BRICS RS-98-26 Quaglia & Walker: On Encoding $p\pi$ in $m\pi$

BRICS

Basic Research in Computer Science

On Encoding $p\pi$ in $m\pi$

Paola Quaglia
David Walker

BRICS Report Series **RS-98-26**

ISSN 0909-0878 **October 1998**

See back inner page for a list of recent BRICS Report Series publications.
Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark

Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

`http://www.brics.dk`
`ftp://ftp.brics.dk`
This document in subdirectory `RS/98/26/`

On encoding $p\pi$ in $m\pi$

Paola Quaglia* [1] David Walker [2]

[1] BRICS**, Aarhus University
[2] Oxford University Computing Laboratory, U.K.

Abstract

This paper is about the encoding of $p\pi$, the polyadic π-calculus, in $m\pi$, the monadic π-calculus. A type system for $m\pi$ processes is introduced that captures the interaction regime underlying the encoding of $p\pi$ processes respecting a sorting. A full-abstraction result is shown: two $p\pi$ processes are typed barbed congruent iff their $m\pi$ encodings are monadic-typed barbed congruent.

1 Introduction

The π-calculus is a model of computation in which one can naturally express systems where the inter-connection structure among the parts changes during evolution. Its basic entities are names. They may be thought of as names of communication links. Processes, terms expressing systems, use names to interact, and pass names to one another by mentioning them in interactions. Names received by a process may be used and mentioned by it in further interactions.

In $m\pi$, the monadic π-calculus [MPW92], an interaction between processes involves the transmission of a single name. In $p\pi$, the polyadic π-calculus [Mil92], a tuple of names is passed in an interaction. As shown in [MPW92] atomic communication of tuples of names is expressible in the monadic calculus. Using standard notation (a reader unfamiliar with π-calculus may care to refer to Section 2) the key clauses in an inductively-defined translation $[\![\cdot]\!]$ from polyadic processes to monadic processes are

$$[\![x(z_1 \dots z_n).\, P]\!] = x(w).\, w(z_1).\, \dots .\, w(z_n).\, [\![P]\!]$$

and

$$[\![\overline{x}\langle a_1 \dots a_n \rangle.\, Q]\!] = \nu w\, \overline{x}w.\, \overline{w}a_1.\, \dots .\, \overline{w}a_n.\, [\![Q]\!]$$

where in each case w is a fresh name, i.e. is not free in the translated process. The transmission of a tuple $\vec{a} = a_1 \dots a_n$ is expressed by

*Funded by the EU, under the Marie Curie TMR Programme.
**Basic Research in Computer Science, Centre of the Danish National Research Foundation.

$$\begin{array}{ll} & [\![x(z_1 \dots z_n).\, P \mid \overline{x}\langle a_1 \dots a_n\rangle.\, Q]\!] \\ = & [\![x(z_1 \dots z_n).\, P]\!] \mid [\![\overline{x}\langle a_1 \dots a_n\rangle.\, Q]\!] \\ \longrightarrow & \nu w\, (w(z_1).\, \dots .\, w(z_n).\, [\![P]\!] \mid \overline{w}a_1.\, \dots .\, \overline{w}a_n.\, [\![Q]\!]) \\ \longrightarrow^n & [\![P]\!]\{\vec{a}/\vec{z}\} \mid [\![Q]\!] \\ = & [\![P\{\vec{a}/\vec{z}\} \mid Q]\!] \end{array}$$

where $\vec{z} = z_1 \dots z_n$. Although communication of an n-tuple is represented using $n+1$ interactions, it can be considered atomic since

$$\nu w\, (w(z_1).\, \dots .\, w(z_n).\, [\![P]\!] \mid \overline{w}a_1.\, \dots .\, \overline{w}a_n.\, [\![Q]\!]) \approx [\![P]\!]\{\vec{a}/\vec{z}\} \mid [\![Q]\!]$$

where $\approx$ is (weak) barbed congruence. The first interaction creates a private link; the subsequent semantically-inert communications transfer the names $a_1 \dots a_n$ from sender to receiver via that link.

In the polyadic calculus there is a pressing need for some kind of typing discipline, as among the processes are terms – $w(v).\, v(yz).\, \mathbf{0} \mid \overline{w}\langle x\rangle.\, \overline{x}\langle abc\rangle.\, \mathbf{0}$ is an example – where the components disagree on the length of the tuple to be passed – in the second communication in the example. Even on well-typed processes, however, the translation $[\![\cdot]\!]$ is not fully abstract when barbed congruence is adopted as process equivalence. If P and Q are well-typed polyadic processes then the equivalence of $[\![P]\!]$ and $[\![Q]\!]$ implies that of P and Q, but the converse does not hold, the reason being, briefly, that there are monadic contexts that do not conform to the interaction regime that underlies the translation. A simple example is $P = \overline{x}y.\, \overline{x}y.\, \mathbf{0}$ and $Q = \overline{x}y.\, \mathbf{0} \mid \overline{x}y.\, \mathbf{0}$: the monadic processes $\mathcal{K}[[\![P]\!]]$ and $\mathcal{K}[[\![Q]\!]]$ are not barbed bisimilar where $\mathcal{K}$ is the monadic context $[\cdot] \mid x(z).\, x(w).\, a(v).\, \mathbf{0}$.

In this paper we introduce a typing system for monadic processes that captures the interaction regime underlying the translation, and use it to obtain a full-abstraction result. The following informal and incomplete account is filled out in the paper. Fix a set $\mathcal{S}$ of sorts and a polyadic sorting λ, a partial function from $\mathcal{S}$ to $\mathcal{S}^+$, essentially of the kind introduced in [Mil92]. Write $\Psi \vdash_\lambda P$ if P is well-typed under λ assuming its free names have the sorts recorded in Ψ, a finite partial function from names to $\mathcal{S}$-sorts. Write $P \approx_\lambda Q$ if P and Q are mutually well-typed and $\mathcal{C}[P] \overset{\cdot}{\approx} \mathcal{C}[Q]$ for every suitably well-typed context $\mathcal{C}$, where $\overset{\cdot}{\approx}$ is barbed bisimilarity. We construct a set $\mathcal{S}^m$ of monadic sorts and a monadic sorting λ^m expressed using a graph. We give a typing system for inferring judgments of the form $\Psi; \Delta; \Gamma \vdash^m_\lambda M$ where M is a monadic process and Δ, Γ are finite partial functions from names to $\mathcal{S}^m$-sorts. One property of the system is that $\Psi \vdash_\lambda P$ iff $\Psi; \emptyset; \emptyset \vdash^m_\lambda [\![P]\!]$. As a guide, in a judgment $\Psi'; \Delta; \Gamma \vdash^m_\lambda M$ appearing in an inference of a judgment $\Psi; \emptyset; \emptyset \vdash^m_\lambda [\![P]\!]$, the functions Δ and Γ will record the monadic sorts of the monadic names introduced in translating P that are free in M and that M may use immediately for receiving and for sending, respectively. The sort of name that M may receive or send via a monadic name w is determined by λ^m; in general, that sort changes from one use of w to the next. Using the typing system we define $\approx^m_\lambda$ on monadic processes by setting $M \approx^m_\lambda N$ if M and N are mutually well-typed in the monadic system and $\mathcal{K}[M] \overset{\cdot}{\approx} \mathcal{K}[N]$ for every suitably well-typed monadic context $\mathcal{K}$. The main theorem is that if P and Q are well-typed polyadic processes then

$$P \approx_\lambda Q \quad \text{iff} \quad [\![P]\!] \approx^m_\lambda [\![Q]\!].$$

Thus the monadic typing system captures the interaction regime that underlies the polyadic-monadic translation. The main theorem is not easy to prove. The proof, however, sheds light on an important class of monadic processes, and several of the techniques used in it may be useful in other situations.

There has been much work on typing for π-calculus processes; a sample of papers is [Hon93, Kob97, KPT96, LW95, Mil92, NS97, PS93, PS97, San97, Tur96, VH93]. The work to which that presented here is most closely related is [Yos96]. We will explain the relationship at the end of the paper. In Section 2 we recall necessary background material, in Section 3 we introduce the typing system for monadic processes, and in Section 4 we prove the main theorem.

2 Background

In this section we recall necessary definitions and notations. We refer to the papers cited in the Introduction for further explanation and detailed proofs.

We presuppose a countably-infinite set of names, ranged over by lower-case letters. We write $\vec{x}$ for a tuple $x_1 \ldots x_n$ of names.

The *prefixes* are given by

$$\pi \quad ::= \quad \overline{x}\vec{y} \quad | \quad x(\vec{z})$$

where $\vec{y}$ and $\vec{z}$ are nonempty and $\vec{z}$ is a tuple of distinct names.

The *processes* are given by

$$P \quad ::= \quad \mathbf{0} \quad | \quad \pi. P \quad | \quad P \mid P' \quad | \quad \nu z\, P \quad | \quad !P.$$

We write $\mathcal{P}$ for the set of processes; P, Q, R range over $\mathcal{P}$.

A *context* is an expression obtained from a process by replacing an occurrence of '$\mathbf{0}$' by the *hole* $[\cdot]$; $\mathcal{C}$ ranges over contexts. We write $\mathcal{C}[P]$ for the process obtained by replacing the occurrence of the hole in $\mathcal{C}$ by P.

We sometimes refer to processes and contexts collectively as *terms* and use T to range over terms. A term is *monadic* if for each subterm $\overline{x}\vec{y}.T$ or $x(\vec{y}).T$ of it, $|\vec{y}| = 1$. We write $\mathcal{M}$ for the set of monadic processes; M, N, K range over $\mathcal{M}$. Also, $\mathcal{K}, \mathcal{H}$ range over monadic contexts.

In $x(\vec{z}).P$ and in $\nu z\, P$ the displayed occurrences of $\vec{z}$ and z are *binding* with *scope* P. An occurrence of a name in a term is *free* if it is not within the scope of a binding occurrence of the name. We write $\mathrm{fn}(P)$ for the set of names that have a free occurrence in P, and $\mathrm{fn}(P, Q, \ldots)$ for $\mathrm{fn}(P) \cup \mathrm{fn}(Q) \cup \ldots$. We write also $\mathrm{bn}(P)$ for the set of names that have a binding occurrence in P.

A *substitution* is a function on names that is the identity except on a finite set. We use θ to range over substitutions, and write $x\theta$ for θ applied to x. The *support* of θ, $\mathsf{supp}(\theta)$, is $\{x \mid x\theta \neq x\}$, and the *cosupport* is $\mathsf{cosupp}(\theta) = \{x\theta \mid x \in \mathsf{supp}(\theta)\}$. If $\mathsf{supp}(\theta) = \{x_1, \ldots, x_n\}$ and $x_i\theta = y_i$ for each i, we write $\{y_1 \cdots y_n/x_1 \ldots x_n\}$ for θ. If X is a set of names we write $X\theta$ for $\{x\theta \mid x \in X\}$. We write $P\theta$ for the process obtained by replacing each free occurrence of each name x in P by $x\theta$, with change of bound names to avoid captures.

We adopt the following conventions. We identify processes that differ only by change of bound names. Further, when considering a collection of processes and substitutions we tacitly assume that the free names of the processes are different from their bound names, that no name has more than one binding occurrence, and that no bound name is in the support or cosupport of any of the substitutions. This convention is referred to as the non-homonymy condition.

Definition 1 *Structural congruence* is the smallest congruence, $\equiv$, on processes such that

1. $P_1 \mid (P_2 \mid P_3) \equiv (P_1 \mid P_2) \mid P_3$, $P_1 \mid P_2 \equiv P_2 \mid P_1$, $P \mid \mathbf{0} \equiv P$
2. $\nu z\, \nu w\, P \equiv \nu w\, \nu z\, P$, $\nu z\, \mathbf{0} \equiv \mathbf{0}$
3. $\nu z\, (P_1 \mid P_2) \equiv P_1 \mid \nu z\, P_2$ provided $z \notin \mathrm{fn}(P_1)$
4. $!P \equiv P \mid !P$.

An occurrence of one term in another is *unguarded* if it is not underneath a prefix.

Lemma 2 If $P \equiv Q$ then $P\theta \equiv Q\theta$ and $\mathrm{fn}(P) = \mathrm{fn}(Q)$ and for each unguarded $\pi.\, P'$ in P there is an unguarded $\pi.\, Q'$ in Q with $P' \equiv Q'$.

Proof: The assertions follow easily from the definitions. $\Box$

Definition 3 *Intraaction* is the smallest relation, $\longrightarrow$, on processes such that

1. $\overline{x}\vec{y}.\, P \mid x(\vec{z}).\, Q \longrightarrow P \mid Q\{\vec{y}/\vec{z}\}$ provided $|\,\vec{y}\,|=|\,\vec{z}\,|$
2. $P \longrightarrow P'$ implies $P \mid Q \longrightarrow P' \mid Q$
3. $P \longrightarrow P'$ implies $\nu z\, P \longrightarrow \nu z\, P'$
4. $P \equiv Q \longrightarrow Q' \equiv P'$ implies $P \longrightarrow P'$.

We write $\Longrightarrow$ for the reflexive and transitive closure of $\longrightarrow$.

Lemma 4 If $P \longrightarrow Q$ then $\mathrm{fn}(Q) \subseteq \mathrm{fn}(P)$.

Proof: The proof is by induction on the inference of $P \longrightarrow Q$. It uses Lemma 2. $\Box$

In writing terms we assume composition associates to the left. Any intraaction arises from two complementary unguarded prefixes not underneath replications:

Lemma 5 If $P \longrightarrow Q$ then $P \equiv P' = \nu\vec{w}\, (\overline{x}\vec{y}.\, P_1 \mid x(\vec{z}).\, P_2 \mid P_3)$ where $|\,\vec{y}\,|=|\,\vec{z}\,|$ and $Q \equiv Q' = \nu\vec{w}\, (P_1 \mid P_2\{\vec{y}/\vec{z}\} \mid P_3)$ and $P' \mapsto Q'$, i.e. $P' \longrightarrow Q'$ may be inferred without use of the structural rule (rule 4).

Proof: The proof is by induction on the inference of $P \longrightarrow Q$. $\Box$

If x is a name then $\overline{x}$ is a *co-name*. We use μ to range over names and co-names.

