STUDIA UNIV. "BABEŞ-BOLYAI", INFORMATICA, Volume XLIV, Number 1, 1999

JC-NETS

MIHAI ROTARU

1. Introduction

By switching from input-boxing to output-boxing and by allowing sharing in the π-nets of Milner we obtain a formal model of concurrency, the jc-nets, which significantly reduces the original π-nets, while preserving the computational power. We use our model to provide a faithful graphical representation for a Turing complete fragment of the join-calculus of Fournet and Gonthier.

2. Control Structures

We review here the control structures as far as it is necessary for the following sections. The presentation of the control structures uses the notion of symmetric strict monoidal category. We suppose the reader has a little familiarity with this notion. However, since we work with the control structures in a rather algebraic than categorical style, no further knowledge of categories is needed.

Most of the ideas on the control structures we present in this subsection can be found in the introductory paper [MMP95]. The reader can also find there detailed poofs for all the background results we state in this review. A detailed account of the control structures may be found in [M96].

From an algebraic point of view, a control structure consists in a set of terms together with an equational theory and a reduction relation, called *reaction*, upon terms.

A control structure can be also seen as a symmetric strict monoidal category (ssmc for short) with additional structure. The ssmc morphisms, denoted by $a, b, c, \ldots$, correspond to the terms of the control structure (in the algebraic setting). They are called *actions*. The ssmc objects, denoted by $m, n, k, \ldots$, are called *arities*. The monoid of the arities $(M, \otimes, \epsilon)$ is assumed to be freely generated by a set P. The elements of P, denoted by $p, q, \ldots$, are called *prime* arities. If the arities m and n are the domain and respectively the codomain of a, we write $a : m \to n$; with some abuse of terminology, we say that a has the arity $m \to n$. We write $a \cdot b : m \to k$ for the ssmc composite of $a : m \to n$ and $b : n \to k$, and $a \otimes b : m \otimes k \to n \otimes l$ for the ssmc tensorial product of $a : m \to n$ and $b : k \to l$. By $\mathtt{id}_m : m \to m$ we denote the ssmc identities, and by $\mathtt{p}_{m,n} : m \otimes n \to n \otimes m$ the ssmc symmetries.

A control structure uses a denumerable set $\mathcal{X}$ of *names* denoted by $x, y, z, \ldots$, and is completely determined by several information as follows. To define the terms, it is needed a signature $(P, \mathcal{K})$, where P is the set of prime arities, and $\mathcal{K}$ is a set of *control* operators. Each name $x \in \mathcal{X}$ must be equipped with a prime $p \in P$, written $x : p$. A control is used to construct complex terms from more simple ones. Each control $K \in \mathcal{K}$ must be equipped with an *arity rule* of the form:

$$\frac{a_1 : m_1 \to n_1 \quad \ldots \quad a_r : m_r \to n_r}{K(a_1, \ldots, a_r) : m \to n} (\chi)$$

where χ may constrain the value of the integer r and the arities m_i, n_i, m, n. When fixed, r is called the *rank* of K. To define the reaction, it is needed a set of *reaction rules* R. A reaction rule is an ordered pair of terms having the same arity. The equational theory needs not specific information, it is common to all control structures.

Besides the ssmc and control operators, every control structure contains a *datum* operator $\langle x \rangle : \epsilon \to p$ where $x : p$, a *discard* operator $\omega_p : p \to \epsilon$, and an *abstractor* operator $\mathtt{ab}_x\, a : p \otimes m \to p \otimes n$ where $x : p$ and $a : m \to n$.

Notation We omit the arity subscripts when apparent. We suppose all terms used are well-formed, and all equations are between terms of the same arity.

Definition 2.1. *The* action *terms are constructed by the following grammar:*

$$a \ ::= \ \langle x \rangle \ \big| \ \omega_p \ \big| \ \mathtt{id}_m \ \big| \ \mathtt{p}_{m,n} \ \big| \ a \cdot b \ \big| \ a \otimes b \ \big| \ \mathtt{ab}_x\, a \ \big| \ K(a_1 \ldots a_r)$$

Definition 2.2. *The following derived operator is a new kind of abstractor*

$$(x)a \stackrel{def}{=} \mathtt{ab}_x\, a \cdot (\omega_p \otimes \mathtt{id}_n) \qquad\qquad (x : p, a : m \to n)$$

Definition 2.3. *The* equational theory *of a control structure is the congruence upon the terms generated by the ssmc axioms:*

$$\begin{aligned}
&a \cdot \mathtt{id}_n = a && (a : m \to n)\\
&\mathtt{id}_m \cdot a = a && (a : m \to n)\\
&(a \cdot b) \cdot c = a \cdot (b \cdot c)
\end{aligned}$$

$$\begin{aligned}
&a \otimes \mathtt{id}_\epsilon = a\\
&\mathtt{id}_\epsilon \otimes a = a\\
&(a \otimes b) \otimes c = a \otimes (b \otimes c)
\end{aligned}$$

$$\begin{aligned}
&\mathtt{id}_m \otimes \mathtt{id}_n = \mathtt{id}_{m \otimes n}\\
&(a \cdot b) \otimes (c \cdot d) = (a \otimes c) \cdot (b \otimes d)
\end{aligned}$$

$$\begin{aligned}
&\mathtt{p}_{m,n} \cdot \mathtt{p}_{n,m} = \mathtt{id}_{m \otimes n}\\
&\mathtt{p}_{k \otimes m,n} = (\mathtt{id}_k \otimes \mathtt{p}_{m,n}) \cdot (\mathtt{p}_{k,n} \otimes \mathtt{id}_m)\\
&\mathtt{p}_{m,n} \cdot (b \otimes a) = (a \otimes b) \cdot \mathtt{p}_{k,l} && (a : m \to k, b : n \to l)
\end{aligned}$$

together with the following axioms:

$$\begin{array}{ll}
\mathtt{ab}_x\,\langle x\rangle \cdot (\omega_p \otimes \mathtt{id}_p) = \mathtt{id}_p & (x:p)\\
\mathtt{ab}_x\,\langle y\rangle = \mathtt{id}_p \otimes \langle y\rangle & (x:p, y \neq x)\\
\mathtt{ab}_x\,\omega_q = \mathtt{id}_p \otimes \omega_q & (x:p)\\
\mathtt{ab}_x\,\mathtt{id}_m = \mathtt{id}_{p\otimes m} & (x:p)\\
\mathtt{ab}_x\,\mathtt{p}_{m,n} = \mathtt{id}_p \otimes \mathtt{p}_{m,n} & (x:p)\\
\mathtt{ab}_x\,(a\cdot b) = \mathtt{ab}_x\,a \cdot \mathtt{ab}_x\,b & \\
\mathtt{ab}_x\,(a \otimes \mathtt{id}_m) = \mathtt{ab}_x\,a \otimes \mathtt{id}_m & \\
\mathtt{ab}_x\,\mathtt{ab}_x\,a = \mathtt{id}_p \otimes \mathtt{ab}_x\,a & (x:p)\\
\mathtt{ab}_x\,\mathtt{ab}_y\,a \cdot (\mathtt{p}_{p,q} \otimes \mathtt{id}_n) = (\mathtt{p}_{p,q} \otimes \mathtt{id}_m) \cdot \mathtt{ab}_y\,\mathtt{ab}_x\,a & (x:p, y:q, y\neq x)\\
& \\
(\langle x\rangle \otimes \mathtt{id}_m)\cdot (x)a = a & (x:p, a: m\to n)\\
\langle x\rangle \cdot (y)(\langle y\rangle \otimes \langle y\rangle) = \langle x\rangle \otimes \langle x\rangle & (x,y:p)\\
(\langle x\rangle \otimes \mathtt{id}_m)\cdot (y)K(a_1,\ldots) = K(((\langle x\rangle \otimes \mathtt{id}_{m_1})\cdot (y)a_1,\ldots) & (x,y:p)
\end{array}$$

Notation We use the equality $=$ between two actions a and b, if the equation $a = b$ can be proved using the above axioms together with the rules of a congruence. Otherwise, we write $a \neq b$.

Definition 2.4. *Each action a possesses a* surface *defined to be the following set*

$$\mathtt{surf}(a) = \{x \in X \mid \mathtt{ab}_x\,a \neq \mathtt{id} \otimes a\}$$

Proposition 2.1. *We have the following properties of the surface:*

1. $\mathtt{surf}(\langle x\rangle) \subseteq \{x\}$
2. $\mathtt{surf}(\omega) = \emptyset$
3. $\mathtt{surf}(\mathtt{id}) = \emptyset$
4. $\mathtt{surf}(\mathtt{p}) = \emptyset$
5. $\mathtt{surf}(a\cdot b) \subseteq \mathtt{surf}(a) \cup \mathtt{surf}(b)$
6. $\mathtt{surf}(a\otimes b) \subseteq \mathtt{surf}(a) \cup \mathtt{surf}(b)$
7. $\mathtt{surf}(\mathtt{ab}_x\,a) = \mathtt{surf}(a) - \{x\}$
8. $\mathtt{surf}(K(a_1,\ldots,a_r)) \subseteq \mathtt{surf}(a_1) \cup \ldots \cup \mathtt{surf}(a_r)$

Definition 2.5. *We define now second derived operator by*

$$[x/y]a \stackrel{def}{=} (\langle x\rangle \otimes \mathtt{id}_m)\cdot (y)a \qquad (x,y:p, a:m\to n)$$

The next two lemmas motivate the substitution-like notation chosen for this derived operator. The first shows a kind of α-conversion, and the second shows properties verified by any standard capture-avoiding substitution (with the surface understood as the set of the free names). Note that, with this abbreviation, the last tree axioms in definition 2.3 are simple substitution properties.

