ;
- (3) if  $M \rightarrow_R N$  then  $(M, l)^\circ \rightarrow_R (N, l)^\circ$ , for  $R \in \{\mathbf{x}, \lambda\mathbf{ex}\}$ ;
- (4) if  $x \notin l$  then  $\langle M/x \rangle (N, l)^\circ \rightarrow_x^* \langle \langle M/x \rangle N, l \rangle^\circ$ ;
- (5)  $((M, l)^\circ u^\circ, l_0)^\circ = (M, \mathbf{a}(l, u, l_0))^\circ$ ;
- (6)  $\langle (t^\circ u^\circ, l)^\circ / x \rangle v^\circ \rightarrow_x^- t(u, l, (x)v)^\circ$ ;
- (7)  $\langle (t^\circ u^\circ, l)^\circ / x \rangle v^\circ \rightarrow_x^- t(u, l, (x)v)^\circ$ ;
- (8)  $\langle (t^\circ u^\circ, l)^\circ / x \rangle (M, l_0)^\circ \rightarrow_x^* (\langle (t^\circ u^\circ, l)^\circ / x \rangle M, t(u, l, (x)l_0))^\circ$ .



PROOF. The conjuncts of 1 are proved by simultaneous induction on  $t$  and  $l$ .

Part 2 is proved together with the property  $\mathbf{s}(t^\circ, x, (M, l)^\circ) = (\mathbf{s}(t^\circ, x, M), \mathbf{s}'(t, x, l))^\circ$ , for all  $t, l \in \lambda\mathbf{J}^{\mathbf{m}}$ ,  $M \in \lambda\mathbf{ex}$ , by simultaneous induction on  $u$  and  $l$ .

Parts 3, 4 and 5 are simple inductions on  $l$ .

Both part 6 and part 7 follow by case analysis on  $v$ .

Part 8 follows by induction on  $l_0$ , with the help of part 7.  $\square$

PROPOSITION 4.30 PRESERVATION OF  $\beta\mu\rho q$ -STEPS BY  $\circ$ . For all  $t, u \in \lambda\mathbf{J}^{\mathbf{m}}$ ,

- (1) if  $t \rightarrow_\beta u$  then  $t^\circ \rightarrow_{\lambda\mathbf{ex}}^+ u^\circ$ .
- (2) if  $t \rightarrow_{\mu\rho} u$  then  $t^\circ \rightarrow_{\mathbf{x}}^+ u^\circ$ .
- (3) if  $t \rightarrow_q u$  then  $t^\circ \rightarrow_{\mathbf{x}}^* u^\circ$ .

PROOF. Each of the statements is proved, by induction, together with the corresponding analogue for lists, for all  $l_1, l_2 \in \lambda\mathbf{J}^{\mathbf{m}}$ ,  $M \in \lambda\mathbf{ex}$ : (1) if  $l_1 \rightarrow_\beta l_2$  then  $(M, l_1)^\circ \rightarrow_{\lambda\mathbf{ex}}^+ (M, l_2)^\circ$ ; (2) if  $l_1 \rightarrow_{\mu\rho} l_2$  then  $(M, l_1)^\circ \rightarrow_{\mathbf{x}}^+ (M, l_2)^\circ$ ; (3) if  $l_1 \rightarrow_q l_2$  then  $(M, l_1)^\circ \rightarrow_{\mathbf{x}}^* (M, l_2)^\circ$ .

We illustrate the base cases. The base case for  $q$  gives  $t^\circ = u^\circ$ , so we illustrate also the inductive case where reduction steps not justified by IH can be generated.

Case  $\beta_1$ :  $(\lambda x.t)(u, [], (y)v) \rightarrow \mathbf{s}(\mathbf{s}(u, x, t), y, v)$ .

We assume  $v \neq y$ . The case  $v = y$  is analogous, but simpler.

$$\begin{aligned} (\lambda x.t)(u, [], (y)v)^\circ &= \langle (\lambda x.t^\circ)u^\circ / y \rangle v^\circ \\ &\rightarrow_{\mathbf{B}} \langle \langle u^\circ / x \rangle t^\circ / y \rangle v^\circ \\ &\rightarrow_{\mathbf{ex}}^+ \mathbf{s}(\mathbf{s}(u^\circ, x, t^\circ), y, v^\circ) \quad (\text{Full composition}) \\ &= \mathbf{s}(\mathbf{s}(u, x, t), y, v)^\circ \quad (\text{Lemma 4.29.2}) \end{aligned}$$

Case  $\beta_2$ :  $(\lambda x.t)(u, u_0 :: l, (y)v) \rightarrow \mathbf{s}(u, x, t)(u_0, l, (y)v)$ .

Assume  $v \neq y$  (the case  $v = y$  is analogous, but simpler).

$$\begin{aligned} (\lambda x.t)(u, u_0 :: l, (y)v)^\circ &= \langle \langle (\lambda x.t^\circ)u^\circ u_0^\circ, l \rangle / y \rangle v^\circ \\ &\rightarrow_{\lambda\mathbf{ex}} \langle \langle \langle u^\circ / x \rangle t^\circ \rangle u_0^\circ, l \rangle / y \rangle v^\circ \quad (\text{Lemma 4.29.3}) \\ &\rightarrow_{\mathbf{ex}}^+ \langle \langle \mathbf{s}(u^\circ, x, t^\circ)u_0^\circ, l \rangle / y \rangle v^\circ \quad (\text{Full composition and Lemma 4.29.3}) \\ &= \langle \langle \mathbf{s}(u, x, t)u_0^\circ, l \rangle / y \rangle v^\circ \quad (\text{Lemma 4.29.2}) \\ &= \mathbf{s}(u, x, t)(u_0, l, (y)v)^\circ \end{aligned}$$

Case  $\mu$ :  $t(u, l, (x)x(u_0, l_0, (y)v)) \rightarrow t(u, \mathbf{a}(l, u_0, l_0), (y)v)$  if  $x \notin u_0, l_0, v$ .

Again, we assume  $v \neq y$  (the case  $v = y$  being simpler).

$$\begin{aligned} &t(u, l, (x)x(u_0, l_0, (y)v))^\circ \\ = &\langle \langle t^\circ u^\circ, l \rangle / x \rangle \langle \langle (x u_0^\circ, l_0) \rangle / y \rangle v^\circ \\ \rightarrow_{\text{Comp}} &\langle \langle \langle t^\circ u^\circ, l \rangle / x \rangle (x u_0^\circ, l_0) \rangle / y \rangle \langle \langle t^\circ u^\circ, l \rangle / x \rangle v^\circ \\ &\quad (x \in (x u_0^\circ, l_0)^\circ, \text{ by } x \in x u_0^\circ \text{ and Lemma 4.29.1}) \\ \rightarrow_{\text{Gc}} &\langle \langle \langle t^\circ u^\circ, l \rangle / x \rangle (x u_0^\circ, l_0) \rangle / y \rangle v^\circ \quad (x \notin v^\circ, \text{ by } x \notin v \text{ and Lemma 4.29.1}) \\ \rightarrow_{\mathbf{x}}^* &\langle \langle \langle \langle t^\circ u^\circ, l \rangle / x \rangle (x u_0^\circ, l_0) \rangle / y \rangle v^\circ \quad (\text{by Lemma 4.29.4, as } x \notin l_0) \\ \rightarrow_{\mathbf{x}}^3 &\langle \langle \langle t^\circ u^\circ, l \rangle / x \rangle u_0^\circ, l_0 \rangle / y \rangle v^\circ \quad (x \notin u_0^\circ, \text{ by } x \notin u_0 \text{ and Lemma 4.29.1}) \\ = &\langle \langle t^\circ u^\circ, \mathbf{a}(l, u_0, l_0) \rangle / y \rangle v^\circ \quad (\text{Lemma 4.29.5}) \\ = &t(u, \mathbf{a}(l, u_0, l_0), (y)v)^\circ \end{aligned}$$

Case  $p_1$ :  $t(u, l, (x)y) \rightarrow y$ , if  $x \neq y$ .

$$t(u, l, (x)y)^\circ = \langle \langle t^\circ u^\circ, l \rangle / x \rangle y \rightarrow_{\text{Gc}} y = y^\circ$$

Case  $p_2$ :  $t(u, l, (x)\lambda y.v) \rightarrow \lambda y.t(u, l, (x)v)$ .

$$\begin{aligned} t(u, l, (x)\lambda y.v)^\circ &= \langle (t^\circ u^\circ, l)^\circ / x \rangle \lambda y.v^\circ \\ &\rightarrow_{\text{Lamb}} \lambda y. \langle (t^\circ u^\circ, l)^\circ / x \rangle v^\circ \\ &\rightarrow_x^* \lambda y.t(u, l, (x)v)^\circ \quad (\text{Lemma 4.29.7}) \\ &= (\lambda y.t(u, l, (x)v))^\circ \end{aligned}$$

Case  $p_3$ :  $t(u, l, (x)t_0(u_0, l_0, (y)v)) \rightarrow t(u, l, (x)t_0)(t(u, l, (x)u_0), t(u, l, (x)l_0), (y)v)$ , if  $x \notin v$ .