The *observability predicates*, $\downarrow_\mu$, are defined by: $P \downarrow_x$ if P has an unguarded subterm $x(\vec{z}).\, Q$ and $x \in \mathrm{fn}(P)$; and $P \downarrow_{\overline{x}}$ if P has an unguarded subterm $\overline{x}\vec{y}.\, Q$ and $x \in \mathrm{fn}(P)$. Further, $\Downarrow_\mu$ is $\Longrightarrow\downarrow_\mu$.

Lemma 6 The relations $\downarrow_\mu$ are closed under $\equiv$, and $P \downarrow_x$ iff $P \equiv \nu\vec{w}\,(x(\vec{z}).\,Q \mid Q')$ where $x \notin \vec{w}$, and $P \downarrow_{\overline{x}}$ iff $P \equiv \nu\vec{w}\,(\overline{x}\vec{y}.\,Q \mid Q')$ where $x \notin \vec{w}$.

Proof: The assertions follow easily from the definitions and Lemma 2. $\square$

Definition 7 *Barbed bisimilarity* is the largest symmetric relation, $\dot{\approx}$, such that if $P \dot{\approx} Q$ then $P \downarrow_\mu$ implies $Q \Downarrow_\mu$, and $P \longrightarrow P'$ implies $Q \Longrightarrow\ \dot{\approx}\ P'$.

Definition 8 Monadic processes M and N are *monadic barbed congruent*, $M \approx^m N$, if $\mathcal{K}[M] \dot{\approx} \mathcal{K}[N]$ for every monadic context $\mathcal{K}$.

A simple but important observation is

Lemma 9 If $w \notin \mathrm{fn}(M, N, \vec{y})$ and $\vec{y} = y_1 \ldots y_n$ and $\vec{z} = z_1 \ldots z_n$, then

$$\nu w\,(\overline{w}y_1.\ \ldots .\,\overline{w}y_n.\,M \mid w(z_1).\ \ldots .\,w(z_n).\,N) \approx^m M \mid N\{\vec{y}/\vec{z}\}.$$

Proof: The simplest way to prove this is to appeal to the account of process behaviour given by the labelled transition relations (which is not given in this paper). The two processes in question are clearly congruent and hence monadic barbed congruent.

Alternatively, consider the relation containing all pairs of processes of the forms

$$\mathcal{K}[K_1, \ldots, K_m] \quad \text{and} \quad \mathcal{K}[K'_1, \ldots, K'_m]$$

where $\mathcal{K}$ is an m-ary monadic context for some $m \geq 0$, and for each i the process K_i is of the form

$$\nu w\,(\overline{w}y_1.\ \ldots .\,\overline{w}y_n.\,M \mid w(z_1).\ \ldots .\,w(z_n).\,N)$$

and the process K'_i of the form

$$M \mid N\{\vec{y}/\vec{z}\}$$

where K_i and K'_i satisfy the conditions of the lemma. It can be shown that this relation is a barbed bisimulation up to $\equiv$. $\square$

Definition 10 The translation $[\![\cdot]\!]$ from terms to monadic terms is defined as follows:

$$\begin{aligned} [\![\overline{x}\langle y_1 \ldots y_n\rangle.\,T]\!] &= \nu w\,\overline{x}w.\,\overline{w}y_1.\ \ldots .\,\overline{w}y_n.\,[\![T]\!] \\ [\![x(z_1 \ldots z_n).\,T]\!] &= x(w).\,w(z_1).\ \ldots .\,w(z_n).\,[\![T]\!] \end{aligned}$$

and $[\![[\cdot]]\!] = [\cdot]$, $[\![\mathbf{0}]\!] = \mathbf{0}$, $[\![T \mid T']\!] = [\![T]\!] \mid [\![T']\!]$, $[\![\nu z\,T]\!] = \nu z\,[\![T]\!]$, and $[\![\,!T]\!] = ![\![T]\!]$.

The translation enjoys the following properties.

Lemma 11 $\mathrm{fn}([\![P]\!]) = \mathrm{fn}(P)$ and $[\![P\theta]\!] = [\![P]\!]\theta$.

Proof: The proof is by induction on the structure of P. $\square$

Lemma 12 If $P \equiv Q$ then $[\![P]\!] \equiv [\![Q]\!]$.

Proof: The assertion follows directly from the definitions. □

Lemma 13 $P \downarrow_\mu$ iff $[\![P]\!] \downarrow_\mu$.

Proof: The proof is by induction on the structure of P. □

Lemma 14 If $P \longrightarrow Q$ then $[\![P]\!] \longrightarrow M \Longrightarrow [\![Q]\!]$ with $M \approx^m [\![Q]\!]$.

Proof: The proof uses Lemmas 5, 9, 11, and 12. □

We now consider typing of polyadic processes. Fix a set $\mathcal{S}$ of *sorts*, ranged over by s, t, and a *sorting* $\lambda : \mathcal{S} \rightharpoonup \mathcal{S}^+$.

We use Ψ to range over finite partial functions from names to sorts. We write $\mathrm{n}(\Psi)$ for the domain of Ψ. If $\mathrm{n}(\Psi) = \{x_1, \ldots, x_n\}$ and $\Psi(x_i) = s_i$ for each i, we write $\{x_1 : s_1, \ldots, x_n : s_n\}$ for Ψ. We write $\Psi(x) \simeq s$ to mean 'if $x \in \mathrm{n}(\Psi)$ then $\Psi(x) = s$'. Ψ and Ψ' are *compatible* if $x \in \mathrm{n}(\Psi) \cap \mathrm{n}(\Psi')$ implies $\Psi(x) = \Psi'(x)$. If Ψ and Ψ' are compatible we write Ψ, Ψ' for $\Psi \cup \Psi'$, and we abbreviate $\Psi, \{x : s\}$ to $\Psi, x : s$. We write $\Psi\theta$ for $\{x\theta : s \mid x : s \in \Psi\}$. We say θ *respects* Ψ if $x, y \in \mathrm{n}(\Psi)$ and $x\theta = y\theta$ implies $\Psi(x) = \Psi(y)$.

Definition 15 P is a λ-*process* if $\Psi \vdash P$ may be inferred for some Ψ using the rules in Table 1, where the side conditions are

1. $z \notin \mathrm{n}(\Psi)$,
2. $\lambda(s) = (t_1 \ldots t_n)$, $\Psi(x) \simeq s$, $\Psi(y_i) \simeq t_i$, $x = y_i$ implies $s = t_i$, and $y_i = y_j$ implies $t_i = t_j$,
3. $\lambda(s) = (t_1 \ldots t_n)$, $\Psi(x) \simeq s$, and $z_i \notin \mathrm{n}(\Psi)$.

$$\frac{}{\Psi \vdash \mathbf{0}} \qquad \frac{\Psi \vdash P}{\Psi \vdash !P} \qquad \frac{\Psi \vdash P_1 \quad \Psi \vdash P_2}{\Psi \vdash P_1 \mid P_2} \qquad \frac{\Psi, z : s \vdash P}{\Psi \vdash \nu z\, P} \ (1)$$

$$\frac{\Psi \vdash P}{\Psi, x : s, y_1 : t_1, \ldots, y_n : t_n \vdash \overline{x}\langle y_1 \ldots y_n \rangle. P} \ (2)$$

$$\frac{\Psi, z_1 : t_1, \ldots, z_n : t_n \vdash P}{\Psi, x : s \vdash x(z_1 \ldots z_n). P} \ (3)$$

Table 1: The polyadic typing rules

The type system enjoys the following properties.

Lemma 16 If $\Psi \vdash P$ then $\mathrm{fn}(P) \subseteq \mathrm{n}(\Psi)$ and $\mathrm{bn}(P) \cap \mathrm{n}(\Psi) = \emptyset$.

Proof: The proof is by induction on the inference of $\Psi \vdash P$. $\square$

In view of the last assertion, when we write a judgment $\Psi \vdash P$ we tacitly assume that the bound names of P are chosen not to be in $\mathrm{n}(\Psi)$.

Lemma 17 If $\Psi \vdash P$ and $\mathrm{n}(\Psi') \cap \mathrm{n}(\Psi) = \emptyset$, then $\Psi, \Psi' \vdash P$.

Proof: The proof is by induction on the inference of $\Psi \vdash P$. $\square$

Lemma 18 If $\Psi \vdash P$ and $\Psi' \subseteq \Psi$ and $\mathrm{fn}(P) \subseteq \mathrm{n}(\Psi')$, then $\Psi' \vdash P$.

Proof: The proof is by induction on the inference of $\Psi \vdash P$. $\square$

Lemma 19 If $\Psi \vdash P$ and $Q \equiv P$, then $\Psi \vdash Q$.

Proof: The assertion follows from the definitions using Lemmas 17 and 18. $\square$

Lemma 20 If $\Psi \vdash P$ and θ respects Ψ, then $\Psi\theta \vdash P\theta$.

Proof: The proof is by induction on the inference of $\Psi \vdash P$. $\square$

Lemma 21 If $\Psi \vdash P$ and $P \longrightarrow P'$, then $\Psi \vdash P'$.

Proof: The assertion follows from the definitions using Lemmas 19 and 20. $\square$

Lemma 22 If P is a λ-process and $P \Longrightarrow Q \equiv \nu\vec{w}(\overline{x}\vec{y}.\, Q_1 \mid x(\vec{z}).\, Q_2 \mid Q_3)$, then $|\vec{y}| = |\vec{z}|$.

Proof: The assertion follows from the definitions and Lemma 21. $\square$

We write $\Psi \vdash P, Q$ if $\Psi \vdash P$ and $\Psi \vdash Q$. The rules for typing contexts are like the rules for typing processes, with the addition of the rule: $\Psi \vdash [\cdot]$ for any Ψ. A context $\mathcal{C}$ is a $\lambda(\Psi)$-*context* if using the rule-instance $\Psi \vdash [\cdot]$ we can infer $\Psi' \vdash \mathcal{C}$ for some Ψ'. If $\Psi \vdash P$ and $\mathcal{C}$ is a $\lambda(\Psi)$-context, then $\mathcal{C}[P]$ is a λ-process.

Definition 23 P and Q are *barbed λ-congruent*, $P \approx_\lambda Q$, if there is Ψ such that $\Psi \vdash P, Q$ and $\mathcal{C}[P] \mathrel{\dot\approx} \mathcal{C}[Q]$ for every $\lambda(\Psi)$-context $\mathcal{C}$.

3 The monadic type system

We now introduce the monadic type system and establish some of its properties. Fix a set $\mathcal{S}$ of sorts and a sorting λ.

Definition 24 The set of *m-sorts*, ranged over by σ, τ, is

$$\mathcal{S}^m = \{\circ, \bullet\} \cup \{s^i \mid 1 \leq i \leq |\lambda(s)|, s \in \mathcal{S}\}.$$

For example, if $\mathcal{S} = \{s, t, r\}$ and $\lambda(s) = (t\, r)$, $\lambda(t) = (s)$ and $\lambda(r) = (r)$, then $\mathcal{S}^m = \{\circ, \bullet, s^1, s^2, t^1, r^1\}$.

Definition 25 The labelled directed graph $\mathcal{G}_\lambda$ has nodes $\mathcal{S}^m$, labels $\mathcal{S}$, and arrows

$$\circ \xrightarrow{s} s^1 \xrightarrow{t_1} s^2 \xrightarrow{t_2} \ldots \xrightarrow{t_{n-1}} s^n \xrightarrow{t_n} \bullet$$

if $\lambda(s) = (t_1 \ldots t_n)$.

For the example sorting above the arrows are

$$\circ \xrightarrow{s} s^1 \xrightarrow{t} s^2 \xrightarrow{r} \bullet \text{ and } \circ \xrightarrow{t} t^1 \xrightarrow{s} \bullet \text{ and } \circ \xrightarrow{r} r^1 \xrightarrow{r} \bullet.$$

We use Δ, Γ to range over finite partial functions from names to $\mathcal{S}^m - \{\bullet\}$, and use analogous notations to those mentioned earlier in connection with functions Ψ.

The following notion will be important in typing compositions.

Definition 26 Suppose Δ_1 and Γ_1 are compatible, and Δ_2 and Γ_2 are compatible. Then Δ_1, Γ_1 and Δ_2, Γ_2 are *complementary* if $\mathrm{n}(\Delta_1) \cap \mathrm{n}(\Delta_2) = \emptyset$, $\mathrm{n}(\Gamma_1) \cap \mathrm{n}(\Gamma_2) = \emptyset$, and Δ_1, Δ_2 and Γ_1, Γ_2 are compatible.

Definition 27 A monadic process M is a *λ^m-process* if $\Psi; \Delta; \Gamma \vdash M$ may be inferred for some Ψ, Δ, Γ using the rules in Table 2.

We write $\langle \Psi; \Delta; \Gamma \vdash M \rangle$ for an inference of the judgment $\Psi; \Delta; \Gamma \vdash M$. We say w is *monadic in* $\langle \Psi; \Delta; \Gamma \vdash M \rangle$ if for some judgment $\Psi'; \Delta'; \Gamma' \vdash M'$ in $\langle \Psi; \Delta; \Gamma \vdash M \rangle$ we have $w \in \mathrm{n}(\Delta', \Gamma')$.

In the premises of the rules for prefixes, $\{w : \bullet\}$ is read as $\emptyset$. This saves writing, as illustrated in the root of the left branch of the following sample inference, where $\emptyset; \emptyset; \emptyset \vdash \mathbf{0}$ stands for $\emptyset; \emptyset; \{w : \bullet\} \vdash \mathbf{0}$.

Note that in the rules for input prefix, $z \neq x$ by the convention on free and bound names.