Proposition 2.2. *If $x \notin \mathtt{surf}(a)$ then $(y)a = (x)[x/y]a$.*

Proposition 2.3. *The following are provable in any control structure:*

1. $[x/y]\langle y\rangle = \langle x\rangle$
2. $[x/y]\langle z\rangle = \langle z\rangle$ *if* $z \neq y$
3. $[x/y]\omega = \omega$
4. $[x/y]\mathtt{id} = \mathtt{id}$
5. $[x/y]\mathtt{p} = \mathtt{p}$
6. $[x/y](a \cdot b) = [x/y]a \cdot [x/y]b$
7. $[x/y](a \otimes b) = [x/y]a \otimes [x/y]b$
8. $[x/y](z)a = (z)[x/y]a$ *if* $z \notin \{x, y\}$
9. $[x/y](y)a = (y)a$
10. $[x/y](x)a = (w)[x/y][w/x]a$ *if* $w \notin \mathtt{surf}(a) \cup \{x, y\}$ *and* $x \neq y$
11. $[x/y]K(a_1, \ldots) = K([x/y]a_1, \ldots)$

Proposition 2.4. *The following hold in any control structure whenever* $x \notin \mathtt{surf}(c)$*:*

1. $\mathtt{p}_{n,\epsilon} = \mathtt{id}_n$
2. $a \otimes b = a \otimes b$ $(a, b : \epsilon \to \epsilon)$
3. $(x)(c \cdot b) = (\mathtt{id}_p \otimes c) \cdot (x)b$ $(x : p)$
4. $(x)(c \otimes b) = c \otimes (x)b$ $(c : \epsilon \to n)$
5. $(x)(a \otimes c) = (x)a \otimes c$
6. $(z)(y)a = (\mathtt{p}_{p,q} \otimes \mathtt{id}_m) \cdot (y)(z)a$ $(z : p, y : q, a : m \to n)$

Definition 2.6. Reaction $\searrow$ *is defined to be the smallest relation upon the actions which satisfies the reaction rules R, and is closed under composition, tensor, abstraction, and equality.*

3. JCNET: THE CONTROL STRUCTURE OF THE jc-NETS

3.1. **Hypergraphs.** In this section we recall the definition of the hypergraphs together with some standard related concepts such as isomorphism, graphical representation, and contraction on nodes or edges.

Definition 3.1 (Hypergraph). *A* rooted hypergraph *is a tuple* $H = \langle S, V, E, s\rangle$ *where* S *is a set of* hyperedges, V *is a set of* vertices, $E \subseteq S \times V$ *is a relation, called the* incidence, *and* $s \in S$ *is the* root *hyperedge.*

The components of a hypergraph H are denoted by S_H, V_H, E_H, and s_H. If $(t, v) \in E_H$ for a hyperedge $t \in S_H$ and a vertex $v \in V_H$, we say "v lies on t".

The graphical representation of a hypergraph H is as follows. Hyperedges $t \in S_H$ are represented as unfilled ovals ○ with the name t outside. Vertices $v \in V_H$ are represented as points • carrying the name v. Incidences $(t, v) \in E_H$ are represented as tentacles from the oval named by t to the point named by v (the length of such a tentacle is often taken to be zero). Finally, the root is emphasized by a red flash pointing to the oval named by s_H.

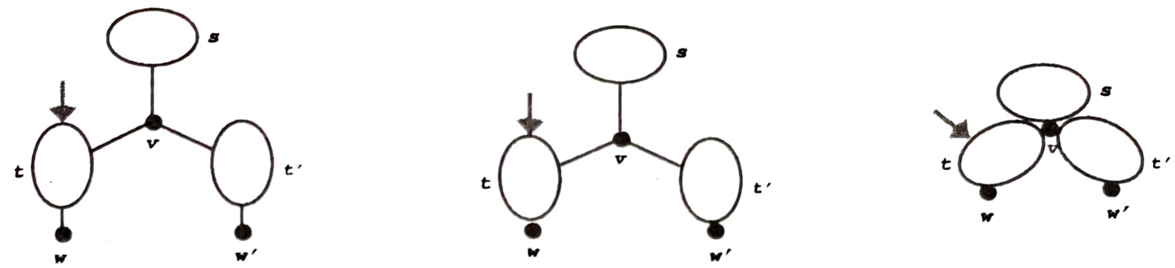

FIGURE 1. Example of a hypergraph.

Example 3.1. *We show in Fig. 1 the graphical representation of a hypergraph H. Here $S_H = \{s, t, t'\}$, $V_H = \{v, w, w'\}$, $E_H = \{(s,v), (t,v), (t,w), (t',v), (t',w')\}$, and $s_H = \{t\}$. The left picture represents H using only tentacles of nonzero length, and the right one only tentacles of zero length. We often use the representations as in the middle picture when one makes a compromise between size and clarity of visualization.*

Definition 3.2 (Contraction). *Let H be a rooted hypergraph. A nonempty subset of nodes $W \subseteq V_H$ gives rise to a* contraction on vertices *H/W which is the rooted hypergraph defined by*

- $S_{H/W} = S_H$
- $V_{H/W} = (V_H \setminus W) \dot\cup \{v\}$
- $E_{H/W} = (E_H \setminus S_H \times W) \cup \{\ (t,v) \mid \{t\} \times W \cap E_H \neq \emptyset\ \}$
- $s_{H/W} = s_H$

A nonempty subset of hyperedges $T \subseteq S_H$ gives rise to a contraction on hyperedges *H/T which is the rooted hypergraph defined by*

- $S_{H/T} = (S_H \setminus T) \dot\cup \{t\}$
- $V_{H/T} = V_H$
- $E_{H/T} = (E_H \setminus T \times V_H) \cup \{\ (t,v) \mid T \times \{v\} \cap E_H \neq \emptyset\ \}$
- $s_{H/T} = \underline{\texttt{if}}\ s_H \in T\ \underline{\texttt{then}}\ t\ \underline{\texttt{else}}\ s_H$

Given a hypergraph H, two vertices $v, w \in V_H$, and two hyperedges $s, t \in S_H$, we write $H_{v=w}$ for the contraction on vertices $H/\{v, w\}$, and $H_{s=t}$ for the contraction on hyperedges $H/\{s, t\}$.

Example 3.2. *If we take H to be the hypergraph used in Example 3.1 then we show in Fig. 2 a contraction on vertices $H_{w=w'}$ and a contraction on hyperedges $H_{t=t'}$.*

Definition 3.3 (Isomorphism). *Two hypergraphs H and H' are* isomorphic *if there exist two bijective functions $\phi_S : S_H \to S_{H'}$ and $\phi_V : V_H \to V_{H'}$ satisfying $\phi_S(s_H) = s_{H'}$ and for all $s \in S_H$ and $v \in V_H$, $(s, v) \in E_H$ if and only if $(\phi_S(s), \phi_V(v)) \in E_{H'}$.*

Of course, the relation of isomorphism is an equivalence over hypergraphs. Since for the present purpose, the names of hyperedges or vertices have no significance,

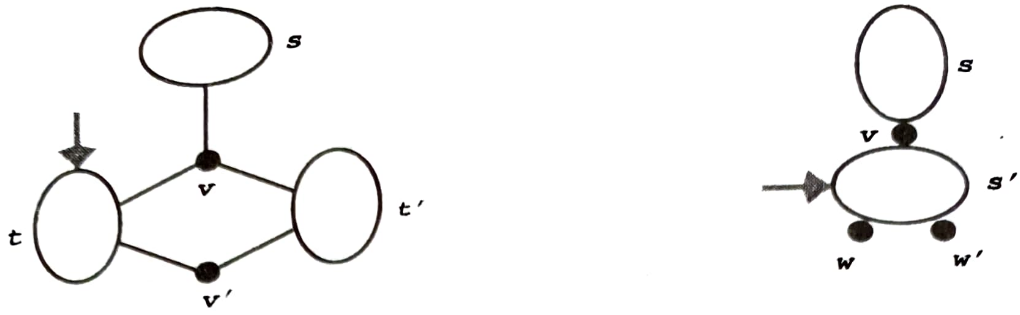

FIGURE 2. Example of contractions on vertices and hyperedges.

we do not distinguish between isomorphic hypergraphs. In accordance with this convention, if H and H' are isomorphic hypergraphs then we simply write $H = H'$. So we work with the equivalence classes of hypergraphs. For the equivalence classes of hypergraphs we use the same graphical representation as for hypergraphs; we have only to eliminate all names of vertices and of hyperedges.

3.2. **Actions.** In this section we define the control structures JCNET of the jc-nets. The arity monoid of JCNET is considered to be the additive monoid $(\mathbf{N}, +, 0)$ of the natural numbers. We let $m, n, k, \ldots$ range over natural numbers. By $[n]$ we denote the first n naturals, i.e. $[n] = \{1, 2, \ldots, n\}$. Given a function $f : [n] \to Y$ and a natural k, we denote by $k \oplus f : \{k+1, \ldots k+n\} \to Y$ the function defined by $(k \oplus f)(i) = f(i-k)$.