We do the case  $v \neq y$  (the case  $v = y$  is simpler). First we consider the subcase where  $x \notin t_0$ ,  $x \notin u_0$ , and  $x \notin l_0$ .<sup>22</sup>

$$\begin{aligned} &t(u, l, (x)t_0(u_0, l_0, (y)v))^\circ \\ &= \langle (t^\circ u^\circ, l)^\circ / x \rangle \langle (t_0^\circ u_0^\circ, l_0)^\circ / y \rangle v^\circ \\ &\rightarrow_{\text{Gc}} \langle (t_0^\circ u_0^\circ, l_0)^\circ / y \rangle v^\circ \quad (x \notin (t_0^\circ u_0^\circ, l_0)^\circ, x \notin v^\circ, \text{ by } x \notin t_0, u_0, l_0, v \text{ and Lemma 4.29.1}) \\ &= t_0(u_0, l_0, (y)v)^\circ \\ &= t(u, l, (x)t_0)(t(u, l, (x)u_0), t(u, l, (x)l_0), (y)v)^\circ \quad (x \notin t_0, u_0, l_0) \end{aligned}$$

Now we consider the subcase where  $x \in t_0$  or  $x \in u_0$  or  $x \in l_0$ .

$$\begin{aligned} &t(u, l, (x)t_0(u_0, l_0, (y)v))^\circ \\ &= \langle (t^\circ u^\circ, l)^\circ / x \rangle \langle (t_0^\circ u_0^\circ, l_0)^\circ / y \rangle v^\circ \\ &\rightarrow_{\text{Comp}} \langle (t^\circ u^\circ, l)^\circ / x \rangle (t_0^\circ u_0^\circ, l_0)^\circ / y \rangle \langle (t^\circ u^\circ, l)^\circ / x \rangle v^\circ \\ &\quad (x \in (t_0^\circ u_0^\circ, l_0)^\circ, \text{ by Lemma 4.29.1 and } x \in t_0, \text{ or } x \in u_0, \text{ or } x \in l_0) \\ &\rightarrow_{\text{Gc}} \langle (t^\circ u^\circ, l)^\circ / x \rangle (t_0^\circ u_0^\circ, l_0)^\circ / y \rangle v^\circ \quad (x \notin v^\circ, \text{ by } x \notin v \text{ and Lemma 4.29.1}) \\ &\rightarrow_x^* \langle (t^\circ u^\circ, l)^\circ / x \rangle (t_0^\circ u_0^\circ, l_0)^\circ / y \rangle v^\circ \quad (\text{Lemma 4.29.8}) \\ &\rightarrow_{\text{App}} \langle (t^\circ u^\circ, l)^\circ / x \rangle t_0^\circ \langle (t^\circ u^\circ, l)^\circ / x \rangle u_0^\circ, t(u, l, (x)l_0)^\circ / y \rangle v^\circ \\ &\rightarrow_x^* \langle t(u, l, (x)t_0)^\circ t(u, l, (x)u_0)^\circ, t(u, l, (x)l_0)^\circ / y \rangle v^\circ \quad (\text{Lemma 4.29.3/7}) \\ &= t(u, l, (x)t_0)(t(u, l, (x)u_0), t(u, l, (x)l_0), (y)v)^\circ \end{aligned}$$

Case  $q$ :  $t(u, u_0 :: l, (x)v) \rightarrow t(u, [], (y)y)(u_0, l, (x)v)$ .

Assume  $v \neq x$ . (It is analogous if  $v = x$ .)

$$\begin{aligned} t(u, u_0 :: l, (x)v)^\circ &= \langle (t^\circ u^\circ)u_0^\circ, l)^\circ / x \rangle v \\ &= \langle (t(u, [], (y)y)^\circ u_0^\circ, l)^\circ / x \rangle v \\ &= (t(u, [], (y)y)(u_0, l, (x)v))^\circ \end{aligned}$$

Notice that the second step would fail if, in the definition of mapping  $\circ$  for gm-applications, we uniformly generated an explicit substitution.

Let us consider the following inductive case:  $t_0(u_0, l_0, (x)v_0) \rightarrow_q t_0(u_0, l_0, (x)v_1)$ , with  $v_0 \rightarrow_q v_1$ .

$$\begin{aligned} t_0(u_0, l_0, (x)v_0)^\circ &= \langle (t_0^\circ u_0^\circ, l_0)^\circ / x \rangle v_0^\circ \quad (v_0 \neq x, \text{ as } v_0 \rightarrow_q v_1) \\ &\rightarrow_x^* \langle (t_0^\circ u_0^\circ, l_0)^\circ / x \rangle v_1^\circ \quad (\text{IH}) \\ &\rightarrow_x^* t_0(u_0, l_0, (x)v_1)^\circ \quad (\text{Lemma 4.29.6}) \end{aligned}$$

□

<sup>22</sup>It is this subcase that does not go through, if we consider permutation  $p'_3$  (see Subsection 2.4), and try to adapt the argument in order to obtain SN of  $\rightarrow_{\beta p'}$ . Note that if  $x \notin t_0, u_0, l_0, v$ ,  $\text{LHS}^\circ = \langle (t^\circ u^\circ, l)^\circ / x \rangle \langle (t_0^\circ u_0^\circ, l_0)^\circ / y \rangle v^\circ$  and now this term: (i) can be reduced to  $t_0(u_0, l_0, (y)v)^\circ$  in the same way, but, unlike for  $p_3$ , this expression is not reducible to  $\text{RHS}^\circ$  of  $p'_3$ ; (ii) cannot be reduced with  $\text{Comp}$ , as the side condition of this rule is not met.

**THEOREM 4.31 STRONG NORMALISATION.**  $\rightarrow_{\beta\mu\eta}$  is SN on typable terms.

**PROOF.** Follows from the previous proposition and the facts that  $\lambda\text{ex}$  is SN,  $\circ$  preserves typing (Proposition 4.30), and  $\rightarrow_q$  is SN.  $\square$

## 5. CONCLUSION

**Contributions.** From a strict proof-theoretical point of view, this work is a direct successor of [Dyckhoff and Pinto 1999; Schwichtenberg 1999], both for the aspects that remain constant (the kind of results sought w.r.t permutative conversions, the technical realisation via term annotations, the insistence on the multiary variant of the sequent calculus), and for novelties introduced (allowance and study of cuts and cut-elimination rules). By not restricting itself to the cut-free fragment, and by the inclusion of a thorough account of the connection with natural deduction (an involuntary development that proved to be necessary), this paper immediately becomes a revisit, for the implicational fragment of intuitionistic logic, and with totally new techniques, of Zucker and Pottinger classical results [Zucker 1974; Pottinger 1977]. Indeed, proof identity (i.e., equal interpretation into natural deduction) is characterized via a well-behaved set of permutation rules; and even normalisation (if by this we mean  $\beta$ -reduction in the  $\lambda$ -calculus) is proved to be an “homomorphic” image of cut-elimination.

An alternative way of seeing our contribution is as the development of the meta-theory of the extension  $\lambda\mathbf{J}^m$  of the  $\lambda$ -calculus, which proves to be a very useful task because  $\lambda\mathbf{J}^m$  is, not merely a device of proof annotations, but mainly a rich and meaningful computational interpretation of (our fragment of) sequent calculus. In this vein, we established all the main theoretical properties, of permutative conversions and reduction rules, both in isolation and in combined systems; and gave an account of the “internal structure” of the system, that is, of how several meaningful systems coexist inside  $\lambda\mathbf{J}^m$ , thereby making precise the connection with the  $\lambda$ -calculus itself. Through this development the richness of the system is made more precise, although enough evidence for that richness is the mere coexistence of permutative conversions and reduction rules, and among the latter the coexistence of an already complex  $\beta$ -rule with rules  $\pi$  and  $\mu$ .

**Related and future work.** In  $\lambda\mathbf{J}^m$  cuts have a restricted form (the right cut-formula is main in a left introduction), precisely that form that is closer to the elimination rule of natural deduction, and that corresponds to (an enlarged concept of) application. Therefore,  $\lambda\mathbf{J}^m$  is still a fragment of the full, multiary sequent calculus where all forms of cut are allowed. This is made precise in [Espírito Santo et al. 2007] by means of a “spectrum” of successively stronger proof systems (presented as typed  $\lambda$ -calculi), leading from the  $\lambda$ -calculus to the intuitionistic fragment of  $\bar{\lambda}\mu\tilde{\mu}$  [Curien and Herbelin 2000] — a type theoretic presentation of classical sequent calculus  $LK$  —, a spectrum where  $\lambda\mathbf{J}^m$  finds its place (under a slightly different clothing) approximately half way between the end-points. Essentially, systems stronger than  $\lambda\mathbf{J}^m$  in the spectrum are characterised by the fact that cut plays a double role, either as an application or as an explicit substitution, and, accordingly, the  $\beta$ -rule becomes a rule for the generation of substitution, whose actual

execution if left to another, complementary reduction rule.<sup>23</sup> So, what makes  $\lambda\mathbf{J}^m$  closer in spirit to the ordinary  $\lambda$ -calculus, and therefore what gives it an identity and character, is precisely the fact that only some forms of cut are primitive.<sup>24</sup>

Combining permutation and reduction rules is necessary if one wants to extract from a  $\lambda\mathbf{J}^m$ -term, not only a cut-free expression, but also a permutation-free one; or, not only a  $\lambda$ -term, but also a  $\beta$ -normal one. In this paper we developed a complete study of all the combinations of both kinds of rules capturing the class of  $\beta$ -normal  $\lambda$ -terms. Nonetheless, the richness of  $\lambda\mathbf{J}^m$  allows to consider other meaningful rules. [Espírito Santo et al. 2006] initiates a study of combined systems involving these new rules, which allows the identification inside  $\lambda\mathbf{J}^m$  of other interesting classes of normal forms for sequent calculus (including those of [Herbelin 1995] and of [Mints 1996]) and is computationally connected to the fact that multiple function application finds in  $\lambda\mathbf{J}^m$  a wealth of alternative representations. However, the complexity of [Espírito Santo et al. 2006] shows that the topic is beyond the scope of the present article, and in need of an expanded and future account.

Our presentation of  $\lambda\mathbf{J}^m$  as a system where the multiary  $\lambda$ -calculus and the  $\lambda$ -calculus with generalised application coexist, together with the fact that permutative conversions are separated into two kinds, each dedicated to the elimination of one of the features of multiarity and generality, may give the wrong impression that  $\lambda\mathbf{J}^m$  is a system erected from the  $\lambda$ -calculus by the modular addition of those two dimensions. But the  $\mu$ -reduction rule, and its replacing of certain uses of the generality facility by a heavier use of lists, shows that multiarity and generality are overlapping. The study of these “structural overlaps”, already started in [Espírito Santo and Pinto 2004] and matured in [Espírito Santo et al. 2006], led to the re-definition and re-classification of permutation and reduction rules, and to the refinement of the view of  $\lambda\mathbf{J}^m$  as a two-dimensional system. This topic also deserves an expanded and future treatment.