The rules are best understood via an example. Assuming the example sorting above, let $P = !\nu a\, (Q \mid R)$ where $Q = \overline{a}\langle bc \rangle.\, \mathbf{0}$ and $R = a(uv).\, \mathbf{0}$. Then

$$[\![P]\!] = !\nu a\, ((\nu w\, \overline{a}w.\, \overline{w}b.\, \overline{w}c.\, \mathbf{0}) \mid a(z).\, z(u).\, z(v).\, \mathbf{0})$$

and we have, in full,

$$nil \qquad \frac{}{\Psi; \emptyset; \emptyset \vdash \mathbf{0}}$$

$$out_1 \qquad \frac{\Psi; \emptyset; \{y : \sigma\} \vdash M}{\Psi, x : s; \emptyset; \{y : \circ\} \vdash \overline{x}y.\, M} \quad \circ \xrightarrow{s} \sigma \text{ and } \Psi(x) \simeq s \text{ and } y \notin \mathrm{n}(\Psi, x)$$

$$out_2 \qquad \frac{\Psi; \emptyset; \{x : \tau\} \vdash M}{\Psi, y : s; \emptyset; \{x : \sigma\} \vdash \overline{x}y.\, M} \quad \circ \neq \sigma \xrightarrow{s} \tau \text{ and } \Psi(y) \simeq s \text{ and } x \notin \mathrm{n}(\Psi, y)$$

$$inp_1 \qquad \frac{\Psi; \{z : \sigma\}; \emptyset \vdash M}{\Psi, x : s; \emptyset; \emptyset \vdash x(z).\, M} \quad \circ \xrightarrow{s} \sigma \text{ and } \Psi(x) \simeq s \text{ and } z \notin \mathrm{n}(\Psi)$$

$$inp_2 \qquad \frac{\Psi, z : s; \{x : \tau\}; \emptyset \vdash M}{\Psi; \{x : \sigma\}; \emptyset \vdash x(z).\, M} \quad \circ \neq \sigma \xrightarrow{s} \tau \text{ and } x, z \notin \mathrm{n}(\Psi)$$

$$par \qquad \frac{\Psi; \Delta_1; \Gamma_1 \vdash M_1 \qquad \Psi; \Delta_2; \Gamma_2 \vdash M_2}{\Psi; \Delta_1, \Delta_2; \Gamma_1, \Gamma_2 \vdash M_1 \mid M_2} \quad \Delta_1, \Gamma_1 \text{ and } \Delta_2, \Gamma_2 \text{ are complementary}$$

$$res_1 \qquad \frac{\Psi, z : s; \Delta; \Gamma \vdash M}{\Psi; \Delta; \Gamma \vdash \nu z\, M} \quad z \notin \mathrm{n}(\Psi, \Delta, \Gamma) \qquad res_2 \qquad \frac{\Psi; \Delta; \Gamma, z : \circ \vdash M}{\Psi; \Delta; \Gamma \vdash \nu z\, M} \quad z \notin \mathrm{n}(\Psi, \Delta, \Gamma)$$

$$res_3 \qquad \frac{\Psi; \Delta, z : \sigma; \Gamma, z : \sigma \vdash M}{\Psi; \Delta; \Gamma \vdash \nu z\, M} \quad z \notin \mathrm{n}(\Psi, \Delta, \Gamma) \text{ and } \sigma \neq \circ \qquad rep \qquad \frac{\Psi; \emptyset; \emptyset \vdash M}{\Psi; \emptyset; \emptyset \vdash !M}$$

Table 2: The monadic typing rules

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\emptyset; \emptyset; \emptyset \vdash \mathbf{0}}}{\{c : r\}; \emptyset; \{w : s^2\} \vdash \overline{w}c.\, \mathbf{0}}}{\{b : t, c : r\}; \emptyset; \{w : s^1\} \vdash \overline{w}b.\, \overline{w}c.\, \mathbf{0}}}{\{a : s, b : t, c : r\}; \emptyset; \{w : \circ\} \vdash \overline{a}w.\, \overline{w}b.\, \overline{w}c.\, \mathbf{0}}}{\{a : s, b : t, c : r\}; \emptyset; \emptyset \vdash \llbracket Q \rrbracket} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\{u : t, v : r, b : t, c : r\}; \emptyset; \emptyset \vdash \mathbf{0}}}{\{u : t, b : t, c : r\}; \{z : s^2\}; \emptyset \vdash z(v).\, \mathbf{0}}}{\{b : t, c : r\}; \{z : s^1\}; \emptyset \vdash z(u).\, z(v).\, \mathbf{0}}}{\{a : s, b : t, c : r\}; \emptyset; \emptyset \vdash \llbracket R \rrbracket}}{\dfrac{\{a : s, b : t, c : r\}; \emptyset; \emptyset \vdash \llbracket Q \rrbracket \mid \llbracket R \rrbracket}{\dfrac{\{b : t, c : r\}; \emptyset; \emptyset \vdash \nu a\, (\llbracket Q \rrbracket \mid \llbracket R \rrbracket)}{\{b : t, c : r\}; \emptyset; \emptyset \vdash \llbracket P \rrbracket}}}$$

Note that in the judgment

$$\{a : s, b : t, c : r\}; \emptyset; \{w : \circ\} \vdash \overline{a}w.\, \overline{w}b.\, \overline{w}c.\, \mathbf{0}$$

the name w is recorded in the Γ-component with m-sort $\circ$, and that the second of the restriction rules is applied to infer

$$\{a : s, b : t, c : r\}; \emptyset; \emptyset \vdash [\![Q]\!].$$

In the judgment

$$\{b : t, c : r\}; \emptyset; \{w : s^1\} \vdash \overline{w}b.\, \overline{w}c.\, \mathbf{0}$$

w is ascribed m-sort s^1, indicating that the process can immediately send on w. The graph $\mathcal{G}_\lambda$ stipulates that the name which can be sent via w must be of sort t. In the judgment

$$\{c : r\}; \emptyset; \{w : s^2\} \vdash \overline{w}c.\, \mathbf{0}$$

w has m-sort s^2 and c sort r as given by $\mathcal{G}_\lambda$. After being used for sending for the second and last time, w disappears from the Γ-component.

In a complementary way, m-sorts are ascribed to z in the Δ-components of the judgments involving subterms of R to indicate how they use the name for receiving.

We now state some properties of the type system.

Lemma 28 Suppose $\Psi; \Delta; \Gamma \vdash M$. Then

1. Δ and Γ are compatible.
2. $\mathrm{n}(\Psi) \cap \mathrm{n}(\Delta, \Gamma) = \emptyset$.
3. $\mathrm{fn}(M) \subseteq \mathrm{n}(\Psi, \Delta, \Gamma)$.
4. $\mathrm{n}(\Delta, \Gamma) \subseteq \mathrm{fn}(M)$.
5. $\circ \notin \mathsf{cosupp}(\Delta)$.
6. $\Gamma(x) = \circ$ implies $x \notin \mathrm{n}(\Delta)$.
7. $\mathrm{bn}(M) \cap \mathrm{n}(\Psi, \Delta, \Gamma) = \emptyset$.

Proof: All the items are proved by induction on the derivation of $\Psi; \Delta; \Gamma \vdash M$. Each rule is considered in turn as the last rule applied. Only a few issues are worth commenting on.

- The proof of (2), relatively to the prefix cases, appeals to the side conditions of the corresponding rules.
- In the proof of (6), the case of the *par* rule is carried out as follows. Suppose that $\Psi; \Delta; \Gamma \vdash M_1 \mid M_2$ with $x : \circ \in \Gamma$. Assume that $x \in \mathrm{n}(\Delta)$. Then, from Δ and Γ compatible, it follows $x : \circ \in \Delta$. Absurd by (5).
- As for the proof of (7), by (4) and the non-homonymy condition it is sufficient to prove that $\mathrm{bn}(M) \cap \mathrm{n}(\Psi) = \emptyset$.

□

In view of the last part, when we write a judgment $\Psi; \Delta; \Gamma \vdash M$ we tacitly assume that the bound names of M are chosen not to be in $\mathrm{n}(\Psi, \Delta, \Gamma)$.

We write $|\,T\,|$ for the number of operators in the term T. The reason the following lemma takes the form it does is that to carry out later arguments by induction on type inference, a handle on the size of terms is needed.

Lemma 29 If $\Psi; \Delta; \Gamma \vdash M$ then there is $M' \equiv M$ such that $|\,M'\,| \leq |\,M\,|$ and:

1. If $w : \circ \in \Gamma$ then $M' = \nu\vec{w}\,\overline{x}w.\,N$ or $M' = \nu\vec{w}\,(\overline{x}w.\,N \mid K)$ where $w \notin \vec{w}$.
2. If $w : \sigma \in \Gamma$ and $\sigma \neq \circ$, then $M' = \nu\vec{w}\,\overline{w}x.\,N$ or $M' = \nu\vec{w}\,(\overline{w}x.\,N \mid K)$ where $w \notin \vec{w}$.
3. If $w : \sigma \in \Delta$ then $M' = \nu\vec{w}\,w(z).\,N$ or $M' = \nu\vec{w}\,(w(z).\,N \mid K)$ where $w \notin \vec{w}$.
4. If $w : \sigma \in \Delta \cap \Gamma$ then $M' = \nu\vec{w}(\overline{w}x.\,N \mid w(z).\,N')$ or $M' = \nu\vec{w}(\overline{w}x.\,N \mid w(z).\,N' \mid K)$ where $w \notin \vec{w}$.

Proof: The proof of the four items is by simulataneous induction on the derivation of $\Psi; \Delta; \Gamma \vdash M$. We show in the following the case when the last rule applied is the *par* rule. The other cases are easier.

[*par*] $M = M_1 \mid M_2$ and $\Psi; \Delta_j; \Gamma_j \vdash M_j$ with $\Delta_1 \cup \Delta_2 = \Delta$ and $\Gamma_1 \cup \Gamma_2 = \Gamma$ and Δ_1, Γ_1 and Δ_2, Γ_2 complementary.

1. As $\mathrm{n}(\Gamma_1) \cap \mathrm{n}(\Gamma_2) = \emptyset$, from $w : \circ \in \Gamma$ it follows that $w : \circ \in \Gamma_i$ and $w : \circ \notin \Gamma_j$ for $i \neq j$. Let us assume $i = 1$. Then $w : \circ \in \Gamma_1$ and by inductive hypothesis (1) there is some $M_1' \equiv M_1$ such that $|\,M_1'\,| \leq |\,M_1\,|$ and either $M_1' = \nu\vec{w}\,\overline{x}w.\,N$ or $M_1' = \nu\vec{w}\,(\overline{x}w.\,N \mid K)$ where $w \notin \vec{w}$. Note that by the non-homonymy condition we can assume $\vec{w} \notin \mathrm{fn}(M_2)$. Then:
   - If $M_1' = \nu\vec{w}\,\overline{x}w.\,N$ set $M' = \nu\vec{w}\,(\overline{x}w.\,N \mid M_2)$.
   - If $M_1' = \nu\vec{w}\,(\overline{x}w.\,N \mid K)$ set $M' = \nu\vec{w}\,(\overline{x}w.\,N \mid (K \mid M_2))$.

   In both cases $M' \equiv M_1' \mid M_2 \equiv M$ and $|\,M'\,| \leq |\,M\,|$.
2. If $w : \sigma \in \Gamma$ with $\sigma \neq \circ$ then the proof is analogous to the above case. It only requires appealing to the inductive hypothesis (2) rather than to the inductive hypothesis (1).
3. If $w : \sigma \in \Delta$ then from $\mathrm{n}(\Delta_1) \cap \mathrm{n}(\Delta_2) = \emptyset$ it follows that $w : \sigma \in \Delta_i$ and $w : \sigma \notin \Delta_j$. Then the thesis follows from inductive hypothesis (3), analogously to the proof of (1).
4. From $\mathrm{n}(\Delta_1) \cap \mathrm{n}(\Delta_2) = \emptyset$ and the hypothesis $w : \sigma \in \Delta \cap \Gamma$ it follows that $w \in \mathrm{n}(\Delta_i)$ and $w \notin \mathrm{n}(\Delta_j)$ for $i \neq j$. Analogously, $w \in \mathrm{n}(\Gamma_h)$ and $w \notin \mathrm{n}(\Gamma_k)$ for $h \neq k$, from $\mathrm{n}(\Gamma_1) \cap \mathrm{n}(\Gamma_2) = \emptyset$. Note that $\sigma \neq \circ$ from $w \in \mathrm{n}(\Delta)$ and Lemma 28(5). We distinguish two cases depending on whether or not $i = h$.

- Suppose $w \in \mathrm{n}(\Delta_i) \cap \mathrm{n}(\Gamma_i)$ and set, say, $i = 1$. Then by inductive hypothesis (4) there is $M_1' \equiv M_1$ such that $|\, M_1' \,| \leq |\, M_1 \,|$ and either $M_1' = \nu \vec{w} (\overline{w}x.\, N \mid w(z).\, N')$ or $M_1' = \nu \vec{w} (\overline{w}x.\, N \mid w(z).\, N' \mid K)$. Note that in either case we can assume $\vec{w} \notin \mathrm{fn}(M_2)$. Then:
  - If $M_1' = \nu \vec{w} (\overline{w}x.\, N \mid w(z).\, N')$ set $M' = \nu \vec{w} (\overline{w}x.\, N \mid w(z).\, N' \mid M_2)$.
  - If $M_1' = \nu \vec{w} (\overline{w}x.\, N \mid w(z).\, N' \mid K)$ set $M' = \nu \vec{w} (\overline{w}x.\, N \mid w(z).\, N' \mid K \mid M_2)$.

  In each of the above cases $M' \equiv M_1' \mid M_2 \equiv M$ and $|\, M' \,| \leq |\, M \,|$.
- Suppose now that $w \in \mathrm{n}(\Delta_i) \cap \mathrm{n}(\Gamma_j)$ and that $i = 1$. Then by inductive hypothesis (2) and inductive hypothesis (3) there are $M_1' \equiv M_1$ and $M_2' \equiv M_2$ such that $|\, M_1' \,| \leq |\, M_1 \,|$ and $|\, M_2' \,| \leq |\, M_2 \,|$ and

$$\begin{array}{rcl} M_1' = \nu \vec{w}_1\, w(z).\, N_1 & \text{or} & M_1' = \nu \vec{w}_1\, (w(z).\, N_1 \mid K_1) \\ M_2' = \nu \vec{w}_2\, \overline{w}x.\, N_2 & \text{or} & M_2' = \nu \vec{w}_2\, (\overline{w}x.\, N_2 \mid K_2) \end{array}$$

  where $w \notin \vec{w}_1, \vec{w}_2$ and we can assume $\vec{w}_1 \notin \mathrm{fn}(M_2')$ and $\vec{w}_2 \notin \mathrm{fn}(M_1')$. Then, analogously to the previous cases for each of the possible four combinations we can find M' so that $M' \equiv M_1' \mid M_2' \equiv M$ and $|\, M' \,| \leq |\, M \,|$.