JCNET is defined over the set $X = \{z_i | i \in \mathbf{N}\}$ of names of the join-calculus. We let $x, y, u, \ldots$ range over arbitrary names. The unique prime arity 1 is associated with each name $x \in X$.

Actions of JCNET are enriched hypergraphs, called *nets*. An action $a = (H, \Sigma)$ of arity $m \to n$ consists of a hypergraph H together with decoration $\Sigma = \langle \mathtt{I}, \mathtt{O}, \lambda, \tau, \mu \rangle$ of H consisting in an injective function $\mathtt{I} : [m] \to V_H$, a function $\mathtt{O} : [n] \to V_H$, an injective function $\lambda : Z \to V_H$, where $Z \subseteq X$, a relation $\tau \subseteq V_H \times V_H$, and a function $\mu : S_H \to \mathbf{N}^{V_H \times V_H}$.

Concepts of isomorphism and contraction introduced for hypergraphs extend in a quite straightforward way to nets. Let $a_i = (H_i, \Sigma_i)$ with $\Sigma_i = \langle \mathtt{I}_i, \mathtt{O}_i, \lambda_i, \tau_i, \mu_i \rangle$, where $i \in [2]$. We say a_1 and a_2 are isomorphic if there exists a hypergraph isomorfism (ϕ_S, ϕ_V) between H_1 and H_2 such that $\phi_V \circ \mathtt{I}_1 = \mathtt{I}_2$, $\phi_V \circ \mathtt{O}_1 = \mathtt{O}_2$, $\phi_V \circ \lambda_1 = \lambda_2$ and for all $s \in S_{H_1}$ and $v, v' \in V_{H_1}$, $(v, v') \in \tau_1$ if and only if $(\phi_V(v), \phi_V(v')) \in \tau_2$ and $\mu_1(s, v, v') = \mu_2(\phi_S(s), \phi_V(v), \phi_V(v'))$. As for hypergraphs, we do not distinguish between isomorphic nets.

We now explain the graphical representations of nets. Let $a = (H, \Sigma)$ be a net with $\Sigma = \langle \mathtt{I}, \mathtt{O}, \lambda, \tau, \mu \rangle$. Represent the hypergraph H as explained in the previous section. Only for exposition, suppose all tentacles in H are of length zero. If $\mathtt{I}(i) = v$, $\mathtt{O}(k) = v'$, and $\lambda(x) = w$ then assign an *input label* (i) to the vertex named by v, an *output label* $\langle k \rangle$ to the vertex named by v, and a *name label* x to the vertex named by w. If $(v, v') \in \tau$ then draw an arc outside any oval from the vertex named by v to the vertex named by v'. If $\mu(s, v, v') = k$ then draw k arcs inside the oval named by s from the vertex named by v to the vertex named by v'. Note that, in general, the last requirement as arcs to be inside an oval might not be satisfied. Nevertheless, in the jc-nets this situation could not appear since $\mu(s, v, v') > 0$ implies $(s, v), (s, v') \in E_H$. As in the case of the hypergraphs, for the isomorphism classes of nets we use the same graphical representation as for nets; we have only to eliminate all names of vertices and of hyperedges.

Definition 3.4 (Datum). *The datum* $\langle x \rangle^\gamma = (H, \Sigma) : 0 \to 1$ *is defined by*

$$H = \langle \{s\}, \{v\}, \{(s, v)\}, s \rangle$$
$$\Sigma = \langle \emptyset, \{1 \mapsto v\}, \{x \mapsto v\}, \emptyset, \emptyset \rangle$$

Definition 3.5 (Discard). *The discard* $\omega^\gamma = (H, \Sigma) : 1 \to 0$ *is defined by*

$$H = \langle \{s\}, \{v\}, \{(s, v)\}, s \rangle$$
$$\Sigma = \langle \{1 \mapsto v\}, \emptyset, \emptyset, \emptyset, \emptyset \rangle$$

Definition 3.6 (Controls). JCNET *is generated by three control operators:*

- $\nu^\gamma = (H, \Sigma) : 0 \to 1$ *is defined by*

$$H = \langle \{s\}, \{v\}, \{(s, v)\}, s \rangle$$
$$\Sigma = \langle \emptyset, \{1 \mapsto v\}, \emptyset, \emptyset, \emptyset \rangle$$

- $\mathtt{out}^\gamma = (H, \Sigma) : 2 \to 0$ *is defined by*

$$H = \langle \{s\}, \{v, v'\}, \{(s, v), (s, v')\}, s \rangle$$
$$\Sigma = \langle \emptyset, \{1 \mapsto v, 2 \mapsto v'\}, \emptyset, \emptyset, \{(s, v', v)\} \rangle$$

- *If* $a = (H, \Sigma) : 1 \to 0$, $\Sigma = \langle \mathtt{I}, \mathtt{O}, \lambda, \tau, \mu \rangle$ *then* $\mathtt{def}^\gamma\, a = (H', \Sigma') : 1 \to 0$ *with*

$$H' = \langle S_H \dot{\cup} \{t\}, V_H \dot{\cup} \{v\}, E_H \cup \{(t, v)\}, t \rangle$$
$$\Sigma' = \langle \{1 \mapsto v\}, \mathtt{O}, \lambda, \tau \cup \{(v, \mathtt{I}(1))\}, \mu \rangle$$

Definition 3.7 (The ssmc operators). *Consider* $a_i = (H_i, \Sigma_i)$ *with* $\Sigma_i = \langle \mathtt{I}_i, \mathtt{O}_i, \lambda_i, \tau_i, \mu_i \rangle$ *and* $\lambda_i : Z_i \to V_{H_i}$, *where* $i \in [2]$. *W.l.o.g. we suppose* $s_{H_1} = s_{H_2} = s$ *and* $(S_{H_1} - \{s_{H_1}\}) \cap (S_{H_2} - \{s_{H_2}\}) = \emptyset$, *as well as* $\lambda_1(z) = \lambda_2(z)$, $\forall z \in Z_1 \cap Z_2$ *and* $(V_{H_1} - \lambda_1(Z_1 \cap Z_2)) \cap (V_{H_2} - \lambda_2(Z_1 \cap Z_2)) = \emptyset$.

- Identity $\mathtt{id}^\gamma_m = (H, \Sigma) : m \to m$ *is defined by*

$$H = \langle \{s\}, \{v_i | i \in [m]\}, \{(s, v_i) | i \in [m]\}, s \rangle$$
$$\Sigma = \langle \{i \mapsto v_i | i \in [m]\}, \{i \mapsto v_i | i \in [m]\}, \emptyset, \emptyset, \emptyset \rangle$$

- Symmetry $\mathtt{p}^\gamma_{m,n} = (H, \Sigma) : m + n \to n + m$ *is defined by*

$$H = \langle \{s\}, \{v_i | i \in [m + n]\}, \{(s, v_i) | i \in [m + n]\}, s \rangle$$
$$\Sigma = \langle \{i \mapsto v_i | i \in [m + n]\}, \{i \mapsto v_{m+i} | i \in [n]\} \cup \{n + i \mapsto v_i | i \in [m]\}, \emptyset, \emptyset, \emptyset \rangle$$

– Tensorial product $a_1 \otimes a_2 : m+k \to n+l$ *of* $a_1 : m \to n$ *and* $a_2 : k \to l$ *is obtained by combining* a_1 *and* a_2 *as follows. Increment with* m *the input labels, and with* n *the output labels in* a_2. *Contract the two roots, as well as vertices in* a_1 *and in* a_2 *bearing the same name label. Formally,* $a_1 \otimes a_2 = (H, \Sigma)$ *where*

$$H = \langle S_{H_1} \cup S_{H_2}, V_{H_1} \cup V_{H_2}, E_{H_1} \cup E_{H_2}, s\rangle$$
$$\Sigma = \langle\{\mathtt{I}_1 \cup m \oplus \mathtt{I}_2, \mathtt{O}_1 \cup n \oplus \mathtt{O}_2, \lambda_1 \cup \lambda_2, \tau_1 \cup \tau_2, \mu_1 \uplus \mu_2\rangle$$

– Composition $a_1 \cdot a_2 : m \to k$ *of* $a_1 : m \to n$ *and* $a_2 : n \to k$ *is obtained by combining* a_1 *and* a_2 *as follows. Contract the two roots, as well as vertices in* a_1 *and in* a_2 *bearing the same name label. For each* $i \in [n]$ *contract the vertex in* a_1 *output labeled by* $\langle i\rangle$ *with the vertex in* a_2 *input labeled by* (i). *Remove these labels* (i) *and* $\langle i\rangle$. *Formally,* $a_1 \cdot a_2 = (H, \Sigma)_{\mathtt{O}_1(1)=\mathtt{I}_2(1),\ldots,\mathtt{O}_1(n)=\mathtt{I}_2(n)}$ *where*

$$H = \langle S_{H_1} \cup S_{H_2}, V_{H_1} \cup V_{H_2}, E_{H_1} \cup E_{H_2}, s\rangle$$
$$\Sigma = \langle\{\mathtt{I}_1, \mathtt{O}_2, \lambda_1 \cup \lambda_2, \tau_1 \cup \tau_2, \mu_1 \uplus \mu_2\rangle$$