## ELECTRONIC APPENDIX

The electronic appendix for this article can be accessed in the ACM Digital Library by visiting the following URL: <http://www.acm.org/pubs/citations/journals/tocl/20YY-V-N/p1->.

## ACKNOWLEDGMENTS

The authors thank the anonymous referees, whose comments led to real improvement of the paper. Both authors are supported by FCT through the Centro de Matemática da Universidade do Minho.

## REFERENCES

- CERRITO, S. AND KESNER, D. 2004. Pattern matching as cut elimination. *Theoretical Computer Science* 323, 71–127.

<sup>23</sup>Maybe a deeper study of such calculi as calculi of explicit substitutions enjoying the property of full composition requires the treatment of some permutation rules as equations, in the style of [Kesner 2009].

<sup>24</sup>Incidentally, [Espírito Santo et al. 2007] introduces a new technique for proving strong normalisation, based on continuation passing-style translations into  $\lambda$ -calculus, and applies it to the sequent calculi in the spectrum, and thus to  $\lambda\mathbf{J}^m$  in particular.

- CURIEN, P.-L. AND HERBELIN, H. 2000. The duality of computation. In *Proc. of the 5th ACM SIGPLAN Int. Conf. on Functional Programming (ICFP '00)*. IEEE, 233–243.
- DYCKHOFF, R. AND PINTO, L. 1999. Permutability of inferences in intuitionistic sequent calculi. *Theoretical Computer Science* 212, 141–155.
- ESPÍRITO SANTO, J. 2002a. Conservative extensions of the  $\lambda$ -calculus for the computational interpretation of sequent calculus. Ph.D. thesis, University of Edinburgh.
- ESPÍRITO SANTO, J. 2002b. An isomorphism between a fragment of sequent calculus and an extension of natural deduction. In *Proc. of LPAR'02*, M. Baaz and A. Voronkov, Eds. LNAI, vol. 2514. Springer-Verlag, 354–366.
- ESPÍRITO SANTO, J., FRADE, M. J., AND PINTO, L. 2006. Structural proof theory as rewriting. In *Proc. of RTA'06*, F. Pfenning, Ed. LNCS, vol. 4098. Springer-Verlag, 197–211.
- ESPÍRITO SANTO, J., MATTHES, R., AND PINTO, L. 2007. Continuation-passing style and strong normalisation for intuitionistic sequent calculi. In *Proc. of TLCA'07*, S. Ronchi Della Rocca, Ed. LNCS, vol. 4583. Springer-Verlag, 286–300.
- ESPÍRITO SANTO, J. AND PINTO, L. 2003. Permutative conversions in intuitionistic multiary sequent calculus with cuts. In *Proc. of TLCA'03*, M. Hoffman, Ed. LNCS, vol. 2701. Springer-Verlag, 286–300.
- ESPÍRITO SANTO, J. AND PINTO, L. 2004. Confluence and strong normalisation of the generalised multiary  $\lambda$ -calculus. In *Revised selected papers from the Int. Workshop TYPES 2003*, S. Berardi, M. Coppo, and F. Damiani, Eds. LNCS, vol. 3085. Springer-Verlag, 194–209.
- GENTZEN, G. 1969. Investigations into logical deduction. In *The collected papers of Gerhard Gentzen*, M. Szabo, Ed. North-Holland, 68–131.
- HERBELIN, H. 1995. A  $\lambda$ -calculus structure isomorphic to a Gentzen-style sequent calculus structure. In *Proc. of CSL'94*, L. Pacholski and J. Tiuryn, Eds. Vol. 933. Springer-Verlag, 61–75.
- JOACHIMSKI, F. AND MATTHES, R. 2000. Standardization and confluence for a lambda-calculus with generalized applications. In *Proc. of RTA'00*, L. Bachmair, Ed. LNCS, vol. 1833. Springer-Verlag, 141–155.
- JOACHIMSKI, F. AND MATTHES, R. 2003. Short proofs of normalisation for the simply typed  $\lambda$ -calculus, permutative conversions and Gödel's T. *Archive for Mathematical Logic* 42, 59–87.
- KESNER, D. 2009. A theory of explicit substitutions with safe and full composition. *Logical Methods in Computer Science* 5(3:1), 1–29.
- MINTS, G. 1996. Normal forms for sequent derivations. In *Kreiseliana*, P. Odifreddi, Ed. A. K. Peters, Wellesley, Massachusetts, 469–492.
- POTTINGER, G. 1977. Normalization as a homomorphic image of cut-elimination. *Annals of Mathematical Logic* 12, 323–357.
- SCHWICHTENBERG, H. 1999. Termination of permutative conversions in intuitionistic Gentzen calculi. *Theoretical Computer Science* 212, 247–260.
- SØRENSEN, M. AND URZYCZYN, P. 2006. *Lectures on the Curry-Howard Isomorphism*. Studies in Logic and the Foundations of Mathematics, vol. 149. Elsevier.
- VON PLATO, J. 2001. Natural deduction with general elimination rules. *Annals of Mathematical Logic* 40, 541–567.
- ZUCKER, J. 1974. The correspondence between cut-elimination and normalization. *Annals of Mathematical Logic* 7, 1–112.

Received July 2009; revised July 2010; accepted July 2010

## APPENDIX

## A. PROOF TRANSFORMATIONS

In this appendix we show the proof transformations associated to each reduction rule and permutative conversion, as required by the proofs of Theorems 2.7 and 2.9 on subject reduction and subject permutation.

**Preliminaries.** In defining a proof transformation, we use the admissible rules proved in Propositions 2.4 and 2.5. This should be understood as indicating that the transformation also comprises the obtention of the conclusion of the admissible rule by the transformation of its premisses, according to the process that is associated to the admissibility of the rule, and that constitutes the content of the proof of that admissibility.

The content of the proof of Proposition 2.4, concerned with the admissibility of the forms (8) of mid-cut, is a certain transformation on derivations: given a derivation  $\mathcal{D}_1$  of  $\Gamma \vdash t : A$ , and given another derivation  $\mathcal{D}_2$  of  $x : A, \Gamma \vdash v : B$  (resp. of  $x : A, \Gamma; C \vdash l : B$ ), there is a derivation  $\mathcal{D}_3$  of  $\Gamma \vdash \mathbf{s}(t, x, v) : B$  (resp. of  $\Gamma; C \vdash \mathbf{s}'(t, x, l) : B$ ), obtained from  $\mathcal{D}_1$  and  $\mathcal{D}_2$  by an obvious (but tedious to define) process that we describe as the *complete permutation, along the subderivation of the right premiss*, of the implicit cut of the form of the left (resp. right) figure of (8) with premisses derived by  $\mathcal{D}_1$  and  $\mathcal{D}_2$ .

The content of the proof of Proposition 2.5, concerned with the admissibility of the form (9) of head-cut, is another transformation of derivations. Given derivations  $\mathcal{D}_1$  of  $\Gamma; C \vdash l : A_1 \supset A_2$  and  $\mathcal{D}_2$  of  $\Gamma \vdash u' : A_1$  and  $\mathcal{D}_3$  of  $\Gamma; A_2 \vdash l' : B$ , there is a derivation  $\mathcal{D}_4$  of  $\Gamma; C \vdash \mathbf{a}(l, u', l') : B$ , obtained from  $\mathcal{D}_1$ ,  $\mathcal{D}_2$  and  $\mathcal{D}_3$  by an obvious process that we describe as the *complete permutation, along the subderivation of the left premiss*, of the implicit cut of the form (10) with premisses derived by  $\mathcal{D}_1$ ,  $\mathcal{D}_2$  and  $\mathcal{D}_3$ .

For readability, in proof transformations we omit contexts as well as term and list annotations from sequents; we use terms and lists for naming derivations instead, and we choose these names exactly as in the definitions of reduction rules and permutative conversions (recall Definition 2.6 and Subsection 2.4). In addition, we do not make explicit the uses of the weakening and strengthening rules.

**Reduction rules.**

Case  $\beta_1$ . The LHS corresponds to a derivation of the form:

$$\frac{\frac{\frac{t}{\vdots} \quad x:A \vdash B}{\vdash A \supset B} \text{Right} \quad \frac{\frac{u}{\vdots} \quad \overline{A} \quad Ax \quad y:B \vdash C}{\vdash A \supset B \vdash C} \text{lm-Left}}{\vdash C} \text{head-cut}$$

In such cut inference the cut formula is main in both premisses. This determines a key step in cut-elimination and the elimination of this cut produces two mid-cuts:

$$\frac{\frac{\frac{u}{\vdots} \quad x:A \vdash B}{\vdash B} \text{mid-cut} \quad \frac{v}{\vdots} \quad y:B \vdash C}{\vdash C} \text{mid-cut}$$

Case  $\beta_2$ . The derivation corresponding to the LHS has the form

$$\frac{\frac{\frac{t}{\vdots} \quad \frac{u}{\vdots} \quad \frac{v}{\vdots} \quad \frac{l}{\vdots} \quad \frac{v'}{\vdots}}{\vdash A \supset B_1 \supset B_2} \text{ Right} \quad \frac{\frac{\frac{u}{\vdots} \quad \frac{v}{\vdots} \quad \frac{l}{\vdots}}{\vdash A \quad ; B_1 \supset B_2 \vdash C} \text{ Lft} \quad \frac{y:C \vdash D}{\vdash D} \text{ lm-Left}}{\vdash A \supset B_1 \supset B_2 \quad ; A \supset B_1 \supset B_2 \vdash D} \text{ head-cut}}{\vdash D} \text{ head-cut}$$