□

Lemma 30 If $\Psi; \Delta; \Gamma \vdash M$ and $\Psi' \subseteq \Psi$ and $\mathrm{n}(\Psi) \cap \mathrm{fn}(M) \subseteq \mathrm{n}(\Psi')$, then $\Psi'; \Delta; \Gamma \vdash M$.

Proof: By induction on the derivation of $\Psi; \Delta; \Gamma \vdash M$. □

Lemma 31 If $\Psi; \Delta; \Gamma \vdash M$ and $\mathrm{n}(\Psi') \cap \mathrm{n}(\Psi; \Delta; \Gamma) = \emptyset$, then $\Psi, \Psi'; \Delta; \Gamma \vdash M$.

Proof: By induction on the derivation of $\Psi; \Delta; \Gamma \vdash M$. □

Typing is, as would be expected, invariant under structural congruence:

Lemma 32 If $\Psi; \Delta; \Gamma \vdash M$ and $N \equiv M$, then $\Psi; \Delta; \Gamma \vdash N$.

Proof: Note that $M \equiv N$ iff $M \equiv_1^* N$, where $M' \equiv_1 N'$ if there are a context $\mathcal{K}$ and processes M'', N'' comprising an instance of one of the axioms of structural congruence such that $M' = \mathcal{K}[M'']$ and $N' = \mathcal{K}[N'']$.

The proof is by induction on the length of the proof $M = N_1 \equiv_1 N_2 \equiv_1 \ldots \equiv_1 N_n = N$. The inductive base is void, since $M = N$. As for the inductive step:

Assuming $\Psi; \Delta; \Gamma \vdash N_{n-1}$ we show that $N_{n-1} \equiv_1 N_n$ implies $\Psi; \Delta; \Gamma \vdash N_n$.

In the case when $N_{n-1} = \mathcal{K}[N_{n-1}'] \equiv_1 \mathcal{K}[N_n'] = N_n$ and $N_{n-1}' \equiv_1 N_n'$ is an instance of one of the axioms in Def. 1, the statement is a consequence of the following lemma:

Lemma 33 If $\Psi; \Delta; \Gamma \vdash \mathcal{K}$ with $\Psi'; \Delta'; \Gamma' \vdash [\cdot]$ and $\Psi'; \Delta'; \Gamma' \vdash M$ then $\Psi; \Delta; \Gamma \vdash \mathcal{K}[M]$.

Proof: By induction on the structure of $\mathcal{K}$. $\square$

It remains to show that the statement holds when $N_{n-1} \equiv_1 N_n$ is itself an axiom instance. We work out the most interesting cases. Those omitted do not present difficulties.

$[\nu z\,(M_1 \mid M_2) \equiv (M_1 \mid \nu z\, M_2)$ provided $z \notin \mathrm{fn}(M_1)]$
We distinguish three cases depending on the last typing rule applied to derive

$$\Psi; \Delta; \Gamma \vdash N_{n-1} = \nu z\,(M_1 \mid M_2).$$

Note that in each case $z \notin \mathrm{n}(\Psi, \Delta, \Gamma)$.

- $[res_1]$ In this case we have: $\Psi, z : s; \Delta; \Gamma \vdash M_1 \mid M_2$ and $\Psi, z : s; \Delta_j; \Gamma_j \vdash M_j$ with $\Delta_1 \cup \Delta_2 = \Delta$ and $\Gamma_1 \cup \Gamma_2 = \Gamma$. Then from $z \notin \mathrm{fn}(M_1)$ and Lemma 30:

$$\Psi; \Delta_1; \Gamma_1 \vdash M_1 \qquad\qquad \Psi, z : s; \Delta_2; \Gamma_2 \vdash M_2$$

with $z \notin \mathrm{n}(\Psi, \Delta_2, \Gamma_2)$. Hence the thesis by res_1 and par.

- $[res_2]$ From typing: $\Psi; \Delta; \Gamma, z : \circ \vdash M_1 \mid M_2$ and $\Psi; \Delta_j; \Gamma_j \vdash M_j$ with $\Delta_1 \cup \Delta_2 = \Delta$ and $\Gamma_1 \cup \Gamma_2 = \Gamma \cup \{z : \circ\}$. Note that it cannot be that $z \in \mathrm{n}(\Gamma_1)$ because this, by Lemma 28(4), would contradict the hypothesis $z \notin \mathrm{fn}(M_1)$. Then:

$$\Psi; \Delta_1; \Gamma_1 \vdash M_1 \qquad\qquad \Psi; \Delta_2; \Gamma_2', z : \circ \vdash M_2$$

with $z \notin \mathrm{n}(\Psi, \Delta_2, \Gamma_2')$. Hence the thesis by res_2 and par.

- $[res_3]$ Now we have: $\Psi; \Delta, z : \sigma; \Gamma, z : \sigma \vdash M_1 \mid M_2$ and $\Psi; \Delta_j; \Gamma_j \vdash M_j$ with $\Delta_1 \cup \Delta_2 = \Delta \cup \{z : \sigma\}$ and $\Gamma_1 \cup \Gamma_2 = \Gamma \cup \{z : \sigma\}$. Note that $z \in \mathrm{n}(\Delta_2) \cap \mathrm{n}(\Gamma_2)$. In fact from $z \notin \mathrm{fn}(M_1)$ and Lemma 28(4), it follows that $z \notin \mathrm{n}(\Delta_1, \Gamma_1)$. Hence:

$$\Psi; \Delta_1; \Gamma_1 \vdash M_1 \qquad\qquad \Psi; \Delta_2', z : \sigma; \Gamma_2', z : \sigma \vdash M_2$$

with $z \notin \mathrm{n}(\Psi, \Delta_2', \Gamma_2')$. Then the thesis by res_3 and par.

$[(M_1 \mid \nu z\, M_2) \equiv \nu z\,(M_1 \mid M_2)$ provided $z \notin \mathrm{fn}(M_1)]$
Here we assume that:

$$\Psi; \Delta; \Gamma \vdash N_{n-1} = M_1 \mid \nu z\, M_2$$

and show that $\Psi; \Delta; \Gamma \vdash \nu z\,(M_1 \mid M_2)$. From typing we have:

$$\Psi; \Delta_1; \Gamma_1 \vdash M_1 \qquad\qquad \Psi; \Delta_2; \Gamma_2 \vdash \nu z\, M_2$$

with $\Delta_1 \cup \Delta_2 = \Delta$ and $\Gamma_1 \cup \Gamma_2 = \Gamma$ and $z \notin \mathrm{n}(\Psi, \Delta_2, \Gamma_2)$. Note that $z \notin \mathrm{n}(\Delta_1, \Gamma_1)$ from $z \notin \mathrm{fn}(M_1)$ and Lemma 28(4).
In the cases when the last rule applied to infer $\Psi; \Delta_2; \Gamma_2 \vdash \nu z\, M_2$ is either res_2 or res_3, i.e. it is either $\Psi; \Delta_2; \Gamma_2, z : \circ \vdash M_2$ or $\Psi; \Delta_2, z : \sigma; \Gamma_2, z : \sigma \vdash M_2$, the thesis is an immediate consequence of the above observation. So, the most interesting case to deal with is when the last rule applied to type $\nu z\, M_2$ is res_1 and hence: $\Psi, z : s; \Delta_2; \Gamma_2 \vdash M_2$. In that case, from the hypothesis $z \notin \mathrm{fn}(M_1)$ and Lemma 31 we get: $\Psi, z : s; \Delta_1; \Gamma_1 \vdash M_1$. Then $\Psi, z : s; \Delta_1, \Delta_2; \Gamma_1, \Gamma_2 \vdash M_1 \mid M_2$ and hence the thesis. $\square$

The final result in this section shows how typing changes under intraaction. To prove the lemma it is necessary to examine the effects of substitution on typing.

Lemma 34 If $\Psi;\Delta;\Gamma \vdash M$ and $\theta = \{y/z\}$ and $y, z \notin \mathrm{n}(\Delta,\Gamma)$ and θ respects Ψ, then $\Psi\theta;\Delta;\Gamma \vdash M\theta$.

Proof: By induction on the derivation of $\Psi;\Delta;\Gamma \vdash M$. Each typing rule is considered in turn as the last rule applied. We report in the following the most interesting cases.

[*nil*] $M = \mathbf{0}$ then $M\theta = M$ and $\Psi\theta;\emptyset;\emptyset \vdash M\theta$ where $\Psi\theta = \Psi$ if $z \notin \mathrm{n}(\Psi)$ and $\Psi\theta = \Psi' \cup \{y : s\}$ if $\Psi = \Psi' \cup \{z : s\}$.

[out_1] $M = \overline{x}w.\,M'$ and $\Psi = \Psi' \cup \{x : s\}$ for s such that $\circ \xrightarrow{s} \sigma$ and

$$\Psi', x : s;\emptyset;\{w : \circ\} \vdash \overline{x}w.\,M' \qquad\qquad \Psi';\emptyset;\{w : \sigma\} \vdash M'$$

with $\Psi'(x) \simeq s$ and $w \notin \mathrm{n}(\Psi', x)$. Note that $w \neq z, y$ from the hypothesis $y, z \notin \mathrm{n}(\Delta,\Gamma)$. Then $M\theta = \overline{x\theta}w.\,M'\theta$. By inductive hypothesis:

$$\Psi'\theta;\emptyset;\{w : \sigma\} \vdash M'\theta$$

Moreover $\Psi'\theta(x\theta) \simeq s$ from $\Psi'\theta(x\theta) = \Psi'(x)$, and $w \notin \mathrm{n}(\Psi'\theta, x\theta) \subseteq \mathrm{n}(\Psi', y, x)$ from $w \notin \mathrm{n}(\Psi', x)$ and $w \neq y$. Then from out_1:

$$\Psi'\theta, x\theta : s;\emptyset;\{w : \circ\} \vdash M\theta = \overline{x\theta}w.\,M'\theta$$

[out_2] $M = \overline{x}w.\,M'$ and $\Psi = \Psi' \cup \{w : s\}$ for s such that $\circ \neq \sigma \xrightarrow{s} \tau$ and

$$\Psi', w : s;\emptyset;\{x : \sigma\} \vdash \overline{x}w.\,M' \qquad\qquad \Psi';\emptyset;\{x : \tau\} \vdash M'$$

with $\Psi'(w) \simeq s$ and $x \notin \mathrm{n}(\Psi', w)$. Note that $x \neq z, y$, so $M\theta = \overline{x}w\theta.\,M'\theta$ and by inductive hypothesis:

$$\Psi'\theta;\emptyset;\{x : \tau\} \vdash M'\theta$$

We have also: $\Psi'\theta(w\theta) = \Psi'(w) \simeq s$, and $x \notin \mathrm{n}(\Psi'\theta, w\theta) \subseteq \mathrm{n}(\Psi', y, w)$ from $x \notin \mathrm{n}(\Psi', w)$ and $x \neq y$. Then from out_2:

$$\Psi'\theta, w\theta : s;\emptyset;\{x : \sigma\} \vdash M\theta = \overline{x}w\theta.\,M'\theta$$

[inp_1] $M = x(w).\,M'$ and $\Psi = \Psi' \cup \{x : s\}$ for s such that $\circ \xrightarrow{s} \sigma$ and

$$\Psi', x : s;\emptyset;\emptyset \vdash x(w).\,M' \qquad\qquad \Psi';\{w : \sigma\};\emptyset \vdash M'$$

with $\Psi'(x) \simeq s$ and $w \notin \mathrm{n}(\Psi')$. Note that we can assume $w \neq y, z$. In fact, if $w = z$ then from the non-homonymy condition $z \notin \mathrm{fn}(M')$, hence $M\theta = M$. On the other hand, if $w = y$ then w is renamed into a fresh w' in $M\theta = x\theta(w').\,M'\{w'/w\}\{w/z\}$ in order to avoid captures. From $w \neq y, z$ and $\Psi';\{w : \sigma\};\emptyset \vdash M'$ we deduce by inductive hypothesis:

$$\Psi'\theta;\{w : \sigma\};\emptyset \vdash M'\theta$$

It also holds: $\Psi'\theta(x\theta) = \Psi'(x) \simeq s$, and $w \notin \mathrm{n}(\Psi'\theta) \subseteq \mathrm{n}(\Psi', y)$. Hence:

$$\Psi'\theta, x\theta : s;\emptyset;\emptyset \vdash M\theta = x\theta(w).\,M'\theta$$

$[inp_2]$ $M = x(w).\, M'$ and

$$\Psi; \{x : \sigma\}; \emptyset \vdash x(w).\, M' \qquad\qquad \Psi, w : s; \{x : \tau\}; \emptyset \vdash M'$$

for s such that $\circ \neq \sigma \xrightarrow{s} \tau$ and with $x, w \notin \mathrm{n}(\Psi)$. As in the above case we can assume $w \neq y, z$. Also note that by hypothesis $y, z \neq x$. Then $M\theta = x(w).\, M'\theta$ and by inductive hypothesis:

$$\Psi\theta, w : s; \{x : \tau\}; \emptyset \vdash M'\theta$$

with $x, w \notin \mathrm{n}(\Psi\theta) \subseteq \mathrm{n}(\Psi, y)$ from $x, w \notin \mathrm{n}(\Psi)$ and $x \neq y$ and $w \neq y$. Hence from inp_2:

$$\Psi\theta; \{x : \sigma\}; \emptyset \vdash M\theta = x(w).\, M'\theta$$

$[par]$ $M = M_1 \mid M_2$ and $\Delta = \Delta_1 \cup \Delta_2$, $\Gamma = \Gamma_1 \cup \Gamma_2$ with Δ_1, Γ_1 and Δ_2, Γ_2 complementary and $\Psi; \Delta_j; \Gamma_j \vdash M_j$. Note that $y, z \notin \mathrm{n}(\Delta_j, \Gamma_j)$ from the hypothesis $y, z \notin \mathrm{n}(\Delta, \Gamma)$. Then by inductive hypothesis: $\Psi\theta; \Delta_j; \Gamma_j \vdash M_j\theta$ and hence: $\Psi\theta; \Delta; \Gamma \vdash M\theta = M_1\theta \mid M_2\theta$. $\Box$

Lemma 35 If $\Psi; \Delta; \Gamma \vdash M$ and $\theta = \{y/z\}$ and $z : \sigma \in \Delta$ and $z \notin \mathrm{n}(\Gamma)$ and $y \notin \mathrm{n}(\Psi, \Delta)$ and ($y : \tau \in \Gamma$ implies $\sigma = \tau$), then $\Psi; \Delta\theta; \Gamma \vdash M\theta$.