Definition 3.8 (Abstractor). *Let* $a = (H, \Sigma) : m \to n$ *with* $\Sigma = \langle \mathtt{I}, \mathtt{O}, \lambda, \tau, \mu\rangle$. *Then* $\mathtt{ab}_x^\gamma\, a : 1+m \to 1+n$ *is obtained from* a *as follows. Increment with* 1 *all input and output labels. Assign to the vertex labeled by* x *the input label* (1) *and the output label* $\langle 1\rangle$. *Remove the label* x. *Formally,* $\mathtt{ab}_x^\gamma\, a = (H, \Sigma')$ *where*

$$\Sigma' = \langle\{1 \mapsto \lambda(x)\} \cup 1 \oplus \mathtt{I}, \{1 \mapsto \lambda(x)\} \cup 1 \oplus \mathtt{O}, \lambda - \{x \mapsto \lambda(x)\}, \tau, \mu\rangle$$

One easily see that the above introduced operators over nets are well-defined, except the abstractor. Indeed, $\mathtt{ab}_x^\gamma\, a$ is not defined if the net a does not contain a vertex labeled by x. The completion we propose in the following definition is on one hand dictated by the need to have all operators well-defined. On the other, one may not have satisfied the axiom $(\langle x\rangle \otimes \mathtt{id}_m) \cdot (x)a = a$ of control structures which, in absence of a similar completion, could introduce new name labels.

Definition 3.9 (Completion). *We complete the definition of the above introduced operators, where* $\mathtt{op}$ *stands for each of these operators, as in the following*

$$\mathtt{op}(a, \ldots) \stackrel{def}{=} \mathtt{op}^\gamma(a \otimes^\gamma \mathbf{i}, \ldots) \otimes^\gamma \mathbf{i}$$

where $\mathbf{i} = (H, \Sigma)$ *is the net defined by*

$$H = \langle\{s\}, \{v_i | i \in \mathbf{N}\}, \{(s, v_i) | i \in \mathbf{N}\}, s\rangle$$
$$\Sigma = \langle\emptyset, \emptyset, \{z_i \mapsto v_i | i \in \mathbf{N}\}, \emptyset, \emptyset\rangle$$

Proposition 3.1. *The operators* $\langle x\rangle$, ω, ν, $\mathtt{out}$, $\mathtt{def}$, $\mathtt{id}$, $\mathtt{p}$, $\cdot$, $\otimes$, *and* $\mathtt{ab}_x$, *introduced in Definition 3.9, define a control structure which we call* JCNET.

Definition 3.10. *We define in the following two derived control operators:*

$$\mathtt{out}_u \stackrel{def}{=} (\langle u\rangle \otimes \mathtt{id}_1) \cdot \mathtt{out}$$
$$\mathtt{def}_u\, a \stackrel{def}{=} \langle u\rangle \cdot \mathtt{def}\, a$$

Lemma 3.1. *We have*

(1) $\texttt{surf}(\texttt{out}_u) \subseteq \{u\}$
(2) $\texttt{surf}(\texttt{def}_u\, a) \subseteq \{u\} \cup \texttt{surf}(a)$

Lemma 3.2. *Let $\sigma = \{x/y\}$ a name substitution. Then*
(1) $[x/y]\texttt{out}_u = \texttt{out}_{\sigma u}$
(2) $[x/y]\texttt{def}_u\, a = \texttt{def}_{\sigma u}\,[x/y]a.$

3.3. Reaction.

Definition 3.11 (Reaction). *The* reaction $\searrow$ *in* JCNET *is the smallest relation over* jc-*nets closed under tensorial product, composition, abstraction, and equality, which satisfies the following control rule*

$$\texttt{out}_u \otimes \texttt{def}_u\, a \searrow a \otimes \texttt{def}_u\, a$$

Lemma 3.3. *If $a \searrow b$ then there exists b' s.t. $b = b'$ and $\texttt{surf}(b') \subseteq \texttt{surf}(a)$.*

Lemma 3.4. *We have $\langle v\rangle \cdot \texttt{out}_w \otimes \texttt{def}_u\,(y)a \searrow b$ iff $u = w$ and $b = [v/y]a$.*

Lemma 3.5. *We have $a_1 \otimes a_2 \otimes a_3 \searrow c$ iff either there exists $i \in [3]$ s.t. $a_i \searrow b$ and $c = b \otimes a_j \otimes a_k$, or there exist $i, j \in [3]$ s.t. $a_i \otimes a_j \searrow b$ and $c = b \otimes a_k$ where $\{i, j, k\} = [3]$.*

Lemma 3.6. *If $u \notin \texttt{surf}(b)$, then $b \otimes \texttt{def}_u\, a \searrow c$ iff $b \searrow b'$ and $c = b' \otimes \texttt{def}_u\, a$.*

Lemma 3.7. *We have $\nu \cdot (x)a \searrow b$ iff $a \searrow a'$ and $b = \nu \cdot (x)a'$.*

4. Expressiveness of the jc-nets

4.1. The Join-Calculus.

In this subsection we recall the definition of the join-calculus [FG96]. A detailed account may be found in [Fou99]. Our presentation is based on [Lev98].

We consider a restriction of the join-calculus where only single input patterns and monadic messages are allowed.

Let $\mathcal{X}$ be an infinite countable set of *names*. We let $x, y, z, u, v, w, \ldots$ range over names. We denote by $\mathcal{P}$ the set of the join-calculus terms which are called processes. We let $P, Q, R, \ldots$ range over processes.

Definition 4.1. *The* processes *are constructed from the following grammar:*

P ::=		
$\mid$	0	empty process
$\mid$	$u\langle v\rangle$	message
$\mid$	$P \mid Q$	parallel composition
$\mid$	$\texttt{def}\ u\langle y\rangle \triangleright P\ \texttt{in}\ Q$	definition

In the above syntax, only the definition binds names. So u and y are considered to be bound. The scope of y is P, whereas the scope of u extends to the whole definition. α-convertibility among processes is defined in the standard way. We write $\{x/y\}P$ to denote the usual capture-avoiding name substitution.

Definition 4.2. *The* free names *of processes are defined inductively by:*

- $\mathtt{fn}(0) = \emptyset$
- $\mathtt{fn}(u\langle v\rangle) = \{u, v\}$
- $\mathtt{fn}(P \mid Q) = \mathtt{fn}(P) \cup \mathtt{fn}(Q)$
- $\mathtt{fn}(\mathtt{def}\ u\langle y\rangle \triangleright P\ \mathtt{in}\ Q) = (\mathtt{fn}(Q) \cup (\mathtt{fn}(P) - \{y\})) - \{u\}$

Definition 4.3. Structural congruence $\equiv\ \subseteq \mathcal{P} \times \mathcal{P}$ *is the smallest congruence relation which satisfies the following axioms:*

A1: $\mathtt{def}\ u\langle y\rangle \triangleright P\ \mathtt{in}\ Q \equiv \mathtt{def}\ u\langle t\rangle \triangleright \{t/y\}P\ \mathtt{in}\ Q$, *if* $t \notin \mathtt{fn}(P)$

A2: $\mathtt{def}\ u\langle y\rangle \triangleright P\ \mathtt{in}\ Q \equiv \mathtt{def}\ w\langle y\rangle \triangleright \{w/u\}P\ \mathtt{in}\ \{w/u\}Q$
if $y \notin \{u, w\}$ *and* $w \notin \mathtt{fn}(P) \cup \mathtt{fn}(Q)$

P1: $P \mid 0 \equiv P$
P2: $P \mid Q \equiv Q \mid P$
P3: $(P \mid Q) \mid R \equiv P \mid (Q \mid R)$

D1: $Q_1 \mid \mathtt{def}\ u\langle y\rangle \triangleright P\ \mathtt{in}\ Q_2 \equiv \mathtt{def}\ u\langle y\rangle \triangleright P\ \mathtt{in}\ (Q_1 \mid Q_2)$ *if* $u \notin \mathtt{fn}(Q_1)$

D2: $\mathtt{def}\ u\langle y\rangle \triangleright P_1\ \mathtt{in}\ \mathtt{def}\ w\langle t\rangle \triangleright P_2\ \mathtt{in}\ Q \equiv \mathtt{def}\ w\langle t\rangle \triangleright P_2\ \mathtt{in}\ \mathtt{def}\ u\langle y\rangle \triangleright P_1\ \mathtt{in}\ Q$
if $u \neq w$, $u \notin \mathtt{fn}(P_2)$, *and* $w \notin \mathtt{fn}(P_1)$

Lemma 4.1. $P \equiv Q$ *implies* $\sigma P \equiv \sigma Q$.