Again, we are in the presence of a key step in cut-elimination. However, in this case one of the cuts generated assumes the particular form corresponding to a gm-elimination:

$$\frac{\frac{\frac{u}{\vdots} \quad \frac{t}{\vdots}}{\vdash A \quad x:A \vdash B_1 \supset B_2} \text{ mid-cut} \quad \frac{\frac{v}{\vdots} \quad \frac{l}{\vdots} \quad \frac{v'}{\vdots}}{\vdash B_1 \quad ; B_2 \vdash C \quad y:C \vdash D} \text{ lm-Left}}{\vdash B_1 \supset B_2 \quad ; B_1 \supset B_2 \vdash D} \text{ head-cut}}{\vdash D}$$

Case  $\pi$ . The LHS corresponds to a derivation of the form

$$\frac{\frac{\frac{t}{\vdots} \quad \frac{u}{\vdots} \quad \frac{l}{\vdots} \quad \frac{v}{\vdots}}{\vdash A \supset B \quad ; A \supset B \vdash D_1 \supset D_2} \text{ (a)} \quad \frac{\frac{u'}{\vdots} \quad \frac{l'}{\vdots} \quad \frac{v'}{\vdots}}{\vdash D_1 \quad ; D_2 \vdash E \quad y:E \vdash F} \text{ (b)}}{\vdash D_1 \supset D_2 \quad ; D_1 \supset D_2 \vdash F} \text{ h-cut}}{\vdash F} \text{ h-cut}$$

where (a) and (b) are *lm-Left* inferences. Since the left cut formula of the outer cut is not main, this cut is left-permutable. The RHS of  $\pi$  results by permuting the outer cut above the other cut and inference (a):

$$\frac{\frac{\frac{t}{\vdots} \quad \frac{u}{\vdots} \quad \frac{l}{\vdots}}{\vdash A \supset B} \quad \frac{\frac{v}{\vdots} \quad \frac{u'}{\vdots} \quad \frac{l'}{\vdots} \quad \frac{v'}{\vdots}}{\vdash D_1 \supset D_2 \quad ; D_1 \supset D_2 \vdash F} \text{ h-cut}}{\vdash A \supset B \quad ; A \supset B \vdash F} \text{ (a)}}{\vdash F} \text{ h-cut}$$

Case  $\mu$ . The LHS corresponds to a derivation of the form

$$\frac{\frac{\frac{t}{\vdots} \quad \frac{u}{\vdots} \quad \frac{l}{\vdots}}{\vdash A \supset B} \quad \frac{\frac{u'}{\vdots} \quad \frac{l'}{\vdots} \quad \frac{v}{\vdots}}{\vdash C_1 \quad ; C_2 \vdash D \quad y:D \vdash E} \text{ m-Left}}{\vdash A \supset B \quad ; A \supset B \vdash E} \text{ lm-Left}}{\vdash E} \text{ head-cut}$$

Notice that the variable  $x$  does not occur in  $u', l', v$  and the inner gm-application is an *m-Left* introduction whose main formula is active in the next inference. So

formula  $C_1 \supset C_2$ , instead of being introduced by an *m-Left* inference, could have been introduced in a linear fashion by a *Lft* inference:

$$\frac{\frac{\frac{\frac{t}{\vdots} \vdash A \supset B}{\vdash A \supset B} \quad \frac{\frac{u}{\vdots} \vdash A \quad \frac{\frac{l}{\vdots} \vdash C_1 \supset C_2}{;B \vdash C_1 \supset C_2}}{;B \vdash D} \quad \frac{\frac{u' \quad l'}{\vdots \quad \vdots} \vdash C_1 \quad ;C_2 \vdash D}{;C_1 \supset C_2 \vdash D} \text{Lft}}{;A \supset B \vdash E} \text{aux-h-cut} \quad \frac{v}{\vdots} y:D \vdash E \text{lm-Left}}{\vdash E} \text{head-cut}$$

This construction requires subderivations indicated by  $u'$ ,  $l'$  and  $v$  to be strengthened by erasure of declaration  $x:C_1 \supset C_2$ .

**Permutative conversions.**

Case  $p_1$ . The LHS corresponds to a derivation of the form:

$$\frac{\frac{\frac{t}{\vdots} y:D \vdash A \supset B}{y:D \vdash A \supset B} \quad \frac{\frac{u}{\vdots} y:D \vdash A \quad \frac{l}{\vdots} y:D;B \vdash C}{y:D;A \supset B \vdash D} \quad \frac{x:C, y:D \vdash D}{x:C, y:D \vdash D} \text{Axiom}}{y:D \vdash D} \text{lm-Left} \quad \text{h-cut}$$

The endsequent can in this case simply be obtained by:

$$\frac{}{y:D \vdash D} \text{Axiom}$$

Case  $p_2$ . The LHS corresponds to a derivation of the form:

$$\frac{\frac{\frac{t}{\vdots} \vdash A \supset B}{\vdash A \supset B} \quad \frac{\frac{u}{\vdots} \vdash A \quad \frac{l}{\vdots} ;B \vdash C}{;A \supset B \vdash D_1 \supset D_2} \quad \frac{\frac{v}{\vdots} x:C, y:D_1 \vdash D_2}{x:C \vdash D_1 \supset D_2} \text{Right}}{\vdash D_1 \supset D_2} \text{lm-Left} \quad \text{h-cut}$$

If  $x \in v$ , the effect of the transformation is to permute the right inference down past the two inferences *lm-Left* and *h-cut*:

$$\frac{\frac{\frac{t}{\vdots} y:D_1 \vdash A \supset B}{y:D_1 \vdash A \supset B} \quad \frac{\frac{u}{\vdots} y:D_1 \vdash A \quad \frac{l}{\vdots} y:D_1;B \vdash C}{y:D_1;A \supset B \vdash D_2} \quad \frac{v}{\vdots} x:C, y:D_1 \vdash D_2 \text{lm-Left}}{\frac{y:D_1 \vdash D_2}{\vdash D_1 \supset D_2} \text{Right}} \text{h-cut}$$

If  $x \notin v$ , admissibility of strengthening justifies the following derivation:

$$\frac{v}{\vdots} \frac{y:D_1 \vdash D_2}{\vdash D_1 \supset D_2} \text{Right}$$



Consider now the special case of permutation  $p_2$  where  $x \in v$  and where  $t$  is a variable, say  $z$ . Its RHS corresponds to

$$\frac{\begin{array}{c} u \\ \vdots \\ z:A \supset B \vdash A \end{array} \quad \begin{array}{c} l \\ \vdots \\ z:A \supset B; B \vdash C \end{array} \quad \begin{array}{c} v \\ \vdots \\ y:D_1, x:C, z:A \supset B \vdash D_2 \\ x:C, z:A \supset B \vdash D_1 \supset D_2 \end{array}}{z:A \supset B \vdash D_1 \supset D_2} \begin{array}{l} \text{Right} \\ \text{m-Left} \end{array}$$

The transformation permutes the right inference below a multiary left inference:

$$\frac{\begin{array}{c} u \\ \vdots \\ y:D_1, z:A \supset B \vdash A \end{array} \quad \begin{array}{c} l \\ \vdots \\ y:D_1, z:A \supset B; B \vdash C \end{array} \quad \begin{array}{c} v \\ \vdots \\ x:C, y:D_1, z:A \supset B \vdash D_2 \end{array}}{\begin{array}{c} y:D_1, z:A \supset B \vdash D_2 \\ z:A \supset B \vdash D_1 \supset D_2 \end{array}} \begin{array}{l} \text{m-Left} \\ \text{Right} \end{array}$$

This transformation relates to permutative conversion (5) of [Schwichtenberg 1999] and, in the unary case ( $l = []$ ), to the permutation of an implication right inference below an ordinary implication left inference.

Case  $p_3$ . The LHS corresponds to a derivation of the form:

$$\frac{\begin{array}{c} t_1 \\ \vdots \\ \vdash A_1 \supset B_1 \end{array} \quad \begin{array}{c} u_1 \\ \vdots \\ \vdash A_1 \end{array} \quad \begin{array}{c} l_1 \\ \vdots \\ ; B_1 \vdash C_1 \end{array} \quad \begin{array}{c} \mathcal{D}_1 \\ x:C_1 \vdash D \end{array}}{\begin{array}{c} \vdash D \end{array}} \begin{array}{l} \text{lm-Left} \\ \text{h-cut} \end{array} \quad (23)$$

where  $\mathcal{D}_1$  is

$$\frac{\begin{array}{c} t_2 \\ \vdots \\ x:C_1 \vdash A_2 \supset B_2 \end{array} \quad \begin{array}{c} u_2 \\ \vdots \\ x:C_1 \vdash A_2 \end{array} \quad \begin{array}{c} l_2 \\ \vdots \\ x:C_1; B_2 \vdash C_2 \end{array} \quad \begin{array}{c} v \\ \vdots \\ y:C_2, x:C_1 \vdash D \end{array}}{\begin{array}{c} x:C_1 \vdash D \end{array}} \begin{array}{l} \text{lm-Left} \\ \text{h-cut} \end{array} \quad (24)$$

We illustrate what happens if  $x \in t_2$  and  $x \in u_2$ . (If this is not the case, we have simplified situations.) The corresponding RHS is

$$\frac{\begin{array}{c} \mathcal{D}_2 \\ \vdash A_2 \supset B_2 \end{array} \quad \begin{array}{c} \mathcal{D}_3 \\ \vdash A_2 \end{array} \quad \begin{array}{c} \mathcal{D}_4 \\ ; B_2 \vdash C_2 \end{array} \quad \begin{array}{c} v \\ \vdots \\ y:C_2 \vdash D \end{array}}{\begin{array}{c} \vdash D \end{array}} \begin{array}{l} \text{lm-Left} \\ \text{h-cut} \end{array} \quad (25)$$

where  $\mathcal{D}_2$  is

$$\frac{\frac{\begin{array}{c} t_1 \\ \vdots \\ \vdots \\ \vdots \end{array} \vdash A_1 \supset B_1 \quad \frac{\begin{array}{c} u_1 \\ \vdots \\ \vdots \\ \vdots \end{array} \vdash A_1 \quad \begin{array}{c} l_1 \\ \vdots \\ \vdots \\ \vdots \end{array} \vdash B_1 \vdash C_1 \quad \begin{array}{c} t_2 \\ \vdots \\ \vdots \\ \vdots \end{array} x:C_1 \vdash A_2 \supset B_2}{\vdash A_1 \supset B_1 \quad ; A_1 \supset B_1 \vdash A_2 \supset B_2} \text{ } lm\text{-Left}}{\vdash A_2 \supset B_2} \text{ } h\text{-cut}$$

the subderivation  $\mathcal{D}_3$  is the same as  $\mathcal{D}_2$ , except that  $t_2$  (resp.  $A_2 \supset B_2$ ) is replaced by  $u_2$  (resp.  $A_2$ ) and subderivation  $\mathcal{D}_4$  is obtained from derivations represented by  $t_1, u_1, l_1$  and  $l_2$  by an application of the following admissible rule

$$\frac{\Gamma \vdash t : A \supset B \quad \Gamma \vdash u : A \quad \Gamma ; B \vdash l : C \quad y : C, \Gamma ; E \vdash l' : D}{\Gamma ; E \vdash t(u, l, (y)l') : D} \quad (26)$$

(The admissibility of this rule is an easy induction on  $l'$ .)