Proof: By induction on the derivation of $\Psi; \Delta; \Gamma \vdash M$.

$[inp_2]$ $M = z(w).\, M'$ and

$$\Psi; \{z : \sigma\}; \emptyset \vdash z(w).\, M' \qquad\qquad \Psi, w : s; \{z : \tau\}; \emptyset \vdash M'$$

for s such that $\circ \neq \sigma \xrightarrow{s} \tau$ and with $z, w \notin \mathrm{n}(\Psi)$. Note that $y \notin \mathrm{n}(\Psi, z)$ by hypothesis. We can also assume $y, z \neq w$ (for details see the inp_1 case of the proof of Lemma 34). Then $M\theta = z\theta(w).\, M'\theta = y(w).\, M'\theta$ and by $y \notin \mathrm{n}(\Psi, w, z)$ and inductive hypothesis:

$$\Psi, w : s; \{y : \tau\}; \emptyset \vdash M'\theta$$

Since $y, w \notin \mathrm{n}(\Psi)$, from inp_2 we have:

$$\Psi; \{y : \sigma\}; \emptyset \vdash M\theta = y(w).\, M'\theta$$

$[par]$ $M = M_1 \mid M_2$ and $\Delta = \Delta_1 \cup \Delta_2$, $\Gamma = \Gamma_1 \cup \Gamma_2$ with Δ_1, Γ_1 and Δ_2, Γ_2 complementary and

$$\Psi; \Delta_j; \Gamma_j \vdash M_j$$

From $z : \sigma \in \Delta$ and the complementarity condition it follows that $z \in \mathrm{n}(\Delta_i)$ and $z \notin \mathrm{n}(\Delta_j)$ with $i \neq j$. Moreover $z \notin \mathrm{n}(\Psi)$ by Lemma 28(2). Then $z \notin \mathrm{fn}(M_j)$ from $z \notin \mathrm{n}(\Gamma)$ and Lemma 28(3). Hence $M\theta \equiv M_i\theta \mid M_j$.
By inductive hypothesis:

$$\Psi; \Delta_i\theta; \Gamma_i \vdash M_i\theta$$

Note that $\Delta_i\theta, \Gamma_i$ and Δ_j, Γ_j are complementary. In fact:

- $\mathrm{n}(\Delta_i\theta) \cap \mathrm{n}(\Delta_j) \subseteq \mathrm{n}(\Delta_i, y) \cap \mathrm{n}(\Delta_j) = \{y\} \cap \mathrm{n}(\Delta_j) = \emptyset$ from $y \notin \mathrm{n}(\Delta)$;
- $\Delta_i\theta, \Delta_j$ and $\Gamma_i \cup \Gamma_j$ are compatible from the hypothesis that $y : \tau \in \Gamma$ implies $\sigma = \tau$.

Hence we have:

$$\Psi; \Delta_i\theta, \Delta_j; \Gamma_i, \Gamma_j \vdash M_i\theta \mid M_j$$

$[res_1,\ res_2,\ res_3]$ $M = \nu w\, M'$. As for the above input prefix case we can assume $z, y \neq w$. Hence in each of the three cases for restriction the thesis is an immediate consequence of the inductive hypothesis. □

Lemma 36 If $\Psi; \Delta; \Gamma \vdash M$ and $M \longrightarrow M'$, then $\Psi; \Delta'; \Gamma' \vdash M'$ where

1. $\Delta' = \Delta$ and $\Gamma' = \Gamma$, or
2. $\Delta' = \Delta \cup \{y : \sigma\}$ and $\Gamma' = \Gamma - \{y : \circ\} \cup \{y : \sigma\}$ where $y : \circ \in \Gamma$ and $\circ \xrightarrow{s} \sigma$, or
3. $\Delta' = \Delta - \{y : \sigma\} \cup \{y : \tau\}$ and $\Gamma' = \Gamma - \{y : \sigma\} \cup \{y : \tau\}$ where $y : \sigma \in \Delta \cap \Gamma$ and $\circ \neq \sigma \xrightarrow{s} \tau$.

Proof: From the hypothesis that $\Psi; \Delta; \Gamma \vdash M \longrightarrow M'$ and Lemma 5, we have:

$$\begin{array}{rcccl} M & \equiv & \nu\vec{w}\, N & = & \nu\vec{w}\,(\overline{x}y.\, M_1 \mid x(z).\, M_2 \mid M_3) \\ M' & \equiv & \nu\vec{w}\, N' & = & \nu\vec{w}\,(M_1 \mid M_2\{y/z\} \mid M_3) \end{array}$$

Then $\Psi; \Delta; \Gamma \vdash \nu\vec{w}\, N$ from Lemma 32. Assume that in $\langle \Psi; \Delta; \Gamma \vdash \nu\vec{w}\, N \rangle$ the typing judgment for $(\overline{x}y.\, M_1 \mid x(z).\, M_2 \mid M_3)$ is

$$\Psi'; \Delta'; \Gamma' \vdash (\overline{x}y.\, M_1 \mid x(z).\, M_2 \mid M_3)$$

and is derived from $\Psi'; \Delta_1; \Gamma_1 \vdash (\overline{x}y.\, M_1 \mid x(z).\, M_2)$ and $\Psi'; \Delta_2; \Gamma_2 \vdash M_3$.
We distinguish two cases depending on the structure of $\langle \Psi'; \Delta_1; \Gamma_1 \vdash \overline{x}y.\, M_1 \mid x(z).\, M_2 \rangle$.

1. Suppose that the last rule applied to type $\overline{x}y.\, M_1$ is out_1.
   Then $y \notin \mathrm{n}(\Psi')$ and $x : s \in \Psi'$ for s such that $\circ \xrightarrow{s} \sigma$. So the last rule applied to type $x(z).\, M_2$ is inp_1 and we have:

$$\frac{\Psi'; \emptyset; \{y : \sigma\} \vdash M_1}{\Psi', x : s; \emptyset; \{y : \circ\} \vdash \overline{x}y.\, M_1} \qquad \frac{\Psi'; \{z : \sigma\}; \emptyset \vdash M_2}{\Psi', x : s; \emptyset; \emptyset \vdash x(z).\, M_2} \qquad (1)$$

   Hence $\Delta' = \Delta_2$ and $\Gamma' = \Gamma_2 \cup \{y : \circ\}$ and we have:

$$\Psi'; \Delta_2; \Gamma_2, y : \circ \vdash (\overline{x}y.\, M_1 \mid x(z).\, M_2 \mid M_3) \qquad (2)$$

Note that $y \notin \mathrm{n}(\Psi', z)$, since $y \neq z$ from the non-homonymy condition. Also note that from typing $z \notin \mathrm{n}(\Psi')$. Then from (1) and Lemma 35:

$$\Psi'; \{y : \sigma\}; \emptyset \vdash M_2\{y/z\} \tag{3}$$

From the complementarity of Δ_1, Γ_1 and Δ_2, Γ_2 and from $y : \circ \in \Gamma_1$ it follows that $y \notin \mathrm{n}(\Gamma_2)$. Also, $y \notin \mathrm{n}(\Delta_2)$ from $y : \circ \in \Gamma'$ and Lemma 28(5). Then $(\{y : \sigma\}, \{y : \sigma\})$ and (Δ_2, Γ_2) are complementary and from (1) and (3) we get:

$$\Psi'; \Delta_2, y : \sigma; \Gamma_2, y : \sigma \vdash (M_1 \mid M_2\{y/z\} \mid M_3) \tag{4}$$

We can now compare the type inference for $\nu\vec{w}\, N'$ rooted in (4) with the type inference for $\nu\vec{w}\, N$ rooted in (2). From this comparison and Lemma 32 we have the following two possible typing judgments for M':

- If $y \notin \vec{w}$ then $\Psi; \Delta, y : \sigma; \Gamma - \{y : \circ\}, y : \sigma \vdash M'$.
  In fact the sequence of restriction rules applied in the type inference rooted in (4) is the same as the sequence of restriction rules applied in the type inference rooted in (2).
- If $y \in \vec{w}$ then $\Psi; \Delta; \Gamma \vdash M'$.
  In this case res_3 is applied correspondingly to νy in the inference rooted in (4) instead of the rule res_2 applied in the inference rooted in (2). The sequence of restriction rules applied in the two type inferences is otherwise the same.

2. Now suppose that the last rule applied to type $\overline{x}y.\, M_1$ is out_2.
Then $y : s \in \Psi'$ for s such that $\circ \neq \sigma \xrightarrow{s} \tau$. So the last rule applied to type $x(z).\, M_2$ is inp_2 and we have:

$$\frac{\Psi'; \emptyset; \{x : \tau\} \vdash M_1}{\Psi', y : s; \emptyset; \{x : \sigma\} \vdash \overline{x}y.\, M_1} \qquad \frac{\Psi', z : s; \{x : \tau\}; \emptyset \vdash M_2}{\Psi'; \{x : \sigma\}; \emptyset \vdash x(z).\, M_2} \tag{5}$$

Note that from the non-homonymy condition and typing $z, y \neq x$ and $z \notin \mathrm{n}(\Psi')$. Then from (5) and Lemma 34:

$$\Psi', y : s; \{x : \tau\}; \emptyset \vdash M_2\{y/z\} \tag{6}$$

Under the above hypotheses $\Delta' = \Delta_2 \cup \{x : \sigma\}$ and $\Gamma' = \Gamma_2 \cup \{x : \sigma\}$. Hence:

$$\Psi'; \Delta_2, x : \sigma; \Gamma_2, x : \sigma \vdash (\overline{x}y.\, M_1 \mid x(z).\, M_2 \mid M_3) \tag{7}$$

From the complementarity of $(\{x : \sigma\}, \{x : \sigma\})$ and (Δ_2, Γ_2), the complementarity of $(\{x : \tau\}, \{x : \tau\})$ and (Δ_2, Γ_2) follows. Hence from (5) and (6) we have:

$$\Psi'; \Delta_2, x : \tau; \Gamma_2, x : \tau \vdash (M_1 \mid M_2\{y/z\} \mid M_3) \tag{8}$$

The type inference for $\nu\vec{w}\, N'$ rooted in (8) is now compared to the type inference for $\nu\vec{w}\, N$ rooted in (7), leading to the following two cases.

- If $x \notin \vec{w}$ then $\Psi; \Delta - \{x : \sigma\}, x : \tau; \Gamma - \{x : \sigma\}, x : \tau \vdash M'$.

- If $x \in \vec{w}$ then $\Psi; \Delta; \Gamma \vdash M'$.

Hence the thesis, from the possible typing judgments for M' collected above. $\square$

The rules for typing monadic contexts are like the rules in Tab. 2, with the addition of the rule: $\Psi; \Delta; \Gamma \vdash [\cdot]$ for any Ψ, Δ, Γ. A monadic context $\mathcal{K}$ is a $\lambda^m(\Psi, \Delta, \Delta')$-*context* if assuming $\Psi; \Delta; \Delta \vdash [\cdot]$ we can infer $\Psi'; \Delta'; \Delta' \vdash \mathcal{K}$ for some Ψ'.

Definition 37 M and N are *barbed λ^m-congruent*, $M \approx_\lambda^m N$, if there are Ψ, Δ such that $\Psi; \Delta; \Delta \vdash M, N$ and $\mathcal{K}[M] \dot{\approx} \mathcal{K}[N]$ for every $\lambda^m(\Psi, \Delta, \emptyset)$-context $\mathcal{K}$.

Recalling the counterexample $P = \overline{x}y.\, \overline{x}y.\, \mathbf{0}$ and $Q = \overline{x}y.\, \mathbf{0} \mid \overline{x}y.\, \mathbf{0}$ to full abstraction of the translation, and the monadic context $\mathcal{K} = [\cdot] \mid x(z).\, x(w).\, a(v).\, \mathbf{0}$ such that $\mathcal{K}[[\![P]\!]] \not\dot{\approx} \mathcal{K}[[\![Q]\!]]$, note that $\mathcal{K}$ is not a $\lambda^m(\Psi, \Delta, \emptyset)$-context for any Ψ and Δ.

4 Main results

In this section we prove the main results. We begin by relating typing of P under λ and typing of $[\![P]\!]$ under λ^m. First, typing is preserved by the translation:

Lemma 38 If $\Psi \vdash P$ then $\Psi; \emptyset; \emptyset \vdash [\![P]\!]$.

Proof: The proof is by induction on the inference of $\Psi \vdash P$.

Suppose $\Psi \vdash P$ where $P = \overline{x}\vec{y}.\, Q$ and $\vec{y} = y_1 \ldots y_n$. Then $\Psi' \vdash Q$ and $\Psi = \Psi', x : s, y_1 : t_1, \ldots, y_n : t_n$ where $\lambda(s) = (t_1, \ldots, t_n)$ and $\Psi'(x) \simeq s$ and $\Psi'(y_i) \simeq t_i$. By assumption, $\Psi'; \emptyset; \emptyset \vdash [\![Q]\!]$. Now $[\![P]\!] = \nu w\, \overline{x}w.\, \overline{w}y_1.\, \ldots.\, \overline{w}y_n.\, [\![Q]\!]$ where $w \notin \mathrm{n}(\Psi)$. Hence as $\circ \neq s^n \xrightarrow{t_n} \bullet$ and $\Psi'(y_n) \simeq t_n$,

$$\Psi', y_n : t_n; \emptyset; \{w : s^n\} \vdash \overline{w}y_n.\, [\![Q]\!]$$

and by similar reasoning, as $\circ \neq s^1 \xrightarrow{t_1} s^2$ and $\Psi'(y_1) \simeq t_1$,

$$\Psi', y_1 : t_1, \ldots, y_n : t_n; \emptyset; \{w : s^1\} \vdash \overline{w}y_1.\, \ldots.\, \overline{w}y_n.\, [\![Q]\!].$$

Hence since $\circ \xrightarrow{s} s^1$ and $\Psi'(x) \simeq s$,

$$\Psi', x : s, y_1 : t_1, \ldots, y_n : t_n; \emptyset; \{w : \circ\} \vdash \overline{x}w.\, \overline{w}y_1.\, \ldots.\, \overline{w}y_n.\, [\![Q]\!]$$

and hence

$$\Psi', x : s, y_1 : t_1, \ldots, y_n : t_n; \emptyset; \emptyset \vdash \nu w\, \overline{x}w.\, \overline{w}y_1.\, \ldots.\, \overline{w}y_n.\, [\![Q]\!]$$

i.e. $\Psi; \emptyset; \emptyset \vdash [\![P]\!]$.