Definition 4.4. Reduction $\rightarrow\subseteq \mathcal{P} \times \mathcal{P}$ *is the smallest relation which satisfies:*

R1 $\mathtt{def}\ u_1\langle y_1\rangle \triangleright Q_1\ \mathtt{in}\ \mathtt{def}\ u_2\langle y_2\rangle \triangleright Q_2\ \mathtt{in}\ \ldots\ \mathtt{def}\ u_n\langle y_n\rangle \triangleright Q_n\ \mathtt{in}\ P \mid u_i\langle v\rangle \rightarrow$
$\mathtt{def}\ u_1\langle y_1\rangle \triangleright Q_1\ \mathtt{in}\ \mathtt{def}\ u_2\langle y_2\rangle \triangleright Q_2\ \mathtt{in}\ \ldots\ \mathtt{def}\ u_n\langle y_n\rangle \triangleright Q_n\ \mathtt{in}\ P \mid \{v/y_i\}Q_i$
if $\{u_{i+1}, \ldots, u_n\} \cap (\mathtt{fn}(Q_i) \cup \{u_i\}) = \emptyset$ *where* $i \in [n]$ *and* $n \geq 1$.

R2 $P_1 \rightarrow P_2$ *implies* $\mathtt{def}\ u\langle y\rangle \triangleright Q\ \mathtt{in}\ P_1 \rightarrow \mathtt{def}\ u\langle y\rangle \triangleright Q\ \mathtt{in}\ P_2$

R3 $P_1 \equiv Q_1$, $Q_1 \rightarrow Q_2$, *and* $Q_2 \equiv P_2$ *implies* $P_1 \rightarrow P_2$.

Remark that, in the definition of the reduction, we have not a rule for parallel composition. The next proposition shows that such a rule is a consequence. A preliminary lemma useful in the proof of the proposition is introduduced first.

Lemma 4.2. $P_1 \rightarrow P_2$ *implies* $\sigma P_1 \rightarrow \sigma P_2$.

Proposition 4.1. $P_1 \rightarrow P_2$ *implies* $Q \mid P_1 \rightarrow Q \mid P_2$.

4.2. Semantics of the join-calculus.

Definition 4.5. *The encoding* $[-] : \mathcal{P} \rightarrow \mathcal{T}(\nu, \mathtt{out}, \mathtt{def})$ *is defined by:*

(1) $[0] = \mathtt{id}_0$
(2) $[u\langle v\rangle] = \langle v\rangle \cdot \mathtt{out}_u$
(3) $[P \mid Q] = [P] \otimes [Q]$
(4) $[\mathtt{def}\ u\langle y\rangle \triangleright P\ \mathtt{in}\ Q] = \nu \cdot (u)([Q] \otimes \mathtt{def}_u\,(y)[P])$

Lemma 4.3. $[P] : \epsilon \rightarrow \epsilon$

Proof. A simple induction on the structure of P. □

Lemma 4.4. $\mathtt{fn}(P) \supseteq \mathtt{surf}([P])$.

Proof. Induction on the structure of P. The proof is imediate using Lemma 3.1. □

Lemma 4.5. $[\{x/y\}P] = [x/y][P]$

Proposition 4.2. *If $P \equiv Q$ then $[P] = [Q]$.*

Proof. Since both $\equiv$ and equality in CS are congruence relations, it will be sufficient to check the statement of the proposition only for the axioms in Definition 4.3 which generate $\equiv$. For P1 and P3, the result is trivial since the associativity of tensor $\otimes$ and the fact that $\mathtt{id}_0$ is neutral for tensor are assured by two axioms of CS. Likewise, for P2, the result follows directly by Proposition 2.4(2).

A1. Suppose $t \notin \mathtt{fn}(P)$. By Lemma 4.4, we have $t \notin \mathtt{surf}([P])$.

$$\begin{array}{lll} & [\mathtt{def}\ u\langle t\rangle \triangleright \{t/y\}P \ \mathtt{in}\ Q] & \\ = & \nu\cdot(u)([Q]\otimes \mathtt{def}_u\,(t)[\{t/y\}P]) & \text{by Lemma 4.5} \\ = & \nu\cdot(u)([Q]\otimes \mathtt{def}_u\,(t)[t/y][P]) & \text{by Lemma 2.2} \\ = & [\mathtt{def}\ u\langle y\rangle \triangleright P \ \mathtt{in}\ Q] & \end{array}$$

A2. Suppose $u \neq y$ and $w \notin \mathtt{fn}(P)\cup\mathtt{fn}(Q)\cup\{y\}$. By Lemma 4.4, we have $w \notin \mathtt{surf}([P])\cup\mathtt{surf}([Q])$. If $u = w$ then the result is trivial. Suppose $u \neq w$.

$$\begin{array}{lll} & [\mathtt{def}\ w\langle y\rangle \triangleright \{w/u\}P \ \mathtt{in}\ \{w/u\}Q] & \\ = & \nu\cdot(w)([\{w/u\}Q]\otimes \mathtt{def}_w\,(y)[\{w/u\}P]) & \text{by Lemma 4.5} \\ = & \nu\cdot(w)([w/u][Q]\otimes \mathtt{def}_w\,(y)[w/u][P]) & \text{by Prop. 2.3(8,7) and Lem. 3.2} \\ = & \nu\cdot(w)[w/u]([Q]\otimes \mathtt{def}_u\,(y)[P]) & \text{by Lemma 3.1 and Lemma 2.2} \\ = & [\mathtt{def}\ u\langle y\rangle \triangleright P \ \mathtt{in}\ Q] & \end{array}$$

D1. Suppose $u \notin \mathtt{fn}(Q_1)$. By Lemma 4.4, we have $u \notin \mathtt{surf}([Q_1])$.

$$\begin{array}{lll} & [\mathtt{def}\ u\langle y\rangle \triangleright P \ \mathtt{in}\ (Q_1 \mid Q_2)] & \\ = & \nu\cdot(u)([Q_1]\otimes[Q_2]\otimes \mathtt{def}_u\,(y)[P]) & \text{by Lemma 4.3 and Proposition 2.4(4)} \\ = & [Q_1 \mid \mathtt{def}\ u\langle y\rangle \triangleright P \ \mathtt{in}\ Q_2] & \end{array}$$

D2. Suppose $u \neq w$, $u \notin \mathtt{fn}(P_2)$, and $w \notin \mathtt{fn}(P_1)$. By Lemma 4.4, we have $u \notin \mathtt{surf}([P_2])$ and $w \notin \mathtt{surf}([P_1])$. Furthermore, by Lemma 3.1, $w \notin \mathtt{surf}(\mathtt{def}_u\,(y)[P_1]$.

$$\begin{array}{lll} & [\mathtt{def}\ u\langle y\rangle \triangleright P_1 \ \mathtt{in}\ \mathtt{def}\ w\langle t\rangle \triangleright P_2 \ \mathtt{in}\ Q] & \\ = & \nu\cdot(u)(\nu\cdot(w)([Q]\otimes \mathtt{def}_w\,(t)[P_2])\otimes \mathtt{def}_u\,(y)[P_1]) & \text{by Lemma 2.4(5)} \\ = & \nu\cdot(u)(\nu\cdot(w)([Q]\otimes \mathtt{def}_w\,(t)[P_2]\otimes \mathtt{def}_u\,(y)[P_1])) & \text{by Lemma 2.4(3)} \\ = & (\nu\otimes\nu)\cdot(u)(w)([Q]\otimes \mathtt{def}_w\,(t)[P_2]\otimes \mathtt{def}_u\,(y)[P_1]) & = \mathbf{X} \end{array}$$

Similarly we obtain

$$\begin{array}{lll} & [\mathtt{def}\ w\langle t\rangle \triangleright P_2 \ \mathtt{in}\ \mathtt{def}\ u\langle y\rangle \triangleright P_1 \ \mathtt{in}\ Q] & \\ = & (\nu\otimes\nu)\cdot(w)(u)([Q]\otimes \mathtt{def}_u\,(y)[P_1]\otimes \mathtt{def}_w\,(t)[P_2]) & = \mathbf{Y} \end{array}$$

Showing that $\mathbf{X} = \mathbf{Y}$ will complete the proof:

$$\begin{array}{lll} & \mathbf{X} & \text{by Lemma 2.4(6)} \\ = & (\nu\otimes\nu)\cdot \mathrm{p}_{1,1}\cdot(w)(u)([Q]\otimes \mathtt{def}_w\,(t)[P_2]\otimes \mathtt{def}_u\,(y)[P_1]) & \text{by Lemma 2.4(2)} \\ = & \mathrm{p}_{0,0}\cdot(\nu\otimes\nu)\cdot(w)(u)([Q]\otimes \mathtt{def}_u\,(y)[P_1]\otimes \mathtt{def}_w\,(t)[P_2]) & \text{by Lemma 2.4(1)} \\ = & \mathbf{Y} & \end{array}$$

□

Theorem 4.1. *$P \to Q$ implies $[P] \searrow [Q]$.*

Proof. By induction on the definition of $P \to Q$.