The role of  $p_3$  is to permute the block of two inferences *lm-Left* and *h-cut* (24) down past the block of two inferences *lm-Left* and *h-cut* (23), swapping in particular the relative order of the head-cuts on  $A_1 \supset B_1$  and on  $A_2 \supset B_2$ . In order to perform the permutation, block (23) needs to be propagated to the derivations of the premises of the block (24) represented by  $t_2, u_2, l_2$  (recall  $x \notin v$ ). In the first two cases it suffices to add block (23) at the end of the derivations corresponding to  $t_2, u_2$ , replacing  $D$  by  $A_2 \supset B_2$  or  $A_2$  respectively. In the last case recall that the derivation corresponding to  $l_2$  is a tower of *Lft* inferences. Here propagation is achieved thus: for each of these *Lft* inferences, add the block (23) at the end of the derivation of its minor premise, whenever  $x$  occurs in the term representing this derivation; this is the proof transformation associated with the admissibility of rule (26).

Consider now the particular case of  $p_3$  where  $t_1$  and  $t_2$  are variables, say  $z_1$  and  $z_2$ . Its LHS corresponds to two multiary left inferences introducing  $z_1 : A_1 \supset B_1$  and  $z_2 : A_2 \supset B_2$ :

$$\frac{\frac{\begin{array}{c} u_1 \\ \vdots \\ \vdots \\ \vdots \end{array} \vdash A_1 \quad \begin{array}{c} l_1 \\ \vdots \\ \vdots \\ \vdots \end{array} \vdash B_1 \vdash C_1 \quad \frac{\begin{array}{c} u_2 \\ \vdots \\ \vdots \\ \vdots \end{array} \Gamma, x:C_1 \vdash A_2 \quad \begin{array}{c} l_2 \\ \vdots \\ \vdots \\ \vdots \end{array} \Gamma, x:C_1; B_2 \vdash C_2 \quad \begin{array}{c} v \\ \vdots \\ \vdots \\ \vdots \end{array} \Gamma, x:C_1, y:C_2 \vdash D}{\Gamma, x:C_1 \vdash D} \text{ } m\text{-Left}}{\Gamma \vdash D} \text{ } m\text{-Left}$$

where  $\Gamma = \{z_1 : A_1 \supset B_1, z_2 : A_2 \supset B_2\}$ . In this case, and assuming  $x \in u_2$ , the transformation on derivations associated with  $p_3$ , as defined before, can be thought of as the permutation of the two multiary left inferences, resulting in

$$\frac{\frac{\begin{array}{c} u_1 \\ \vdots \\ \vdots \\ \vdots \end{array} \vdash A_1 \quad \begin{array}{c} l_1 \\ \vdots \\ \vdots \\ \vdots \end{array} \vdash B_1 \vdash C_1 \quad \begin{array}{c} u_2 \\ \vdots \\ \vdots \\ \vdots \end{array} \Gamma, x:C_1 \vdash A_2}{\Gamma \vdash A_2} \text{ } m\text{-Left} \quad \frac{\mathcal{D}_4 \quad \begin{array}{c} v \\ \vdots \\ \vdots \\ \vdots \end{array} \Gamma, y:C_2 \vdash D}{\Gamma ; B_2 \vdash C_2 \quad \Gamma, y:C_2 \vdash D} \text{ } m\text{-Left}}{\Gamma \vdash D}$$

as long as derivation  $\mathcal{D}_2$  of (25) is replaced by an axiom (because declaration  $z_2 : A_2 \supset B_2$  is in the context) so that 25 is an *m-Left* inference.

Case  $q$ . The LHS corresponds to a derivation of the form

$$\frac{\frac{\frac{t \quad \vdots \quad \vdots}{\vdash A \supset (B_1 \supset B_2)} \quad \frac{\frac{u \quad \vdots \quad \vdots}{\vdash A} \quad \frac{\frac{v \quad l \quad \vdots \quad \vdots}{\vdash B_1 ; B_2 \vdash C} \quad \text{Lft} \quad \frac{v' \quad \vdots \quad \vdots}{x : C \vdash D} \quad \text{lm-Left}}{\vdash A \supset (B_1 \supset B_2) \vdash D} \quad \text{h-cut}}{\vdash D} \quad \text{h-cut}}$$

and the RHS corresponds to a derivation of the form

$$\frac{\frac{\frac{\mathcal{D} \quad \vdots \quad \vdots}{\vdash B_1 \supset B_2} \quad \frac{\frac{v \quad l \quad \vdots \quad \vdots}{\vdash B_1 ; B_2 \vdash C} \quad \text{lm-Left} \quad \frac{v' \quad \vdots \quad \vdots}{x : C \vdash D}}{\vdash B_1 \supset B_2 \vdash D} \quad \text{h-cut}}{\vdash D} \quad \text{h-cut},$$

$$\mathcal{D} = \frac{\frac{t \quad \vdots \quad \vdots}{\vdash A \supset (B_1 \supset B_2)} \quad \frac{\frac{u \quad \vdots \quad \vdots}{\vdash A} \quad \frac{\frac{\quad \quad \quad}{\vdash B_1 \supset B_2 \vdash B_1 \supset B_2} \quad \text{Ax} \quad \frac{\quad \quad \quad}{y : B_1 \supset B_2 \vdash B_1 \supset B_2} \quad \text{Axiom}}{\vdash A \supset (B_1 \supset B_2) \vdash B_1 \supset B_2} \quad \text{lm-Left}}{\vdash B_1 \supset B_2} \quad \text{h-cut}$$

In contrast to  $p_2$  and  $p_3$ , the proof transformation associated to  $(q)$  does not have the flavour of permuting inferences past other inferences. This transformation forces the main formula  $B_1 \supset B_2$  of a *Lft* inference to be the main formula of a linear multiary left inference. As a consequence,  $B_1 \supset B_2$  also becomes the cut-formula of a new head-cut, the right subderivation  $\mathcal{D}$  of which is another head-cut, the descendant of the original head-cut with cut-formula  $A \supset (B_1 \supset B_2)$ . The middle premise of the displayed *lm-Left* inference of  $\mathcal{D}$  has no instance of *Lft*.

## B. RELATIONSHIP WITH $\bar{\lambda}$ -CALCULUS

Herbelin's  $\bar{\lambda}$ -calculus (and its type system, the sequent calculus *LJT*) [Herbelin 1995] was important for the present paper because we recognised in  $\bar{\lambda}$  syntactic solutions for the implementation of multiarity.  $\bar{\lambda}$  continues to be used in literature as a presentation of the intuitionistic sequent calculus equipped with a computational interpretation, so a natural question is: what are the differences and relative advantages between  $\bar{\lambda}$  and  $\lambda\mathbf{J}^{\mathbf{m}}$ ?

Roughly speaking, the difference between the two systems boils down to this:  $\bar{\lambda}$  has a restricted form of (primitive) left introduction, whereas  $\lambda\mathbf{J}^{\mathbf{m}}$  has a restricted form of (primitive) cut. In  $\bar{\lambda}$ , left introduction corresponds to  $u :: l$ , while more general forms are derivable with the help of cut; in  $\lambda\mathbf{J}^{\mathbf{m}}$ , in addition to  $u :: l$ , one has the particular forms of gm-application  $x(u, (y)v)$  and  $x(u, l, (y)v)$  that correspond to ordinary and multiary left introduction. On the other hand, in  $\lambda\mathbf{J}^{\mathbf{m}}$  gm-application corresponds to a cut with right cut-formula main in a left introduction (recall figure (6)), while the unconstrained cut is only admissible (as witnessed by the admissibility of the typing rule for substitution - Proposition 2.4); in  $\bar{\lambda}$  the

general form of cut is primitive, and that is why  $\bar{\lambda}$  has an explicit substitution construction.

We now give some elements of a more formal comparison between  $\bar{\lambda}$  and  $\lambda\mathbf{J}^{\mathbf{m}}$ .

Expressions of Herbelin's  $\bar{\lambda}$ -calculus are, as those of  $\lambda\mathbf{J}^{\mathbf{m}}$ , separated into terms and lists, and are given by:

$$\begin{aligned} t, u, v &::= y\hat{l} \mid \lambda x.t \mid tl \mid v\{x := t\} \\ l &::= [] \mid u :: l \end{aligned}$$

We are omitting two list constructors for simplicity. In addition to  $\lambda$ -abstraction, the term constructors are  $y\hat{l}$  (dereliction),  $tl$  (head-cut) and  $v\{x := t\}$  (mid-cut), with typing rules

$$\frac{y:A, \Gamma; A \vdash l : B}{y:A, \Gamma \vdash y\hat{l} : B} \text{Der}$$

$$\frac{\Gamma \vdash t : A \quad \Gamma; A \vdash l : B}{\Gamma \vdash tl : B} \text{h-cut} \quad \frac{\Gamma \vdash t : A \quad x:A, \Gamma \vdash v : B}{\Gamma \vdash v\{x := t\} : B} \text{m-cut}$$

In  $\bar{\lambda}$  there are reduction rules to eliminate both forms of cuts. Mid-cuts are explicit substitutions and the reduction rules to eliminate them correspond to steps in the execution of explicit substitution.