Suppose $\Psi \vdash P$ where $P = x(\vec{z}).\, Q$ and $\vec{z} = z_1 \ldots z_n$. Then $\Psi', z_1 : t_1, \ldots, z_n : t_n \vdash Q$ and $\Psi = \Psi', x : s$ where $\lambda(s) = (t_1, \ldots, t_n)$ and $\Psi'(x) \simeq s$ and $z_i \notin \mathrm{n}(\Psi')$. By assumption, $\Psi', z_1 : t_1, \ldots, z_n : t_n; \emptyset; \emptyset \vdash [\![Q]\!]$. Now $[\![P]\!] = x(w).\, w(z_1).\, \ldots.\, w(z_n).\, [\![Q]\!]$ where $w \notin \mathrm{n}(\Psi', x, \vec{z})$. Hence as $\circ \neq s^n \xrightarrow{t_n} \bullet$,

$$\Psi', z_1 : t_1, \ldots, z_{n-1} : t_{n-1}; \{w : s^n\}; \emptyset \vdash w(z_n).\, [\![Q]\!]$$

and by similar reasoning, as $\circ \neq s^1 \xrightarrow{t_1} s^2$,

$$\Psi'; \{w : s^1\}; \emptyset \vdash w(z_1). \ldots . w(z_n). [\![Q]\!].$$

Hence since $\circ \xrightarrow{s} s^1$ and $\Psi'(x) \simeq s$,

$$\Psi', x : s; \emptyset; \emptyset \vdash x(w). w(z_1). \ldots . w(z_n). [\![Q]\!]$$

i.e. $\Psi; \emptyset; \emptyset \vdash [\![P]\!]$.

The other cases are straightforward. For instance suppose $\Psi \vdash P$ where $P = !Q$. Then $\Psi \vdash Q$ and so by assumption $\Psi; \emptyset; \emptyset \vdash [\![Q]\!]$ and so $\Psi; \emptyset; \emptyset \vdash ![\![Q]\!]$, i.e. $\Psi; \emptyset; \emptyset \vdash [\![P]\!]$. $\square$

Secondly, if the translation of a process can be typed, then the process itself can be typed:

Lemma 39 If $\Psi; \Delta; \Gamma \vdash [\![P]\!]$ then $\Delta = \Gamma = \emptyset$ and $\Psi \vdash P$.

Proof: The proof is by induction on the structure of P.

Suppose $\Psi; \Delta; \Gamma \vdash [\![P]\!]$ where $P = \overline{x}\vec{y}. Q$ with $\vec{y} = y_1 \ldots y_n$, so

$$\Psi; \Delta; \Gamma \vdash \nu w\, \overline{x}w. \overline{w}y_1. \ldots . \overline{w}y_n. [\![Q]\!]$$

where $w \notin \mathrm{n}(\Psi, \Delta, \Gamma)$.

Suppose $\Psi; \Delta; \Gamma \vdash [\![P]\!]$ is inferred from

$$\Psi, w : s; \Delta; \Gamma \vdash \overline{x}w. \overline{w}y_1. \ldots . \overline{w}y_n. [\![Q]\!].$$

Then it must be that $\Delta = \emptyset$ and $\Gamma = \{x : \sigma\}$ and $\Psi; \emptyset; \Gamma' \vdash \overline{w}y_1. \ldots . \overline{w}y_n. [\![Q]\!]$ where $\circ \neq \sigma \xrightarrow{s} \tau$ and $\Gamma' = \{x : \tau\}$ and $\Psi(w) \simeq s$. But this is impossible by the form of the typing rules for output prefixes and the facts that $\sigma, \tau \neq \circ$.

So suppose $\Psi; \Delta; \Gamma \vdash [\![P]\!]$ is inferred from

$$\Psi; \Delta'; \Gamma' \vdash \overline{x}w. \overline{w}y_1. \ldots . \overline{w}y_n. [\![Q]\!]$$

where $\Delta' = \Delta, w : \sigma$ and $\Gamma' = \Gamma, w : \sigma$ where $\sigma \neq \circ$. Then it must be that $\Delta' = \emptyset$ and so $\sigma = \bullet$. Hence $\Gamma = \emptyset$. But this is impossible by the form of the typing rules for output prefixes by reasoning similar to that above.

Hence $\Psi; \Delta; \Gamma \vdash [\![P]\!]$ is inferred from

$$\Psi; \Delta; \Gamma, w : \circ \vdash \overline{x}w. \overline{w}y_1. \ldots . \overline{w}y_n. [\![Q]\!].$$

So $\Delta = \Gamma = \emptyset$ where $w \notin \mathrm{n}(\Gamma)$, and

$$\Psi_1; \emptyset; \{w : \sigma_1\} \vdash \overline{w}y_1. \ldots . \overline{w}y_n. [\![Q]\!]$$

where $\Psi = \Psi_1, x : s$ and $\Psi_1(x) \simeq s$ and $\circ \xrightarrow{s} \sigma_1$. In turn,

$$\Psi_2; \emptyset; \{w : \sigma_2\} \vdash \overline{w}y_2. \ldots . \overline{w}y_n. [\![Q]\!]$$

where $\Psi_1 = \Psi_2, y_1 : t_1$ and $\Psi_2(y_1) \simeq t_1$ and $\sigma_1 \xrightarrow{t_1} \sigma_2$. By similar reasoning,

$$\Psi_n; \emptyset; \{w : \sigma_n\} \vdash \overline{w} y_n. \llbracket Q \rrbracket$$

where $\Psi_{n-1} = \Psi_n, y_{n-1} : t_{n-1}$ and $\Psi_n(y_{n-1}) \simeq t_{n-1}$ and $\sigma_{n-1} \xrightarrow{t_{n-1}} \sigma_n$, and hence

$$\Psi_{n+1}; \emptyset; \{w : \sigma_{n+1}\} \vdash \llbracket Q \rrbracket$$

where $\Psi_n = \Psi_{n+1}, y_n : t_n$ and $\Psi_{n+1}(y_n) \simeq t_n$ and $\sigma_n \xrightarrow{t_n} \sigma_{n+1}$. By assumption $\{w : \sigma_{n+1}\} = \emptyset$, so $\sigma_{n+1} = \bullet$, and $\Psi_{n+1} \vdash Q$. Hence

$$\circ \xrightarrow{s} \sigma_1 \xrightarrow{t_1} \ldots \xrightarrow{t_{n-1}} \sigma_n \xrightarrow{t_n} \bullet$$

so $\lambda(s) = (t_1 \ldots t_n)$. Further, if $x = y_i$ then $s = t_i$ because $\Psi_{i+1}(y_i) \simeq t_i$, and similarly if $y_i = y_j$ then $t_i = t_j$. Hence $\Psi \vdash P$. So in summary, $\Delta = \Gamma = \emptyset$ and $\Psi \vdash P$.

The case $P = x(\vec{z}). Q$ is similar and slightly simpler. The remaining cases are straightforward. For instance $P = !Q$ and $\Psi; \Delta; \Gamma \vdash \llbracket P \rrbracket$ then $\Delta = \Gamma = \emptyset$ and $\Psi; \emptyset; \emptyset \vdash \llbracket Q \rrbracket$. Hence by assumption $\Psi \vdash Q$, and so $\Psi \vdash P$. $\Box$

This lemma does not hold if we admit prefixes of the forms $\overline{x}\langle\rangle$ and $x()$. For example if $\lambda(s) = (s)$, then $\emptyset; \emptyset; x : s^1 \vdash \llbracket \overline{x}\langle\rangle. \mathbf{0} \rrbracket, \llbracket x(). \mathbf{0} \rrbracket$.

We now prove that the translation is sound.

Theorem 40 If $\llbracket P \rrbracket \approx^m_\lambda \llbracket Q \rrbracket$ then $P \approx_\lambda Q$.

Proof: Consider first $\mathcal{B} = \{(R, \llbracket R \rrbracket) \mid R \text{ a } \lambda\text{-process}\}$. We noted in Section 2 that

1. $R \downarrow_\mu$ iff $\llbracket R \rrbracket \downarrow_\mu$, and
2. if $R \longrightarrow R'$ then $\llbracket R \rrbracket \longrightarrow N \Longrightarrow \llbracket R' \rrbracket$ with $N \approx^m \llbracket R' \rrbracket$.

We show that

$$\text{if } \llbracket R \rrbracket \longrightarrow N \text{ then } N \approx^m \llbracket R' \rrbracket \text{ where } R \longrightarrow R'. \tag{9}$$

Assertion (9) does not hold without the assumption that R is a λ-process: consider for instance $R = \overline{a}\langle bc\rangle. \mathbf{0} \mid a(z). \mathbf{0}$. Assertion (9) is harder to prove than it may at first sight appear. The reason is that the structural rule may be applied arbitrarily in inferring $\llbracket R \rrbracket \longrightarrow N$, and it takes some work to see that a suitable R' can always be found. The monadic type system plays a key role in carrying out that work.

We write $M \rightsquigarrow M'$ if there are $\langle \Psi; \Delta; \Gamma \vdash M \rangle$ and w monadic in it such that M' is obtained from M by replacing a subterm N by N' where

1. $N = \nu w\, \nu z\, K$ and $N' = \nu z\, \nu w\, K$ where z is not monadic in $\langle \Psi; \Delta; \Gamma \vdash M \rangle$, or
2. $N = \nu w\, (K \mid K')$ and $N' = K \mid \nu w\, K'$ where $w \notin \mathrm{fn}(K)$, or
3. $N = \nu w\, (K \mid K')$ and $N' = \nu w\, K \mid K'$ where $w \notin \mathrm{fn}(K')$.

Note that $M \rightsquigarrow M'$ implies $M' \equiv_1 M$. In a structural manipulation of $\llbracket R \rrbracket$, a $\rightsquigarrow$-transformation involving νw changes the term structure in such a way as to move the restriction towards the prefix of the form $\overline{x} w$ introduced in translating R. The $\rightsquigarrow$-transformations and their inverses have no direct counterparts in a manipulation of R. We have, however,

Lemma 41 If $[\![R]\!] \equiv M$ then $M \rightsquigarrow^* [\![R'']\!]$ where $R'' \equiv R$.

Proof: Since $\equiv$ is $\equiv_1^*$, it suffices to show that

> if $M \rightsquigarrow^* [\![R'']\!]$ and $M \equiv_1 M'$, then either $M' \rightsquigarrow^* [\![R'']\!]$, or $M' \rightsquigarrow^* [\![R']\!]$ where $R' \equiv_1 R''$.

If $M' \rightsquigarrow M$ then clearly $M' \rightsquigarrow^* [\![R'']\!]$. Further, if $M \rightsquigarrow M'$ then again $M' \rightsquigarrow^* [\![R'']\!]$, since the $\rightsquigarrow$-transformations commute with one another. So suppose the transformation between M and M' is not of these kinds. The proof is then a case analysis on the axiom of structural congruence applied in the transformation. We give just one case. Suppose M is $\mathcal{K}[\,!N]$ and M' is $\mathcal{K}[N \mid !N]$. In $M \rightsquigarrow^* [\![R'']\!]$, each restricted monadic name is reunited with its partner via $\rightsquigarrow$-transformations. The same (fourth) axiom of structural congruence can then be applied to the corresponding subterm of R'' to yield R'. Then using $\rightsquigarrow^{-1}$-transformations, the term M' can be obtained from $[\![R']\!]$. A more formal proof can be given by appealing to a notion of embedding of an inference of $\Psi \vdash R$ in an inference of $\Psi; \emptyset; \emptyset \vdash [\![R]\!]$. This makes precise the notion of 'corresponding subterm' referred to above. $\square$

Now returning to the proof of (1), suppose $[\![R]\!] \equiv M = \nu\vec{w}\,(\overline{x}w.\,M_1 \mid x(z).\,M_2 \mid M_3)$ and $M \mapsto M' = \nu\vec{w}\,(M_1 \mid M_2\{^w\!/\!_z\} \mid M_3) \equiv N$. Then $R \equiv R''$ where $M \rightsquigarrow^* [\![R'']\!]$. By the form of M, $R'' = \nu\vec{v}\,(\overline{x}\vec{y}.\,P_1 \mid x(\vec{z}).\,P_2 \mid P_3)$ and $R'' \mapsto R' = \nu\vec{v}\,(P_1 \mid P_2\{^{\vec{y}}\!/\!_{\vec{z}}\} \mid P_3)$ where $\nu w\,\overline{x}w.\,M_1 \rightsquigarrow^* [\![\overline{x}\vec{y}.\,P_1]\!]$, $x(z).\,M_2 \rightsquigarrow^* [\![x(\vec{z}).\,P_2]\!]$, and $\nu\vec{u}\,M_3 \rightsquigarrow^* [\![P_3]\!]$ where $\vec{w}$ is a permutation of $\vec{v}w\vec{u}$. Further $N \Longrightarrow [\![R']\!]$ and $[\![R']\!] \approx^m N$ as required. By 1, 2 and (9) we have

Corollary 42 If R is a λ-process then $[\![R]\!] \dot{\approx} R$.