R1: $P \to Q$ is
$$\texttt{def } u_1\langle y_1\rangle \rhd Q_1 \texttt{ in def } u_2\langle y_2\rangle \rhd Q_2 \texttt{ in } \dots \texttt{ def } u_n\langle y_n\rangle \rhd Q_n \texttt{ in } R \mid u_i\langle v\rangle \to$$
$$\texttt{def } u_1\langle y_1\rangle \rhd Q_1 \texttt{ in def } u_2\langle y_2\rangle \rhd Q_2 \texttt{ in } \dots \texttt{ def } u_n\langle y_n\rangle \rhd Q_n \texttt{ in } R \mid \{v/y_i\}Q_i$$
where $\{u_{i+1},\dots,u_n\} \cap (\texttt{fn}(Q_i) \cup \{u_i\}) = \emptyset$, $i \in [n]$, and $n \geq 1$. By Lemma 3.1 and Lemma 4.4 one has $\{u_{i+1},\dots,u_n\} \cap \texttt{surf}(\texttt{def}_{u_i}(y_i)[Q_i]) = \emptyset$. Using this with Proposition 2.4, and also using the compatibility of $\searrow$ with composition, tensorial product, and abstraction, one has

$$
\begin{aligned}
& [P] \\
= \; & \nu \cdot (u_1)(\texttt{def}_{u_1}(y_1)[Q_1] \otimes \\
& \quad \vdots \\
& \quad \nu \cdot (u_i)(\texttt{def}_{u_i}(y_i)[Q_i] \otimes \\
& \qquad \vdots \\
& \qquad \nu \cdot (u_n)(\texttt{def}_{u_n}(y_n)[Q_n] \otimes [R] \otimes \langle v\rangle \cdot \texttt{out}_{u_i}) \dots) \dots) \\
= \; & \nu \cdot (u_1)(\texttt{def}_{u_1}(y_1)[Q_1] \otimes && \text{by Lemma 3.4} \\
& \quad \vdots \\
& \quad \nu \cdot (u_{i-1})(\texttt{def}_{u_{i-1}}(y_{i-1})[Q_{i-1}] \otimes \\
& \quad \nu \cdot (u_i)(\\
& \quad\; \nu \cdot (u_{i+1})(\texttt{def}_{u_{i+1}}(y_{i+1})[Q_{i+1}] \otimes \\
& \qquad \vdots \\
& \qquad \nu \cdot (u_n)(\texttt{def}_{u_n}(y_n)[Q_n] \otimes [R] \otimes \langle v\rangle \cdot \texttt{out}_{u_i} \otimes \texttt{def}_{u_i}(y_i)[Q_i]) \dots))) \dots) \\
\searrow \; & \nu \cdot (u_1)(\texttt{def}_{u_1}(y_1)[Q_1] \otimes && \text{by Lemma 4.5} \\
& \quad \vdots \\
& \quad \nu \cdot (u_i)(\texttt{def}_{u_i}(y_i)[Q_i] \otimes \\
& \qquad \vdots \\
& \qquad \nu \cdot (u_n)(\texttt{def}_{u_n}(y_n)[Q_n] \otimes [R] \otimes [v/y_i][Q_i]) \dots) \dots) \\
= \; & [Q]
\end{aligned}
$$

R2: $P \to Q$ is $\texttt{def } u\langle y\rangle \rhd R \texttt{ in } P' \to \texttt{def } u\langle y\rangle \rhd R \texttt{ in } Q'$ with $P' \to Q'$. By induction, $[P'] \searrow [Q']$. As $\searrow$ is closed under composition, tensor, and abstraction, we have $[P] = \nu \cdot (u)([P'] \otimes \texttt{def}_u(y)[R]) \searrow \nu \cdot (u)([Q'] \otimes \texttt{def}_u(y)[R]) = [Q]$.

R3: $P \to Q$ with $P \equiv P'$, $P' \to Q'$, and $Q' \equiv Q$. By induction, $[P'] \searrow [Q']$. By Proposition 4.2, $[P] = [P']$ and $[Q'] = [Q]$. Since $\searrow$ is closed under equality, it follows that $[P] \searrow [Q]$. □

Lemma 4.6. *$\langle v\rangle \cdot \texttt{out}_u \otimes [P] \searrow a$ iff $[P] \searrow b$ and $a = \langle v\rangle \cdot \texttt{out}_u \otimes b$.*

Proof. ($\Rightarrow$) Trivial as the reaction is closed under tensorial product and equality.

($\Leftarrow$) Induction on the structure of P.

– If P is the empty process or a message, then $\langle v\rangle \cdot \mathtt{out}_u \otimes [P] \not\searrow$. So, the statement of the lemma is obviously true as its premise is not satisfied.

– If P is a parallel composition $P_1 \mid P_2$, then $\langle v\rangle \cdot \mathtt{out}_u \otimes [P_1] \otimes [P_2] \searrow a$. As $\langle v\rangle \cdot \mathtt{out}_u \not\searrow$, it follows from Lemma 3.5 that one of the following cases remains possible:

(1) $[P_i] \searrow b'$ and $a = \langle v\rangle \cdot \mathtt{out}_u \otimes b' \otimes [P_j]$,
(2) $[P_i] \otimes [P_j] \searrow b'$ and $a = \langle v\rangle \cdot \mathtt{out}_u \otimes b'$, or
(3) $\langle v\rangle \cdot \mathtt{out}_u \otimes [P_i] \searrow a'$ and $a = a' \otimes [P_j]$,

where $\{i, j\} = [2]$. Note that by Proposition 2.4(2) one has $[P] = [P_i] \otimes [P_j]$. In case (1), one has $[P] \searrow b' \otimes [P_j]$. Take $b = b' \otimes [P_j]$. In case (2), take $b = b'$. In case (3), by induction, one has $[P_i] \searrow b'$ and $a' = \langle v\rangle \cdot \mathtt{out}_u \otimes b'$. So $[P] \searrow b' \otimes [P_j]$. Take $b = b' \otimes [P_j]$.

– If P is a definition $\mathtt{def}\ w\langle t\rangle \triangleright P_1\ \mathtt{in}\ P_2$, then without loss of generality, we assume that $w \notin \{u, v\}$. It follows from Lemma 3.1 together with Proposition 2.4(4) that $\nu \cdot (w)(\langle v\rangle \cdot \mathtt{out}_u \otimes [P_2] \otimes \mathtt{def}_w\,(t)[P_1]) \searrow a$. By Lemma 3.7, $\langle v\rangle \cdot \mathtt{out}_u \otimes [P_2] \otimes \mathtt{def}_w\,(t)[P_1] \searrow a'$ and $a = \nu \cdot (w)a'$. As $\langle v\rangle \cdot \mathtt{out}_u \not\searrow$, $\mathtt{def}_w\,(t)[P_1] \not\searrow$, and $\langle v\rangle \cdot \mathtt{out}_u \otimes \mathtt{def}_w\,(t)[P_1] \not\searrow$ (according to Lemma 3.4), it follows from Lemma 3.5 that one of the following cases remains possible:

(1) $[P_2] \searrow b'$ and $a' = \langle v\rangle \cdot \mathtt{out}_u \otimes b' \otimes \mathtt{def}_w\,(t)[P_1]$,
(2) $[P_2] \otimes \mathtt{def}_w\,(t)[P_1] \searrow b'$ and $a' = \langle v\rangle \cdot \mathtt{out}_u \otimes b'$, or
(3) $\langle v\rangle \cdot \mathtt{out}_u \otimes [P_2] \searrow a''$ and $a' = a'' \otimes \mathtt{def}_w\,(t)[P_1]$.

In case (1), one has $[P] = \nu \cdot (w)([P_2] \otimes \mathtt{def}_w\,(t)[P_1]) \searrow \nu \cdot (w)(b' \otimes \mathtt{def}_w\,(t)[P_1])$. Take $b = \nu \cdot (w)(b' \otimes \mathtt{def}_w\,(t)[P_1])$. In case (2), one has $[P] \searrow \nu \cdot (w)b'$. Take $b = \nu \cdot (w)b'$. In case (3), by induction, one has $[P_2] \searrow b'$ and $a'' = \langle v\rangle \cdot \mathtt{out}_u \otimes b'$. One has $[P] \searrow \nu \cdot (w)(b' \otimes \mathtt{def}_w\,(t)[P_1])$. Take $b = \nu \cdot (w)(b' \otimes \mathtt{def}_w\,(t)[P_1])$. □

Lemma 4.7. $[P] \otimes [Q] \searrow a$ *iff one of the following conditions holds:*

(1) $[P] \searrow b$ *and* $a = b \otimes [Q]$.
(2) $[Q] \searrow b$ *and* $a = [P] \otimes b$.

Proof. ($\Rightarrow$) Trivial as the reaction is closed under tensorial product and equality.
($\Leftarrow$) Induction on the structure of P.

– If P is the empty process 0 then *2.* obviously holds.

– If P is a message then *2.* holds by Lemma 4.6.

– If P is a parallel composition $P_1 \mid P_2$, then $[P_1] \otimes [P_2] \otimes [Q] \searrow a$. By Lemma 3.5 it follows that one of the following cases is possible:

(1) $[Q] \searrow b'$ and $a = [P_1] \otimes [P_2] \otimes b'$,
(2) $[P_i] \searrow b'$ and $a = b' \otimes [P_j] \otimes [Q]$,
(3) $[P_i] \otimes [P_j] \searrow b'$ and $a = b' \otimes [Q]$, or
(4) $[P_i] \otimes [Q] \searrow a'$ and $a = a' \otimes [P_j]$,

where $\{i, j\} = [2]$. Note that by Proposition 2.4(2) one has $[P] = [P_i] \otimes [P_j]$. In case (1), *2.* holds by taking $b = b'$. In case (2), one has $[P] \searrow b' \otimes [P_j]$. Then *1.*

holds by taking $b = b' \otimes [P_j]$. In case (3), *1.* holds by taking $b = b'$. In case (4), by induction, we distinguish two sub-cases. Then

(a) $[P_i] \searrow b'$ and $a' = b' \otimes [Q]$ or
(b) $[Q] \searrow b'$ and $a' = [P_i] \otimes b'$.