We consider the question of mapping  $\bar{\lambda}$  into  $\lambda\mathbf{J}^{\mathbf{m}}$ . Insofar lists are restricted to the forms  $[]$  and  $u :: l$ , the following suffices for a type preserving mapping (whose range is actually contained in  $\lambda^{\mathbf{m}}$ ):

$$\begin{aligned} y\hat{[]} &\rightsquigarrow y \\ y\hat{(u :: l)} &\rightsquigarrow y(u, l) \\ v\{x := t\} &\rightsquigarrow \mathbf{s}(t, x, v) \\ t\hat{[]} &\rightsquigarrow t \\ t\hat{(u :: l)} &\rightsquigarrow t(u, l) \end{aligned}$$

The cut-free fragment of  $\bar{\lambda}$  is, of course, the fragment where constructors  $tl$  and  $v\{x := t\}$  are omitted. The interpretation of dereliction given by the first two clauses above establishes a bijection between cut-free  $\bar{\lambda}$ -terms and the normal forms of  $\lambda^{\mathbf{m}}$  as observed in [Espírito Santo 2002a]. The interpretation of mid-cut by meta-substitution means that the reduction steps for the explicit execution of substitution in  $\bar{\lambda}$  are mapped to equations in the target.

Let us now turn into the question of mapping  $\lambda\mathbf{J}^{\mathbf{m}}$  into  $\bar{\lambda}$ . There is an interpretation determined by the following transformations:

$$y \rightsquigarrow y\hat{[]} \tag{27}$$

$$t(u, l, (x)v) \rightsquigarrow v\{x := t(u :: l)\} \tag{28}$$

Thus gm-application is interpreted as explicit substitution. The same interpretation was given in [Schwichtenberg 1999] to  $v_x\{y, u :: l\}$ , i.e.  $y(u, l, (x)v)$ .

In terms of inference rules, (28) corresponds to interpreting gm-eliminations as the following combination of inferences in  $\bar{\lambda}$ :

$$\frac{\frac{\Gamma \vdash t : A \supset B \quad \frac{\Gamma \vdash u : A \quad \Gamma; B \vdash l : C}{\Gamma; A \supset B \vdash u :: l : C} \text{Lft}}{\Gamma \vdash t(u :: l) : C} \text{h-cut} \quad x : C, \Gamma \vdash v : D}{\Gamma \vdash v\{x := t(u :: l)\} : D} \text{m-cut} \quad (29)$$

This should be compared with (6): instead of a linear multiary left inference, one has a *Lft*-inference plus a mid-cut.

Interpretation (28) is related to the translation of *LJ* left inferences into inferences of  $\bar{\lambda}$  given in [Herbelin 1995]. Writing unary left inferences by means of  $\lambda\mathbf{J}^m$ -terms, Herbelin's translation reads

$$y(u, (x)v) \rightsquigarrow v\{x := \hat{y}[u]\} , \quad (30)$$

and is readily extended to multiary left inferences thus:

$$y(u, l, (x)v) \rightsquigarrow v\{x := \hat{y}(u :: l)\} . \quad (31)$$

In the case  $t = y$ , (31) is a slight improvement over (28) in the sense that

$$y(u, l, (x)v) \rightsquigarrow v\{x := \hat{y}[(u :: l)]\} \quad (32)$$

$$\rightarrow^* v\{x := \hat{y}(u :: l)\} , \quad (33)$$

where (32) is by (27) and (28), and (33) is a reduction allowed in  $\bar{\lambda}$ .

The interpretation (28) shows that  $\bar{\lambda}$ , or rather its type system, proves the same logical sequents as the type system of  $\lambda\mathbf{J}^m$ . But there are problems with interpretation (28) as a mapping between two structures, problems which are certainly shared to a large extent by Herbelin's interpretation (30) of *LJ*. The cut-free fragment of  $\bar{\lambda}$  is permutation-free [Dyckhoff and Pinto 1999] and, accordingly, is small and has no notion of permutative conversion. So it is not surprising that interpretation (28) does not preserve cut-freeness, and maps permutative conversion steps, at best, to cut-elimination steps (but the simulation of  $p_3$ -permutations poses problems, as it would require the propagation of a substitution inside another substitution, a feature not available in  $\bar{\lambda}$ ). Finally, also the simulation of cut-elimination steps is problematic. For instance, a  $\pi$ -step is mapped to

$$v'\{y := v\{x := t(u :: l)\}(u' :: l')\} \rightarrow v'\{y := v(u' :: l')\}\{x := t(u :: l)\} ,$$

a step for enlarging the scope of substitution  $\{x := t(u :: l)\}$  not reproducible by the cut-elimination rules of  $\bar{\lambda}$ .

### C. RESULTS ON PERMUTATIVE CONVERSIONS

In this appendix we present, in the first subsection, the proofs of the main results on permutative conversions, established in Section 4.1 in terms of mapping  $\phi$ ; and detail, in the second subsection, analogous results for mappings  $\mathbf{p}$ ,  $\mathbf{q}$ ,  $\mathbf{p}^m$  and  $\mathbf{q}\mathbf{J}$ .

## C.1 Proofs of main results

Firstly we introduce some basic facts about  $\phi$  used throughout.

LEMMA C.1. *For all  $t, u, v, v_1, v_2 \in \mathcal{TJ}^m$ ,  $l \in \mathcal{LJ}^m$ :*

- (1)  $\phi(t(u)) = \phi(t)(\phi(u))$ ;
- (2)  $\phi(t(u, l, (x)v)) = \mathbf{s}(\phi(t(u, l)), x, \phi(v)) = \phi(t(u, l, (x)v))$ ;
- (3)  $\phi(\mathbf{s}(t, x, v)) = \mathbf{s}(\phi(t), x, \phi(v))$  and  
 $\phi'(\mathbf{s}(\phi(t), x, \phi(v_1)), \mathbf{s}(\phi(t), x, \phi(v_2)), \mathbf{s}'(t, x, l), y, \mathbf{s}(\phi(t), x, \phi(v))) =$   
 $\mathbf{s}(\phi(t), x, \phi(v_1(v_2, l, (y)v)))$ .

PROOF. Part 1 is by a simple calculation:

$$\phi(t(u)) = \phi'(\phi(t), \phi(u), [], x, \phi(x)) = \mathbf{s}(\phi(t)(\phi(u)), x, x) = \phi(t)(\phi(u)).$$

The first equality of 2 follows by routine induction on  $l$  and the second equality follows then easily from the first. The two conjuncts of 3 are proved together, by simultaneous induction on  $v$  and  $l$ .  $\square$

Now we address two key properties in establishing the permutability theorem for mapping  $\phi$ . These properties assert that permutation reduction is invariant under  $\phi$  (Proposition C.2) and that each term can be reduced to its  $\phi$ -image using solely permutations (Proposition C.4).

PROPOSITION C.2. *If  $t \rightarrow_{pq}^* u$  then  $\phi(t) = \phi(u)$ , for all  $t, u \in \mathcal{TJ}^m$ .*

PROOF. The proof follows by induction on the relation  $\rightarrow_{pq}^*$ . Below we consider the base cases corresponding to the various permutations.

Case  $p_1$ .  $\phi(t(u, l, (x)y))$   
 $= \mathbf{s}(\phi(t(u, l)), x, \phi(y))$  (Lemma C.1.2)  
 $= \phi(y)$   $x \neq y$

Case  $p_2$ .  $\phi(t(u, l, (x)\lambda y.v))$   
 $= \mathbf{s}(\phi(t(u, l)), x, \phi(\lambda y.v))$  (Lemma C.1.2)  
 $= \lambda y.\mathbf{s}(\phi(t(u, l)), x, \phi(v))$   
 $= \lambda y.\phi(t(u, l, (x)v))$  (Lemma C.1.2)  
 $= \phi(\lambda y.t(u, l, (x)v))$

Case  $p_3$ .

$$\begin{aligned} & \phi(t_1(u_1, l_1, (x)t_2(u_2, l_2, (y)v))) \\ = & \mathbf{s}(\phi(t_1(u_1, l_1)), x, \phi(t_2(u_2, l_2, (y)v))) \quad (\text{Lemma C.1.2}) \\ = & \phi'(\mathbf{s}(\phi(t_1(u_1, l_1)), x, \phi(t_2)), \mathbf{s}(\phi(t_1(u_1, l_1)), x, \phi(u_2)), \mathbf{s}'(t_1(u_1, l_1), x, l_2), y, \phi(v)) \\ & (\text{Lemma C.1.3 and } x \notin v) \\ = & \phi'(\phi(t_1(u_1, l_1, (x)t_2)), \phi(t_1(u_1, l_1, (x)u_2)), t_1(u_1, l_1, (x)l_2), y, \phi(v)) \\ & (\text{Lemma C.1.2 and the fact} \\ & \phi'(t, u, \mathbf{s}'(t_1(u_1, l_1), x, l_2), y, v) = \phi'(t, u, t_1(u_1, l_1, (x)l_2), y, v) \\ & \text{proved by induction on } l_2) \\ = & \phi(t_1(u_1, l_1, (x)t_2)(t_1(u_1, l_1, (x)u_2), t_1(u_1, l_1, (x)l_2), (y)v)) \quad (\text{Lemma C.1.2}) \end{aligned}$$

Case  $q$ .  $\phi(t(u, v::l, (x)v'))$   
 $= \phi'(\phi(t)(\phi(u)), \phi(v), l, x, \phi(v'))$   
 $= \phi(t(u)(v, l, (x)v'))$  (Lemma C.1.1)

□

Proposition C.4 is established with the help of the following auxiliary results.