To complete the proof of the theorem, suppose $[\![P]\!] \approx_\lambda^m [\![Q]\!]$ and Ψ is such that $\Psi; \emptyset; \emptyset \vdash [\![P]\!], [\![Q]\!]$ and $\mathcal{K}[[\![P]\!]] \dot{\approx} \mathcal{K}[[\![Q]\!]]$ for every $\lambda^m(\Psi, \emptyset, \emptyset)$-context $\mathcal{K}$. Then $\Psi \vdash P, Q$ and if $\mathcal{C}$ is a $\lambda(\Psi)$-context then $[\![\mathcal{C}]\!]$ is a $\lambda^m(\Psi, \emptyset, \emptyset)$-context and so

$$\mathcal{C}[P] \dot{\approx} [\![\mathcal{C}[P]]\!] = [\![\mathcal{C}]\!][[\![P]\!]] \dot{\approx} [\![\mathcal{C}]\!][[\![Q]\!]] = [\![\mathcal{C}[Q]]\!] \dot{\approx} \mathcal{C}[Q].$$

Hence $P \approx_\lambda Q$. $\square$

We now prove that the translation is complete.

Theorem 43 If $P \approx_\lambda Q$ then $[\![P]\!] \approx_\lambda^m [\![Q]\!]$.

Proof: Suppose $P \approx_\lambda Q$ and Ψ_0 is such that $\Psi_0 \vdash P, Q$ and $\mathcal{C}[P] \dot{\approx} \mathcal{C}[Q]$ for every $\lambda(\Psi_0)$-context $\mathcal{C}$. Then $\Psi_0; \emptyset; \emptyset \vdash [\![P]\!], [\![Q]\!]$. Suppose $\mathcal{K}$ is a $\lambda^m(\Psi_0, \emptyset, \emptyset)$-context. The crucial fact is:

$$\text{There is a } \lambda(\Psi_0)\text{-context } \mathcal{C} \text{ such that } \mathcal{K}[[\![P]\!]] \dot{\approx} [\![\mathcal{C}]\!][[\![P]\!]] \text{ and } \mathcal{K}[[\![Q]\!]] \dot{\approx} [\![\mathcal{C}]\!][[\![Q]\!]]. \quad (10)$$

From this, using Corollary 42 we have

$$\mathcal{K}[[\![P]\!]] \dot{\approx} [\![\mathcal{C}]\!][[\![P]\!]] = [\![\mathcal{C}[P]]\!] \dot{\approx} \mathcal{C}[P] \dot{\approx} \mathcal{C}[Q] \dot{\approx} [\![\mathcal{C}[Q]]\!] = [\![\mathcal{C}]\!][[\![Q]\!]] \dot{\approx} \mathcal{K}[[\![Q]\!]].$$

To establish (10) we show

Lemma 44 Suppose $\Psi;\Delta;\Delta,\vec{w}:\circ\vdash\mathcal{K}$ assuming $\Psi_0;\Delta_0;\Delta_0,\vec{w}_0:\circ\vdash[\cdot]$. Then there exists $\mathcal{C}$ such that

1. $\Psi\vdash\mathcal{C}$ assuming $\Psi_0\vdash[\cdot]$, and
2. if $\Psi_0;\Delta_0;\Delta_0,\vec{w}_0:\circ\vdash M$ then $\nu\vec{v}\vec{w}\,\mathcal{K}[M]\approx^m_\lambda [\![\mathcal{C}]\!][\nu\vec{v}_0\vec{w}_0\,M]$ where $\vec{v}=\mathrm{n}(\Delta)$ and $\vec{v}_0=\mathrm{n}(\Delta_0)$.

Proof: By induction on the derivation of $\Psi;\Delta;\Delta,\vec{w}:\circ\vdash\mathcal{K}$. Note that the assertion concerns $\approx^m_\lambda$ and not $\dot{\approx}$; this is important in the induction.

The most difficult cases are the prefixes. Of these, we only show the argument for the input prefix; the argument for output prefix is dual to it.

$[inp_1]$ $\mathcal{K}=x(z).\mathcal{K}_0$ and by the typing rules $\Psi=\Psi'\cup\{x:s\}$ and $\Delta,\vec{w}=\emptyset$ and $\Psi';\{z:s^1\};\emptyset\vdash\mathcal{K}_0$ (where $\circ\xrightarrow{s}s^1$). For clarity we assume that $\lambda(s)=(t_1t_2)$ – this retains the essence of the problem. By Lemma 29 we have (in the most complicated of the four possible combinations):

$$\mathcal{K}\equiv\mathcal{K}'=x(z).\,\nu\vec{u}_1\,(z(z_1).\,\nu\vec{u}_2\,(z(z_2).\,\mathcal{K}_3\mid\mathcal{K}_2)\mid\mathcal{K}_1)$$

with $|\,\mathcal{K}'\,|\leq|\,\mathcal{K}\,|$. (We abuse notation here: each $\mathcal{K}_i$ is a term, i.e. a process or a context. We may also omit sorts when they are not important.) From the typing rules, setting $\vec{u}_3=\emptyset$, in $\langle\Psi;\emptyset;\emptyset\vdash\mathcal{K}'\rangle$ the judgment for $\mathcal{K}_i$ is of the form

$$\Psi_i,\vec{x}_1,\vec{x}_2;\Delta_i;\Delta_i,\vec{w}_i:\circ\vdash\mathcal{K}_i\ \text{ where }\mathrm{n}(\Delta_i)=\vec{v}_i\text{ and }\vec{x}_i\vec{v}_i\vec{w}_i=\vec{u}_i.$$

By the inductive hypothesis there are $\mathcal{C}_1,\mathcal{C}_2,\mathcal{C}_3$ such that $\Psi_i,\vec{x}_1,\vec{x}_2\vdash\mathcal{C}_i$ assuming $\Psi_0\vdash[\cdot]$, and if $\Psi_0;\Delta_0;\Delta_0,\vec{w}_0:\circ\vdash M$ then $\nu\vec{v}_i\vec{w}_i\,\mathcal{K}_i[M]\approx^m_\lambda[\![\mathcal{C}_i]\!][\nu\vec{v}_0\vec{w}_0\,M]$. Then set

$$\mathcal{C}=x(z_1z_2).\,\nu\vec{x}_1\vec{x}_2\,(\mathcal{C}_3\mid\mathcal{C}_2\mid\mathcal{C}_1).$$

We show that for every M such that $\Psi_0;\Delta_0;\Delta_0,\vec{w}_0:\circ\vdash M$,

$$\mathcal{K}'[M]\approx^m_\lambda[\![\mathcal{C}]\!][\nu\vec{v}_0\vec{w}_0\,M].\tag{11}$$

As an aside, note here the fundamental role of typing in the inductive argument. It is not in general the case that $\mathcal{K}'[M]\approx^m[\![\mathcal{C}]\!][\nu\vec{v}_0\vec{w}_0\,M]$. For example if $\nu\vec{u}_1\,\mathcal{K}_1\downarrow_\mu$ for some μ then $\mathcal{H}'[\mathcal{K}'[M]]\not\dot{\approx}\,\mathcal{H}'[[\![\mathcal{C}]\!][\nu\vec{v}_0\vec{w}_0\,M]]$ where $\mathcal{H}'$ is $[\cdot]\mid\nu w\,\overline{x}w.\,\mathbf{0}$.

Returning to the main proof, if $\Psi_0;\Delta_0;\Delta_0,\vec{w}_0:\circ\vdash M$ then $\mathcal{K}''[M]\approx^m_\lambda[\![\mathcal{C}]\!][\nu\vec{v}_0\vec{w}_0\,M]$ where

$$\mathcal{K}''=x(z).\,z(z_1).\,z(z_2).\,\nu\vec{u}_1\vec{u}_2\,(\mathcal{K}_3\mid\mathcal{K}_2\mid\mathcal{K}_1).$$

So assertion (11) follows from

$$\mathcal{K}'[M]\approx^m_\lambda\mathcal{K}''[M].$$

This in turn follows from

Lemma 45 Suppose $\Psi;\emptyset;\emptyset\vdash K',K''$ where

$$\begin{aligned}K' &= x(z).\,\nu\vec{u}_1\,(z(z_1).\,\nu\vec{u}_2\,(z(z_2).\,K_3\mid K_2)\mid K_1)\\ K'' &= x(z).\,z(z_1).\,z(z_2).\,\nu\vec{u}_1\vec{u}_2\,(K_3\mid K_2\mid K_1).\end{aligned}$$

Then $K' \approx_\lambda^m K''$.

Proof: Let $\Theta = \{\theta \mid \theta \text{ respects } \Psi\}$. We show that

$$\mathcal{R} = \{(\mathcal{H}[K'\theta], \mathcal{H}[K''\theta]) \mid \ \theta \in \Theta \text{ and } \mathcal{H} \text{ is a } \lambda^m(\Psi\theta, \emptyset, \emptyset)\text{-context}\} \cup \dot{\approx}$$

is a barbed bisimulation up to $\approx^m$. Then for any $\lambda^m(\Psi, \emptyset, \emptyset)$-context $\mathcal{H}$ we have $\mathcal{H}[K'] \dot{\approx} \mathcal{H}[K'']$, and hence $K' \approx_\lambda^m K''$.

Let $(\mathcal{H}[K'\theta], \mathcal{H}[K''\theta]) \in \mathcal{R}$. The two processes $\mathcal{H}[K'\theta]$ and $\mathcal{H}[K''\theta]$ have the same observables. Assume now that $\mathcal{H}[K'\theta] \longrightarrow X$. Then from Lemma 5:

$$\mathcal{H}[K'\theta] \equiv \nu\vec{w}\,(\overline{z}y.\, T_1 \mid z(u).\, T_2 \mid T_3) \longrightarrow X \equiv \nu\vec{w}\,(T_1 \mid T_2\{y/u\} \mid T_3) \tag{12}$$

From this and definition of K', K'' and $\equiv$, we have:

$$\mathcal{H}[K''\theta] \equiv \nu\vec{w}\,(\overline{z}y.\, T_1' \mid z(u).\, T_2' \mid T_3') \longrightarrow Y \equiv \nu\vec{w}\,(T_1' \mid T_2'\{y/u\} \mid T_3') \tag{13}$$

where $T_j' = T_j$ if T_j does not contain a copy of $K'\theta$ which comes from filling the hole in $\mathcal{H}$. Note that by definition of $\lambda^m(\Psi\theta, \emptyset, \emptyset)$-context, there exist Ψ', Ψ'' such that $\Psi'; \emptyset; \emptyset \vdash \mathcal{H}[K'\theta]$ and $\Psi''; \emptyset; \emptyset \vdash \mathcal{H}[K''\theta]$. From this and Lemma 36, it follows that:

$$\Psi'; \emptyset; \emptyset \vdash X \qquad\qquad \Psi''; \emptyset; \emptyset \vdash Y \tag{14}$$

We distinguish two cases:

[CASE 1] If the intraaction does not involve a copy of $K'\theta$ which comes from filling the hole in $\mathcal{H}$, then we can further distinguish the following two subcases.

1. If T_2 does not contain a copy of $K'\theta$ then:

   $$X \equiv \mathcal{H}'[K'\theta] \qquad\qquad Y \equiv \mathcal{H}'[K''\theta]$$

   and from (14) $\mathcal{H}'$ is a $\lambda^m(\Psi\theta, \emptyset, \emptyset)$-context. Hence $(X, Y) \in \mathcal{R}$.

2. If T_2 does contain a copy of $K'\theta$ then:

   $$X \equiv \mathcal{H}'[K'\theta\{y/u\}] \qquad\qquad Y \equiv \mathcal{H}'[K''\theta\{y/u\}]$$

   We show that $\theta\{y/u\} \in \Theta$ by investigating on the typing of $(\overline{z}y.\, T_1 \mid z(u).\, T_2)$.

   - If the last rule applied in the typing judgment for $z(u).\, T_2$ is inp_1 then $\Psi_1; \{u : \sigma\}; \emptyset \vdash T_2$ for some Ψ_1 such that $u \notin \mathrm{n}(\Psi_1)$. Suppose that $u \in \mathrm{n}(\Psi\theta)$. From the non-homonymy condition no binder on u can occur in T_2. Then, since the only rules that can decrement the size of the first component of a typing judgment are inp_2 and res_1, the contradiction $u \in \mathrm{n}(\Psi_1)$ follows. Hence $u \notin \mathrm{n}(\Psi\theta)$, namely $u \notin \mathsf{cosupp}(\theta)$ and either $u \notin \mathrm{n}(\Psi)$ or ($u \in \mathrm{n}(\Psi)$ and $\theta = \theta'\{w/u\}$), i.e. $\theta\{y/u\} = \theta$. In each case $\theta\{y/u\} \in \Theta$.
   - If the last rule applied in the typing judgment for $z(u).\, T_2$ is inp_2 then the last rule applied to type $\overline{z}y.\, T_1$ is out_2 and hence $\Psi_1, u : s; \{z : \tau\}; \emptyset \vdash T_2$ for some Ψ_1 such that $\Psi_1(y) \simeq s$. Then $u : s \in \Psi\theta$ and $\Psi\theta(y) \simeq s$. Hence the thesis simply noting that ($w : s \in \Psi\theta$ implies $\Psi(w) \simeq s$).

From $\theta\{y/u\} \in \Theta$ and Lemma 34 we have: $\Psi\theta\{y/u\}; \emptyset; \emptyset \vdash K'\theta\{y/u\}, K''\theta\{y/u\}$. Then $\mathcal{H}'$ is a $\lambda^m(\Psi\theta\{y/u\}, \emptyset, \emptyset)$-context from (14), and hence $(X, Y) \in \mathcal{R}$.

[CASE 2] If the intraaction does involve a copy of $K'\theta$ which comes from filling the hole then in (12) and (13) the following holds: $z(u).\, T_2 \equiv K'\theta$ and $z(u).\, T_2' \equiv K''\theta$ and $T_1 = T_1'$.