In sub-case (a), one has $[P] \searrow b' \otimes [P_j]$. Then *1.* holds by taking $b = b' \otimes [P_j]$. In sub-case (b), *2.* holds by taking $b = b'$. In both sub-cases we need some action commutations which are assured by Proposition 2.4(2).

– If P is a definition $\mathtt{def}\ w\langle t\rangle \triangleright P_1\ \mathtt{in}\ P_2$, then without loss of generality, we assume that $w \notin \mathtt{fn}(Q)$. By Lemma 4.4, $w \notin \mathtt{surf}([Q])$. It follows from Proposition 2.4(4) that $\nu\cdot(w)([P_2] \otimes \mathtt{def}_w\,(t)[P_1] \otimes [Q]) \searrow a$. By Lemma 3.7, $[P_2] \otimes \mathtt{def}_w\,(t)[P_1] \otimes [Q] \searrow a'$ and $a = \nu\cdot(w)a'$. As $\mathtt{def}_w\,(t)[P_1] \not\searrow$, it follows from Lemma 3.5 and Lemma 3.6 that one of the following cases remains possible:

(1) $[P_2] \searrow b'$ and $a' = b' \otimes \mathtt{def}_w\,(t)[P_1] \otimes [Q]$,
(2) $[Q] \searrow b'$ and $a' = [P_2] \otimes \mathtt{def}_w\,(t)[P_1] \otimes b'$,
(3) $[P_2] \otimes \mathtt{def}_w\,(t)[P_1] \searrow b'$ and $a' = b' \otimes [Q]$, or
(4) $[P_2] \otimes [Q] \searrow a''$ and $a' = a'' \otimes \mathtt{def}_w\,(t)[P_1]$.

In case (1), *1.* holds for $b = \nu\cdot(w)(b' \otimes \mathtt{def}_w\,(t)[P_1])$. In case (2), by Lemma 3.3, there is a b such that $\mathtt{surf}(b) \subseteq \mathtt{surf}([Q])$ and $b = b'$.Then *2.* holds for this b. In case (3), *1.* holds for $b = \nu\cdot(w)b'$. In case (4), by induction, we distinguish two sub-cases. Then

(a) $[P_2] \searrow b'$ and $a'' = b' \otimes [Q]$ or
(b) $[Q] \searrow b'$ and $a'' = [P_2] \otimes b'$. .

In sub-case (a), *1.* holds for $b = \nu\cdot(w)(b' \otimes \mathtt{def}_w\,(t)[P_1])$. In sub-case (b), by Lemma 3.3, there is a b such that $\mathtt{surf}(b) \subseteq \mathtt{surf}([Q])$ and $b = b'$. Then *2.* holds for this b. In all cases and sub-cases we eventually need to use Proposition 2.4.

□

Lemma 4.8. $[P] \otimes \mathtt{def}_u\,(y)[Q] \searrow a$ *iff one of the following conditions holds:*

(1) $[P] \searrow b$ *and* $a = b \otimes \mathtt{def}_u\,(y)[Q]$.
(2) $P \equiv R \mid u\langle v\rangle$ *and* $a = [R \mid \{v/y\}Q] \otimes \mathtt{def}_u\,(y)[Q]$.
(3) $P \equiv \mathtt{def}\ v_1\langle t_1\rangle \triangleright R_1\ \mathtt{in}\ \mathtt{def}\ v_2\langle t_2\rangle \triangleright R_2\ \mathtt{in}\ \ldots\ \mathtt{def}\ v_n\langle t_n\rangle \triangleright R_n\ \mathtt{in}\ (R \mid u\langle v_n\rangle)$,

$$\begin{array}{rcl} a & = & \nu\cdot(v_1)(\mathtt{def}_{v_1}\,(t_1)[R_1]\otimes \\ & & \quad \nu\cdot(v_2)(\mathtt{def}_{v_2}\,(t_2)[R_2]\otimes \\ & & \qquad \vdots \\ & & \qquad \nu\cdot(v_n)(\mathtt{def}_{v_n}\,(t_n)[R_n] \otimes [R \mid \{v_n/y\}Q] \otimes \mathtt{def}_u\,(y)[Q])\ldots)) \end{array}$$

where $v_i \notin \mathtt{fn}(Q) \cup \{u\}$ *for every* $i \in [n]$.

Proof. ($\Rightarrow$) If *1.* holds the proof is obvious. If *2.* holds then one has

$$\begin{array}{lll} & [P] \otimes \mathtt{def}_u\,(y)[Q] & \\ = & [R] \otimes \langle v\rangle\cdot\mathtt{out}_u \otimes \mathtt{def}_u\,(y)[Q] & \text{by Proposition 4.2} \\ \searrow & [R] \otimes [v/y][Q] \otimes \mathtt{def}_u\,(y)[Q] & \text{by Lemma 3.4} \\ = & a & \text{by Lemma 4.5} \end{array}$$

If *3.* holds then by Lemma 3.1 and Lemma 4.4 it follows that $v_i \notin \mathtt{surf}(\mathtt{def}_u\,(y)[Q])$ for every $i \in [n]$. On has

$$\begin{array}{ll} & [P] \otimes \mathtt{def}_u\,(y)[Q] \qquad \text{by Proposition 4.2 and Proposition 2.4(5)} \\ = & \nu \cdot (v_1)(\mathtt{def}_{v_1}\,(t_1)[R_1] \otimes \qquad \text{by Lemma 3.4} \\ & \quad \vdots \\ & \quad \nu \cdot (v_n)(\mathtt{def}_{v_n}\,(t_n)[R_n] \otimes [R] \otimes \langle v_n \rangle \cdot \mathtt{out}_u \otimes \mathtt{def}_u\,(y)[Q]) \dots) \\ \searrow & \nu \cdot (v_1)(\mathtt{def}_{v_1}\,(t_1)[R_1] \otimes \qquad \text{by Lemma 4.5} \\ & \quad \vdots \\ & \quad \nu \cdot (v_n)(\mathtt{def}_{v_n}\,(t_n)[R_n] \otimes [R] \otimes [v_n/y][Q] \otimes \mathtt{def}_u\,(y)[Q]) \dots) \\ = & a \end{array}$$

($\Leftarrow$) Induction on the structure of P.

– If P is the empty process 0 or a message $w\langle v \rangle$ with $w \neq u$ then $[P] \otimes \mathtt{def}_u\,(y)[Q] \not\searrow$. So, the statement of the lemma is obviously true as its premise is not satisfied. On the other hand, if P is a message $u\langle v \rangle$ then

$$\begin{array}{lll} & [P] \otimes \mathtt{def}_u\,(y)[Q] & \text{by Lemma 3.4} \\ \searrow & [v/y][Q] \otimes \mathtt{def}_u\,(y)[Q] \searrow a & \text{by Lemma 4.5} \\ = & [0 \mid \{v/y\}Q] \otimes \mathtt{def}_u\,(y)[Q] & \end{array}$$

Furthermore $P \equiv 0 \mid u\langle v \rangle$. So, *2.* holds.

– If P is a parallel composition $P_1 \mid P_2$, then $[P_1] \otimes [P_2] \otimes \mathtt{def}_u\,(y)[Q] \searrow a$. As $\mathtt{def}_u\,(y)[Q] \not\searrow$, it follows from Lemma 3.5 that one of the following cases remains possible:

(1) $[P_i] \searrow b'$ and $a = b' \otimes [P_j] \otimes \mathtt{def}_u\,(y)[Q]$,
(2) $[P_i] \otimes [P_j] \searrow b'$ and $a = b' \otimes \mathtt{def}_u\,(y)[Q]$, or
(3) $[P_i] \otimes \mathtt{def}_u\,(y)[Q] \searrow a'$ and $a = a' \otimes [P_j]$,

where $\{i, j\} = [2]$. Note that by Proposition 2.4(2) one has $[P] = [P_i] \otimes [P_j]$. In case (1), one has $[P] \searrow b' \otimes [P_j]$. So *1.* holds for $b = b' \otimes [P_j]$. In case (2), *1.* holds for $b = b'$. In case (3), by induction, we distinguish three sub-cases. Then

(*a*) $[P_i] \searrow b'$ and $a' = b' \otimes \mathtt{def}_u\,(y)[Q]$,
(*b*) $P_i \equiv R' \mid u\langle v \rangle$ and $a' = [R' \mid \{v/y\}Q] \otimes \mathtt{def}_u\,(y)[Q]$, or
(*c*) $P_i \equiv \mathtt{def}\ v_1\langle t_1 \rangle \triangleright R_1\ \mathtt{in}\ \dots\ \mathtt{def}\ v_n\langle t_n \rangle \triangleright R_n\ \mathtt{in}\ (R' \mid u\langle v_n \rangle)$ and

$$\begin{array}{ll} a' = & \nu \cdot (v_1)(\mathtt{def}_{v_1}\,(t_1)[R_1] \otimes \\ & \quad \vdots \\ & \quad \nu \cdot (v_n)(\mathtt{def}_{v_n}\,(t_n)[R_n] \otimes [R' \mid \{v_n/y\}Q] \otimes \mathtt{def}_u\,(y)[Q]) \dots) \end{array}$$

where $v_k \notin \mathtt{fn}(Q) \cup \{u\}$ for every $k \in [n]$.