LEMMA C.3. *For all  $t, u, v \in \mathcal{T}^{\mathbf{m}}, l \in \mathcal{L}^{\mathbf{m}}, t(u, l, (x)v) \rightarrow_p^* \mathbf{s}(t(u, l), x, v)$ .*

PROOF. Proved together with the fact, for all  $t, u \in \mathcal{T}^{\mathbf{m}}, l, l_0 \in \mathcal{L}^{\mathbf{m}}$ ,

$$t(u, l, (x)l_0) \rightarrow_p^* \mathbf{s}'(t(u, l), x, l_0),$$

by induction on  $v$  and  $l_0$ . We show the cases relative to  $v$ .

Observe that if  $x \notin v$ , the LHS and the RHS are both equal to  $v$ . Below we assume  $x \in v$ .

$$\text{Case } v = x. \quad t(u, l, (x)x) = t(u, l) = \mathbf{s}(t(u, l), x, x) .$$

$$\begin{aligned} \text{Case } v = \lambda y.v_0. \quad t(u, l, (x)\lambda y.v_0) &= t(u, l, (x)\lambda y.v_0) & (x \in v) \\ &\rightarrow_{p_2} \lambda y.t(u, l, (x)v_0) \\ &\rightarrow_p^* \lambda y.\mathbf{s}(t(u, l), x, v_0) & \text{(IH)} \\ &= \mathbf{s}(t(u, l), x, \lambda y.v_0) . \end{aligned}$$

$$\text{Case } v = t_1(u_1, l_1).$$

$$\begin{aligned} t(u, l, (x)t_1(u_1, l_1)) &= t(u, l, (x)t_1(u_1, l_1)) & (x \in v) \\ &\rightarrow_{p_3} t(u, l, (x)t_1)(t(u, l, (x)u_1), t(u, l, (x)l_1)) \\ &\rightarrow_p^* \mathbf{s}(t(u, l), x, t_1)(\mathbf{s}(t(u, l), x, u_1), \mathbf{s}'(t(u, l), x, l_1)) & \text{(IH)} \\ &= \mathbf{s}(t(u, l), x, t_1(u_1, l_1)) . \end{aligned}$$

□

PROPOSITION C.4.  $t \rightarrow_{pq}^* \phi(t)$  for all  $t \in \mathcal{TJ}^{\mathbf{m}}$ .

PROOF. This result is proved together with the fact

$$\phi(t)(\phi(u), l, (x)\phi(v)) \rightarrow_{pq}^* \mathbf{s}(\phi(t(u, l)), x, \phi(v)), \text{ for all } l \in \mathcal{LJ}^{\mathbf{m}} \text{ and } t, u, v \in \mathcal{TJ}^{\mathbf{m}},$$

by simultaneous induction on the structure of  $t$  and  $l$ . We show the cases relative to  $l$ , where direct use of permutations is required.

$$\begin{aligned} \text{Case } l = []. \quad &\phi(t)(\phi(u), [], (x)\phi(v)) \\ &= \phi(t)(\phi(u), (x)\phi(v)) \\ &\rightarrow_p^* \phi(t)(\phi(u), (x)\phi(v)) & \text{(Lemma 2.8.2)} \\ &\rightarrow_p^* \mathbf{s}(\phi(t)(\phi(u)), x, \phi(v)) & \text{(Lemma C.3)} \\ &= \mathbf{s}(\phi(t(u, [])), x, \phi(v)) & \text{(Lemma C.1.1)} \end{aligned}$$

$$\begin{aligned} \text{Case } l = u_1 :: l_1. \quad &\phi(t)(\phi(u), u_1 :: l_1, (x)\phi(v)) \\ &\rightarrow_q \phi(t)(\phi(u))(u_1, l_1, (x)\phi(v)) \\ &\rightarrow_{pq}^* \phi(t(u))(u_1, l_1, (x)\phi(v)) & \text{(Lemma C.1.1 and IH)} \\ &\rightarrow_{pq}^* \mathbf{s}(\phi(t(u)(u_1, l_1)), x, \phi(v)) & \text{(IH)} \\ &= \mathbf{s}(\phi(t(u, u_1 :: l_1)), x, \phi(v)) & \text{(Lemma C.1.1)} \end{aligned}$$

□

Now we prove the main theorems about permutations. This is done with the help of Propositions C.2 and C.4. We start with the relationship between  $\rightarrow_{pq}$  and the kernel of  $\phi$ .

THEOREM C.5 PERMUTABILITY.  $\phi(t_1) = \phi(t_2)$  iff  $t_1 \leftrightarrow_{pq}^* t_2$ , for all  $t_1, t_2 \in \mathcal{TJ}^{\mathbf{m}}$ .

PROOF. Proposition C.2 guarantees  $\phi(t_1) = \phi(t_2)$  whenever  $t_1 \leftrightarrow_{pq}^* t_2$ . As to the only if part, we use Proposition C.4, obtaining  $t_1 \rightarrow_{pq}^* \phi(t_1)$  and  $t_2 \rightarrow_{pq}^* \phi(t_2)$  and thus, as by hypothesis  $\phi(t_1) = \phi(t_2)$ ,  $t_1$  and  $t_2$  are inter-permutable.  $\square$

The  $pq$ -normal forms are the  $\lambda$ -terms.

**THEOREM C.6 CHARACTERISATION OF  $pq$ -NORMAL FORMS.** *For all  $t \in \mathcal{TJ}^m$ ,  $t$  is  $pq$ -normal iff  $t \in \mathcal{T}$ .*

PROOF. On the one hand,  $\lambda$ -terms have neither  $p$  or  $q$ -redexes and so they are  $pq$ -normal. Consider, on the other hand, that  $t$  is  $pq$ -normal. By Proposition C.4,  $t \rightarrow_{pq}^* \phi(t)$  and thus the normality of  $t$  implies  $t = \phi(t)$ . The proof concludes observing that the co-domain of  $\phi$  is  $\mathcal{T}$ .  $\square$

Now the two main properties of relation  $\rightarrow_{pq}$  are established.

**THEOREM C.7 CONFLUENCE.**  *$\rightarrow_{pq}$  is confluent.*

PROOF. Assuming  $t \rightarrow_{pq}^* t_1$  and  $t \rightarrow_{pq}^* t_2$ , by Proposition C.4 follows that  $t_1 \rightarrow_{pq}^* \phi(t_1)$  and  $t_2 \rightarrow_{pq}^* \phi(t_2)$ . Yet by Proposition C.4 we have  $t \rightarrow_{pq}^* \phi(t)$  and we can now use Proposition C.2 to conclude that  $\phi(t_1) = \phi(t) = \phi(t_2)$ .  $\square$

**THEOREM C.8 TERMINATION.**  *$\rightarrow_{pq}$  is terminating.*

PROOF. We introduce notions of weight  $\mathbf{w}(t)$  and  $\mathbf{w}(l)$ , for terms and lists of  $\lambda\mathbf{J}^m$ , as follows:

$$\begin{aligned} \mathbf{w}(x) &= 1 & \mathbf{w}(\[]) &= 0 \\ \mathbf{w}(\lambda x.t) &= 1 + \mathbf{w}(t) & \mathbf{w}(u::l) &= 2 + \mathbf{w}(u) + \mathbf{w}(l) \\ \mathbf{w}(t(u, l, (x)v)) &= \mathbf{w}(v)(\mathbf{w}(t) + \mathbf{w}(u) + \mathbf{w}(l) + 1) \end{aligned}$$

Note that  $\mathbf{w}(t(u, l, (x)v)) \geq \mathbf{w}(t(u, l, (x)v))$ . Each permutation can be shown to have a RHS of weight lower than its LHS and thus every sequence of permutations must be finite.

Case  $p_1$ :  $\mathbf{w}(t(u, l, (x)y)) = \mathbf{w}(t) + \mathbf{w}(u) + \mathbf{w}(l) + 1 > 1 = \mathbf{w}(y)$  .

Case  $p_2$ :  $\mathbf{w}(t(u, l, (x)\lambda y.v))$   
 $= (1 + \mathbf{w}(v))(\mathbf{w}(t) + \mathbf{w}(u) + \mathbf{w}(l) + 1)$   
 $> 1 + \mathbf{w}(v)(\mathbf{w}(t) + \mathbf{w}(u) + \mathbf{w}(l) + 1)$   
 $\geq \mathbf{w}(\lambda y.t(u, l, (x)v))$  .

Case  $p_3$ :

$$\begin{aligned} & \mathbf{w}(t_1(u_1, l_1, (x)t_2(u_2, l_2, (y)v))) \\ = & \mathbf{w}(v)( (\mathbf{w}(t_2) + \mathbf{w}(u_2))(\mathbf{w}(t_1) + \mathbf{w}(u_1) + \mathbf{w}(l_1) + 1) + \\ & (\mathbf{w}(l_2) + 1)(\mathbf{w}(t_1) + \mathbf{w}(u_1) + \mathbf{w}(l_1)) + \mathbf{w}(l_2) + 1) \\ > & \mathbf{w}(v)((\mathbf{w}(t_2) + \mathbf{w}(u_2))(\mathbf{w}(t_1) + \mathbf{w}(u_1) + \mathbf{w}(l_1) + 1) + \mathbf{w}(t_1(u_1, l_1, (x)l_2)) + 1) \\ \geq & \mathbf{w}(t_1(u_1, l_1, (x)t_2)(t_1(u_1, l_1, (x)u_2), t_1(u_1, l_1, (x)l_2), (y)v)) \end{aligned}$$

where the inequality step follows from the fact

$$\mathbf{w}(t_1(u_1, l_1, (x)l_2)) < (\mathbf{w}(l_2) + 1)(\mathbf{w}(t_1) + \mathbf{w}(u_1) + \mathbf{w}(l_1)) + \mathbf{w}(l_2),$$

which is proved by induction on  $l_2$ .



$$\begin{aligned}
\text{Case } q: \quad & \mathbf{w}(t(u, v :: l, (x)v')) \\
& = \mathbf{w}(v')(\mathbf{w}(t) + \mathbf{w}(u) + 2 + \mathbf{w}(v) + \mathbf{w}(l) + 1) \\
& > \mathbf{w}(v')(\mathbf{w}(t) + \mathbf{w}(u) + 1 + \mathbf{w}(v) + \mathbf{w}(l) + 1) \\
& = \mathbf{w}(t(u)(v, l, (x)v')) .
\end{aligned}$$

□

Since  $\rightarrow_{pq}$  is confluent and terminating, each  $\lambda\mathbf{J}^{\mathbf{m}}$ -term  $t$  has a unique normal form that we denote by  $\downarrow_{pq}(t)$ . The rewriting system  $\rightarrow_{pq}$  calculates  $\phi$ .