Note that, since $\Psi\theta; \emptyset; \emptyset \vdash K'\theta$, the typing judgment for $\overline{z}y.\, T_1$ must have been inferred from out_1, namely:

$$\Psi\theta; \emptyset; \{y : \circ\} \vdash \overline{z}y.\, T_1 \qquad\qquad \Psi\theta; \emptyset; \{y : \sigma\} \vdash T_1$$

with $\circ \xrightarrow{s} \sigma$ and $z : s \in \Psi\theta$. From $\Psi\theta; \emptyset; \{y : \circ\} \vdash \overline{z}y.\, T_1$ and $\Psi'; \emptyset; \emptyset \vdash \mathcal{H}[K'\theta]$ it follows that $y \in \vec{w}$. Also, from $\Psi\theta; \emptyset; \{y : \sigma\} \vdash T_1$ and applying twice Lemma 29 we have that, in the most complicated of the four possible cases, $T_1 \equiv \nu\vec{w}_1\,(\overline{y}x_1.\, \nu\vec{w}_2\,(\overline{y}x_2.\, N_3 \mid N_2) \mid N_1)$ with $y \notin \vec{w}_1, \vec{w}_2$. Hence:

$$\begin{aligned} X \;\equiv\; \nu\vec{w}\,(\;& \nu\vec{w}_1\,(\overline{y}x_1.\, \nu\vec{w}_2\,(\overline{y}x_2.\, N_3 \mid N_2) \mid N_1) \mid \\ & \nu\vec{u}_1\,(y(z_1).\, \nu\vec{u}_2\,(y(z_2).\, K_3\theta' \mid K_2\theta') \mid K_1\theta') \mid \\ & T_3 \quad) \end{aligned}$$

where $\theta' = \theta\{y/u\}$ and where $z_2 \notin \mathrm{fn}(K_2\theta')$ and $z_1, z_2, u_2 \notin \mathrm{fn}(K_1\theta')$ from the non-homonymy condition. Analogously:

$$\begin{aligned} Y \;\equiv\; \nu\vec{w}\,(\;& \nu\vec{w}_1\,(\overline{y}x_1.\, \nu\vec{w}_2\,(\overline{y}x_2.\, N_3 \mid N_2) \mid N_1) \mid \\ & y(z_1).\, y(z_2).\, \nu\vec{u}_1\vec{u}_2\,(K_3\theta' \mid K_2\theta' \mid K_1\theta') \mid \\ & T_3' \quad) \end{aligned}$$

Then, letting $\theta_1 = \{x_1/z_1\}$ and $\theta_2 = \{x_1x_2/z_1z_2\}$, we have:

$$\begin{aligned} X \;\longrightarrow^2\; X'' \;\equiv\; \nu\vec{w}\,(\;& \nu\vec{w}_1\,(\nu\vec{w}_2\,(N_3 \mid N_2) \mid N_1) \mid \\ & \nu\vec{u}_1\,(\nu\vec{u}_2\,(K_3\theta'\theta_2 \mid K_2\theta'\theta_1) \mid K_1\theta') \mid \\ & T_3 \quad) \\ Y \;\longrightarrow^2\; Y'' \;\equiv\; \nu\vec{w}\,(\;& \nu\vec{w}_1\,(\nu\vec{w}_2\,(N_3 \mid N_2) \mid N_1) \mid \\ & \nu\vec{u}_1\vec{u}_2\,(K_3\theta'\theta_2 \mid K_2\theta'\theta_2 \mid K_1\theta'\theta_2) \mid \\ & T_3' \quad) \end{aligned}$$

Noting that $X \approx^m X''$ by $y \in \vec{w}$, we can conclude the proof by the following argument. If the hole in $\mathcal{H}$ is not underneath a replication then $z(u).\, T_2 \equiv K'\theta$ and $z(u).\, T_2' \equiv K''\theta$ are the only copies of $K'\theta$ and $K''\theta$ which come from filling the hole in $\mathcal{H}$. Then $T_3 \equiv T_3'$ and hence:

$$X \approx^m X'' \equiv Y''.$$

If the hole in $\mathcal{H}$ is underneath a replication then X'' and Y'' differ in that the used $K'\theta$ and $K''\theta$ are expanded copies of replications that still occur in T_3 and in T_3' respectively. Then $X'' \equiv \mathcal{H}'[K'\theta]$ and $Y'' \equiv \mathcal{H}'[K''\theta]$ where, from (14) and Lemma 36, $\mathcal{H}'$ is a $\lambda^m(\Psi\theta, \emptyset, \emptyset)$-context. Then:

$$X \approx^m X'' \equiv \mathcal{H}'[K'\theta]\ \mathcal{R}\ \mathcal{H}'[K''\theta] \equiv Y''.$$

□

[*par*] $\mathcal{K} = \mathcal{K}_1 \mid \mathcal{K}_2$ and we can distinguish the following cases.

- If $\Delta = \emptyset$ then by typing

$$\Psi; \emptyset; \vec{w}_i : \circ \vdash \mathcal{K}_i \text{ where } \vec{w}_1\vec{w}_2 = \vec{w}.$$

Then by the inductive hypothesis there are $\mathcal{C}_1, \mathcal{C}_2$ such that $\Psi \vdash \mathcal{C}_i$ assuming $\Psi_0 \vdash [\cdot]$, and if $\Psi_0; \Delta_0; \Delta_0, \vec{w}_0 : \circ \vdash M$ then $\nu\vec{w}_i\, \mathcal{K}_i[M] \approx^m_\lambda [\![\mathcal{C}_i]\!][\nu\vec{v}_0\vec{w}_0\, M]$. Then the thesis, upon setting $\mathcal{C} = \mathcal{C}_1 \mid \mathcal{C}_2$.

- If $\Delta \neq \emptyset$ then $z : \sigma \in \Delta$ for some z. Then by Lemma 29 we have, in the most complicated case,

$$\mathcal{K} \equiv \mathcal{K}' = \nu\vec{u}\,(\overline{z}z_1.\,\mathcal{K}'_1 \mid z(z_2).\,\mathcal{K}'_2 \mid \mathcal{K}'_3)$$

with $|\mathcal{K}'| \leq |\mathcal{K}|$. From the typing rules we have

$$\Psi, \vec{x}_1; \Delta_1; \Delta_1, \vec{w}_1 : \circ, \vec{w} : \circ \vdash (\overline{z}z_1.\,\mathcal{K}'_1 \mid z(z_2).\,\mathcal{K}'_2 \mid \mathcal{K}'_3)$$

where $\mathrm{n}(\Delta_1) = \mathrm{n}(\Delta) \cup \{z, \vec{v}_1\}$ and $\vec{x}_1 z \vec{v}_1 \vec{w}_1 = \vec{u}$. Moreover:

$$\Psi, \vec{x}_1; \{z : \sigma\}; \{z : \sigma\} \vdash (\overline{z}z_1.\,\mathcal{K}'_1 \mid z(z_2).\,\mathcal{K}'_2) = \mathcal{K}''_1$$

$$\Psi, \vec{x}_1; \Delta, \vec{v}_1; \Delta, \vec{v}_1, \vec{w}_1 : \circ, \vec{w} : \circ \vdash \mathcal{K}_3 = \mathcal{K}''_2$$

Then by the inductive hypothesis there are $\mathcal{C}_1, \mathcal{C}_2$ such that $\Psi, \vec{x}_1 \vdash \mathcal{C}_i$ assuming $\Psi_0 \vdash [\cdot]$, and if $\Psi_0; \Delta_0; \Delta_0, \vec{w}_0 : \circ \vdash M$ then $\nu z\, \mathcal{K}''_1[M] \approx^m_\lambda [\![\mathcal{C}_1]\!][\nu\vec{v}_0\vec{w}_0\, M]$ and $\nu\vec{v}\vec{v}_1\vec{w}\vec{w}_1\mathcal{K}''_2[M] \approx^m_\lambda [\![\mathcal{C}_2]\!][\nu\vec{v}_0\vec{w}_0 M]$. Then the thesis, upon setting $\mathcal{C} = \nu\vec{x}_1(\mathcal{C}_1 \mid \mathcal{C}_2)$.

[res_1,res_2,res_3] $\mathcal{K} = \nu z\, \mathcal{K}_1$ and, depending on the typing rule under consideration, by the inductive hypothesis $\mathcal{K}_1$ is appropriately related to a certain $\mathcal{C}_1$. In each case the thesis is then an immediate consequence of the inductive hypothesis, setting $\mathcal{C} = \nu z\, \mathcal{C}_1$ if $\Psi, z : s; \Delta; \Delta, \vec{w} : \circ \vdash \mathcal{K}_1$ (rule res_1), and $\mathcal{C} = \mathcal{C}_1$ otherwise. □

This completes the proof of Theorem 43. □

A final remark: in the translation the clause for output prefix could alternatively be

$$[\![\overline{x}\langle a_1 \ldots a_n\rangle.\,Q]\!] = \nu w\, \overline{x}w.\,(\overline{w}a_1.\,\ldots.\,\overline{w}a_n.\,\mathbf{0} \mid [\![Q]\!]).$$

The same results hold in this case, the type system and Lemma 9 showing clearly why the two translations are in essence the same.

As mentioned in the Introduction, the work to which that presented here is most closely related is [Yos96]. There a notion of graph type for monadic processes is introduced and studied. Nodes of a graph type represent atomic actions, and edges an activation ordering between them. Among other results, a full abstraction theorem for the translation of polyadic processes to monadic processes is shown. The present paper is not, therefore, the first to prove such a result. We believe, however, that the approach introduced in this paper is considerably simpler and clearer. The type system is of a kind which is well understood, and its rules are very natural, given the idea of the graph $\mathcal{G}_\lambda$ arising from a polyadic sorting λ. We found the ability to argue by induction on type inference invaluable to the proof of full abstraction. We believe the techniques introduced here may be useful in other circumstances.

References

[Hon93] K. Honda. Types for Dyadic Interaction. In E. Best, editor, *Proc. 4th Conf. CONCUR*, volume 715 of *LNCS*, pages 509–523. Springer-Verlag, 1993.

[Kob97] N. Kobayashi. A partially deadlock-free typed process calculus. In *Proc. 12th IEEE Symp. LICS*, pages 128–139, 1997.

[KPT96] N. Kobayashi, B.C. Pierce, and D.N. Turner. Linearity and the Pi-Calculus. In *Proc. 23rd ACM Symp. POPL*, pages 358–371, 1996.

[LW95] X. Liu and D. Walker. A Polymorphic Type System for the Polyadic π-calculus. In I. Lee and S.A. Smolka, editors, *Proc. 6th Conf. CONCUR*, volume 962 of *LNCS*, pages 103–116. Springer-Verlag, 1995.

[Mil92] R. Milner. The Polyadic π-Calculus: a Tutorial. In F.L. Bauer, W. Brauer, and H. Schwichtenberg, editors, *Logic and Algebra of Specification*, pages 203–246. Springer-Verlag, 1992.

[MPW92] R. Milner, J. Parrow, and D. Walker. A Calculus of Mobile Processes, Part I and II. *Information and Computation*, 100(1):1–77, 1992.

[NS97] U. Nestmann and M. Steffen. Typing Confluence. In S. Gnesi and D. Latella, editors, *Proc. 2nd Int. ERCIM Workshop on Formal Methods in Industrial Critical Systems*, pages 77–101, 1997.

[PS93] B.C. Pierce and D. Sangiorgi. Typing and Subtyping for Mobile Processes. In *Proc. 8th IEEE Symp. LICS*, pages 376–385, 1993.

[PS97] B.C. Pierce and D. Sangiorgi. Behavioral Equivalence in the Polymorphic Pi-Calculus. In *Proc. 24th ACM Symp. POPL*, pages 242–255, 1997.

[San97] D. Sangiorgi. The name discipline of uniform receptiveness. In P. Degano, R. Gorrieri, and A. Marchetti-Spaccamela, editors, *Proc. 24th ICALP*, volume 1256 of *LNCS*, pages 303–313. Springer, 1997.

[Tur96] D.N. Turner. *The Polymorphic Pi-Calculus: Theory and Implementation.* PhD thesis, Comp. Sc. Dep., Edinburgh University, 1996. Report ECS-LFCS-96-345.

[VH93] V.T. Vasconcelos and K. Honda. Principal typing schemes in a polyadic π-calculus. In E. Best, editor, *Proc. 4th Conf. CONCUR*, volume 715 of *LNCS*. Springer-Verlag, 1993.

[Yos96] N. Yoshida. Graph Types for Monadic Mobile Processes. In *Proc. 16th Conf. FST&TCS*, volume 1180 of *LNCS*, pages 371–386. Springer, 1996.

Recent BRICS Report Series Publications

RS-98-26 Paola Quaglia and David Walker. *On Encoding $p\pi$ in $m\pi$*. October 1998. 27 pp. Full version of paper to appear in *Foundations of Software Technology and Theoretical Computer Science: 18th Conference*, FCT&TCS '98 Proceedings, LNCS, 1998.

RS-98-25 Devdatt P. Dubhashi. *Talagrand's Inequality in Hereditary Settings*. October 1998. 22 pp.

RS-98-24 Devdatt P. Dubhashi. *Talagrand's Inequality and Locality in Distributed Computing*. October 1998. 14 pp.

RS-98-23 Devdatt P. Dubhashi. *Martingales and Locality in Distributed Computing*. October 1998. 19 pp.

RS-98-22 Gian Luca Cattani, John Power, and Glynn Winskel. *A Categorical Axiomatics for Bisimulation*. September 1998. ii+21 pp. Appears in Sangiorgi and de Simone, editors, *Concurrency Theory: 9th International Conference*, CONCUR '98 Proceedings, LNCS 1466, 1998, pages 581–596.

RS-98-21 John Power, Gian Luca Cattani, and Glynn Winskel. *A Representation Result for Free Cocompletions*. September 1998. 16 pp.

RS-98-20 Søren Riis and Meera Sitharam. *Uniformly Generated Submodules of Permutation Modules*. September 1998. 35 pp.

RS-98-19 Søren Riis and Meera Sitharam. *Generating Hard Tautologies Using Predicate Logic and the Symmetric Group*. September 1998. 13 pp.

RS-98-18 Ulrich Kohlenbach. *Things that can and things that can't be done in PRA*. September 1998. 24 pp.

RS-98-17 Roberto Bruni, José Meseguer, Ugo Montanari, and Vladimiro Sassone. *A Comparison of Petri Net Semantics under the Collective Token Philosophy*. September 1998. 20 pp. To appear in *4th Asian Computing Science Conference*, ASIAN '98 Proceedings, LNCS, 1998.

RS-98-16 Stephen Alstrup, Thore Husfeldt, and Theis Rauhe. *Marked Ancestor Problems*. September 1998.