**THEOREM C.9 REPRESENTATION OF  $\phi$ .**  $\phi(t) = \downarrow_{pq}(t)$ , for all  $t \in \mathcal{T}\mathbf{J}^{\mathbf{m}}$ .

**PROOF.** By Proposition C.4,  $t \rightarrow_{pq}^* \phi(t)$  and  $\phi(t)$  is a normal form. □

## C.2 Other results

Propositions C.10, C.11, C.12 and C.13 below are the analogues to  $\mathbf{p}$ ,  $\mathbf{q}$ ,  $\mathbf{p}^{\mathbf{m}}$  and  $\mathbf{q}\mathbf{J}$  respectively of Propositions C.2 and C.4.

**PROPOSITION C.10.** For all  $t, u \in \mathcal{T}\mathbf{J}$ :

- (1) if  $t \rightarrow_p^* u$  then  $\mathbf{p}(t) = \mathbf{p}(u)$ ;
- (2)  $t \rightarrow_p^* \mathbf{p}(t)$ .

**PROOF.** From Proposition C.2, if  $t \rightarrow_p^* u$ ,  $\phi(t) = \phi(u)$ . The proof of 1 concludes observing that  $\mathbf{p}$  is the restriction of  $\phi$  to  $\lambda\mathbf{J}$ -terms.

In order to prove 2, observe first that  $t \rightarrow_{pq}^* \phi(t) = \mathbf{p}(t)$  by Proposition C.4. The proof concludes observing that  $\lambda\mathbf{J}$ -terms have no  $q$ -redexes and  $\lambda\mathbf{J}$  is closed for  $p$ -permutations □

**PROPOSITION C.11.** For all  $t, u \in \mathcal{T}^{\mathbf{m}}$ :

- (1) if  $t \rightarrow_q^* u$  then  $\mathbf{q}(t) = \mathbf{q}(u)$ ;
- (2)  $t \rightarrow_q^* \mathbf{q}(t)$ .

**PROOF.** Analogous to the proof of the proposition above. □

**PROPOSITION C.12.** For all  $t, u \in \mathcal{T}\mathbf{J}^{\mathbf{m}}$ :

- (1) if  $t \rightarrow_p^* u$  then  $\mathbf{p}^{\mathbf{m}}(t) = \mathbf{p}^{\mathbf{m}}(u)$ ;
- (2)  $t \rightarrow_p^* \mathbf{p}^{\mathbf{m}}(t)$ .

**PROOF.** As to 1, the cases  $p_1$  and  $p_2$  follow as the corresponding cases in the proof of Proposition C.2, simply by replacing  $\phi$  by  $\mathbf{p}^{\mathbf{m}}$  and with the help of the following analogue of Lemma C.1.2 for  $\mathbf{p}^{\mathbf{m}}$ ,

$$\mathbf{p}^{\mathbf{m}}(t(u, l, (x)v)) = \mathbf{s}(\mathbf{p}^{\mathbf{m}}(t(u, l)), x, \mathbf{p}^{\mathbf{m}}(v)) = \mathbf{p}^{\mathbf{m}}(t(u, l, (x)v)) , \quad (34)$$

which results by simple calculations. The case  $p_3$  uses additionally the fact

$$\mathbf{p}^{\mathbf{m}'}(t(u, l, (x)l_0)) = \mathbf{s}'(\mathbf{p}^{\mathbf{m}}(\mathbf{p}^{\mathbf{m}}(t)(\mathbf{p}^{\mathbf{m}}(u), \mathbf{p}^{\mathbf{m}}(l))), x, \mathbf{p}^{\mathbf{m}'}(l_0)) , \quad (35)$$

proved by induction on  $l_0$ . The case  $p_3$  is as follows:

$$\begin{aligned}
& \mathbf{p}^{\mathbf{m}}(t_1(u_1, l_1, (x)t_2(u_2, l_2, (y)v))) \\
&= \mathbf{s}(\mathbf{p}^{\mathbf{m}}(t_1)(\mathbf{p}^{\mathbf{m}}(u_1), \mathbf{p}^{\mathbf{m}'}(l_1)), x, \mathbf{p}^{\mathbf{m}}(t_2)(\mathbf{p}^{\mathbf{m}}(u_2), \mathbf{p}^{\mathbf{m}'}(l_2), (y)\mathbf{p}^{\mathbf{m}}(v))) \\
&= \mathbf{s}(\mathbf{p}^{\mathbf{m}}(t_1(u_1, l_1, (x)t_2))(\mathbf{p}^{\mathbf{m}}(t_1(u_1, l_1, (x)u_2)), \mathbf{p}^{\mathbf{m}'}(t_1(u_1, l_1, (x)l_2))), y, \mathbf{p}^{\mathbf{m}}(v)) \\
&\quad \text{(Substitution lemma, } x \notin v \text{ and fact (35))} \\
&= \mathbf{p}^{\mathbf{m}}(t_1(u_1, l_1, (x)t_2)(t_1(u_1, l_1, (x)u_2), t_1(u_1, l_1, (x)l_2), (y)v)) \quad \text{(Fact (34))}
\end{aligned}$$

Statement 2 of this proposition is proved together with property

$$l \rightarrow_p^* \mathbf{p}^{\mathbf{m}'}(l),$$

by simultaneous induction on  $t$  and  $l$ . We illustrate the case where  $t$  is a gm-application.

$$\begin{aligned}
t_1(u_1, l_1, (x)v_1) &\rightarrow_p^* \mathbf{p}^{\mathbf{m}}(t_1)(\mathbf{p}^{\mathbf{m}}(u_1), \mathbf{p}^{\mathbf{m}'}(l_1), (x)\mathbf{p}^{\mathbf{m}}(v_1)) \quad \text{(IH)} \\
&\rightarrow_p^* \mathbf{p}^{\mathbf{m}}(t_1)(\mathbf{p}^{\mathbf{m}}(u_1), \mathbf{p}^{\mathbf{m}'}(l_1), (x)\mathbf{p}^{\mathbf{m}}(v_1)) \quad \text{(Lemma 2.8.2)} \\
&\rightarrow_p^* \mathbf{s}(\mathbf{p}^{\mathbf{m}}(t_1)(\mathbf{p}^{\mathbf{m}}(u_1), \mathbf{p}^{\mathbf{m}'}(l_1)), x, \mathbf{p}^{\mathbf{m}}(v_1)) \quad \text{(Lemma C.3)} \\
&= \mathbf{p}^{\mathbf{m}}(t_1(u_1, l_1, (x)v_1)).
\end{aligned}$$

□

PROPOSITION C.13. For all  $t, u \in \mathcal{TJ}^{\mathbf{m}}$ :

- (1) if  $t \rightarrow_q^* u$  then  $\mathbf{qJ}(t) = \mathbf{qJ}(u)$ ;
- (2)  $t \rightarrow_q^* \mathbf{q}(t)$ .

PROOF. The proof of 1 is analogous to the case corresponding to permutation (q) in the proof of Proposition C.2. It requires the fact

$$\mathbf{qJ}(t(u)) = \mathbf{qJ}(t)(\mathbf{qJ}(u)), \text{ for all } t, u \in \mathcal{TJ}^{\mathbf{m}}.$$

Statement 2 is proved together with property

$$t(u, l, (x)v) \rightarrow_q^* \mathbf{qJ}'(t, u, l, x, v), \text{ for all } t, u, v \in \mathcal{TJ} \text{ and for all } l \in \mathcal{LJ}$$

by simultaneous induction on  $t$  and  $l$ ; the proof is analogous to the proof of Proposition C.4, but simpler. □

The sequence of theorems in Subsection 4.1 (Theorems C.6 to C.9) can now be analogously established for mappings  $\mathbf{p}$ ,  $\mathbf{q}$ ,  $\mathbf{p}^{\mathbf{m}}$  and  $\mathbf{qJ}$ , with the help of Propositions C.10, C.11, C.12 and C.13.

THEOREM C.14. For each of the following combinations of  $F$ ,  $P$ ,  $\mathcal{S}$  and  $\mathcal{S}'$

	$F$	$P$	$\mathcal{S}$	$\mathcal{S}'$
$\mathbf{p}^{\mathbf{m}}$	$p$	$\lambda\mathbf{J}^{\mathbf{m}}$	$\lambda^{\mathbf{m}}$	
$\mathbf{qJ}$	$q$	$\lambda\mathbf{J}^{\mathbf{m}}$	$\lambda\mathbf{J}$	
$\mathbf{p}$	$p$	$\lambda\mathbf{J}$	$\lambda$	
$\mathbf{q}$	$q$	$\lambda^{\mathbf{m}}$	$\lambda$	

the properties below hold.

- (1) For all  $t_1, t_2 \in \mathcal{S}$ ,  $F(t_1) = F(t_2)$  iff  $t_1 \leftrightarrow_P^* t_2$ .
- (2) For all  $t \in \mathcal{S}$ ,  $t$  is a  $P$ -normal form iff  $t \in \mathcal{S}'$ .
- (3)  $\rightarrow_P$  is confluent and terminating in  $\mathcal{S}$  and, for all  $t \in \mathcal{S}$ ,  $\downarrow_P(t) = F(t)$ .