In sub-case (*a*), one has $[P] \searrow b' \otimes [P_j]$. So *1.* holds for $b = b' \otimes [P_j]$. In sub-case (*b*), one has $P \equiv R' \mid P_j \mid u\langle v \rangle$. Furthermore by Proposition 2.4(2), $a = [R' \mid P_j \mid \{v/y\}Q] \otimes \mathtt{def}_u\,(y)[Q]$. So *2.* holds. In sub-case (*c*) we can assume without loss of generality that $v_k \notin \mathtt{fn}(P_j)$ for every $k \in [n]$. Then it follows from Lemma 4.4 that $v_k \notin \mathtt{surf}([P_j])$ for every $k \in [n]$. Then $P \equiv \mathtt{def}\ v_1\langle t_1\rangle \triangleright R_1\ \mathtt{in}\ \ldots\ \mathtt{def}\ v_n\langle t_n\rangle \triangleright R_n\ \mathtt{in}\ (R' \mid P_j \mid u\langle v_n\rangle)$. Furthermore

$$
\begin{aligned}
a & & \text{by Proposition 2.4(5),(2)}\\
= \ & \nu\cdot(v_1)(\mathtt{def}_{v_1}(t_1)[R_1] \otimes \\
& \vdots \\
& \quad \nu\cdot(v_n)(\mathtt{def}_{v_n}(t_n)[R_n] \otimes [R' \mid P_j \mid \{v_n/y\}Q] \otimes \mathtt{def}_u(y)[Q]) \ldots)
\end{aligned}
$$

So *3.* holds.

– If P is a definition $\mathtt{def}\ w\langle t\rangle \triangleright P_1\ \mathtt{in}\ P_2$ then without loss of generality we assume that $w \notin \mathtt{fn}(Q)\cup\{u\}$. By Lemma 4.4 and Lemma 3.1, $w \notin \mathtt{surf}(\mathtt{def}_u(y)[Q])$. It follows from Proposition 2.4(5) that $[P]\otimes\mathtt{def}_u(y)[Q] = \nu\cdot(w)([P_2]\otimes\mathtt{def}_w(t)[P_1]\otimes\mathtt{def}_u(y)[Q]) \searrow a$. By Lemma 3.7 one has that $[P_2]\otimes\mathtt{def}_w(t)[P_1]\otimes\mathtt{def}_u(y)[Q] \searrow a'$ and $a = \nu\cdot(w)a'$. As one has $\mathtt{def}_w(t)[P_1] \not\searrow$, $\mathtt{def}_u(y)[Q] \not\searrow$, and $\mathtt{def}_u(y)[Q]\otimes\mathtt{def}_w(t)[P_1] \not\searrow$ (according to Lemma 3.6), it follows from Lemma 3.4 that one of the following cases remains possible:

(1) $[P_2] \searrow b'$ and $a' = b' \otimes \mathtt{def}_w(t)[P_1] \otimes \mathtt{def}_u(y)[Q]$,
(2) $[P_2] \otimes \mathtt{def}_w(t)[P_1] \searrow b'$ and $a' = b' \otimes \mathtt{def}_u(y)[Q]$, or
(3) $[P_2] \otimes \mathtt{def}_u(y)[Q] \searrow a''$ and $a' = a'' \otimes \mathtt{def}_w(t)[P_1]$.

In case (1), *1.* holds for $b = \nu\cdot(w)(b' \otimes \mathtt{def}_w(t)[P_1])$. In case (2), *1.* holds for $b = \nu\cdot(w)b'$. In case (3), by induction, we distinguish three sub-cases. Then

(a) $[P_2] \searrow b'$ and $a'' = b' \otimes \mathtt{def}_u(y)[Q]$,
(b) $P_2 \equiv R' \mid u\langle v\rangle$ and $a'' = [R' \mid \{v/y\}Q] \otimes \mathtt{def}_u(y)[Q]$, or
(c) $P_2 \equiv \mathtt{def}\ v_1\langle t_1\rangle \triangleright R_1\ \mathtt{in}\ \ldots\ \mathtt{def}\ v_n\langle t_n\rangle \triangleright R_n\ \mathtt{in}\ (R' \mid u\langle v_n\rangle)$ and

$$
\begin{aligned}
a'' = \ & \nu\cdot(v_1)(\mathtt{def}_{v_1}(t_1)[R_1] \otimes \\
& \vdots \\
& \quad \nu\cdot(v_n)(\mathtt{def}_{v_n}(t_n)[R_n] \otimes [R' \mid \{v_n/y\}Q] \otimes \mathtt{def}_u(y)[Q]) \ldots)
\end{aligned}
$$

where $v_k \notin \mathtt{fn}(Q) \cup \{u\}$ for every $k \in [n]$.

In sub-case (a), *1.* holds for $b = \nu\cdot(w)(b' \otimes \mathtt{def}_w(t)[P_1])$. In sub-case (b), one has $P \equiv \mathtt{def}\ w\langle t\rangle \triangleright P_1\ \mathtt{in}\ (R' \mid u\langle v\rangle)$. We distinguish two situations:

(i) $v \neq w$. Then $P \equiv \mathtt{def}\ w\langle t\rangle \triangleright P_1\ \mathtt{in}\ R' \mid u\langle v\rangle$. It is easy to show that $\mathtt{surf}([\{v/y\}Q]) \subseteq \{v\} \cup \mathtt{surf}([Q])$. So $w \notin \mathtt{surf}([\{v/y\}Q])$. So by Proposition 2.4(2),(5) one has $a = [\mathtt{def}\ w\langle t\rangle \triangleright P_1\ \mathtt{in}\ R' \mid \{v/y\}Q] \otimes \mathtt{def}_u(y)[Q]$. So *2.* holds.

(ii) $v = w$. Then $P \equiv \mathtt{def}\ v\langle t\rangle \triangleright P_1\ \mathtt{in}\ (R' \mid u\langle v\rangle)$. Furthermore $a = \nu\cdot(v)(\mathtt{def}_v(t)[P_1] \otimes [R' \mid \{v/y\}Q] \otimes \mathtt{def}_u(y)[Q])$. So *3.* holds. In sub-case (c), $P \equiv \mathtt{def}\ w\langle t\rangle \triangleright P_1\ \mathtt{in}\ \mathtt{def}\ v_1\langle t_1\rangle \triangleright R_1\ \mathtt{in}\ \ldots\ \mathtt{def}\ v_n\langle t_n\rangle \triangleright R_n\ \mathtt{in}\ (R' \mid u\langle v_n\rangle)$. Using Proposition 2.4(2), one has

$$
\begin{aligned}
a = \ & \nu\cdot(w)(\mathtt{def}_w(t)[P_1] \otimes \\
& \quad \nu\cdot(v_1)(\mathtt{def}_{v_1}(t_1)[R_1] \otimes \\
& \quad \vdots \\
& \qquad \nu\cdot(v_n)(\mathtt{def}_{v_n}(t_n)[R_n] \otimes [R' \mid \{v_n/y\}Q] \otimes \mathtt{def}_u(y)[Q]) \ldots))
\end{aligned}
$$

So *3.* holds. □

Theorem 4.2. *$[P] \searrow a$ implies $P \to Q$ and $[Q] = a$.*

Proof. Induction on the structure of P.

– If P is the empty process or a message then $[P] \not\searrow$. So, the statement of the theorem is obviously true since the premise is not satisfied.

– If P is a parallel composition $P_1 \mid P_2$ then $[P_1] \otimes [P_2] \searrow a$. By Lemma 4.7 one of the following cases holds:

(1) $[P_1] \searrow a_1$ and $a = a_1 \otimes [P_2]$ or

(2) $[P_2] \searrow a_2$ and $a = [P_1] \otimes a_2$.

It is sufficient to consider the case (1), the other one being symmetric. By induction one has $P_1 \to Q_1$ and $a_1 = [Q_1]$.

By Proposition 4.1, $P \to \underbrace{Q_1 \mid P_2}_{Q}$ and $a = [Q_1] \otimes [P_2] = [Q]$.

– If P is a definition $\mathtt{def}\ w\langle t \rangle \triangleright P_1\ \mathtt{in}\ P_2$. Then $\nu \cdot (w)([P_2] \otimes \mathtt{def}_w\,(t)[P_1] \searrow a$. It follows from Lemma 3.7 that $[P_2] \otimes \mathtt{def}_w\,(t)[P_1] \searrow a'$ and $a = \nu \cdot (w)a'$. By Lemma 4.8, one of the following cases holds:

(1) $[P_2] \searrow b$ and $a' = b \otimes \mathtt{def}_w\,(t)[P_1]$,

(2) $P_2 \equiv R \mid w\langle v \rangle$ and $a' = [R \mid \{v/t\}Q] \otimes \mathtt{def}_w\,(t)[P_1]$, or

(3) $P_2 \equiv \mathtt{def}\ w_1\langle t_1 \rangle \triangleright R_1\ \mathtt{in}\ \ldots\ \mathtt{def}\ w_n\langle t_n \rangle \triangleright R_n\ \mathtt{in}\ (R \mid u\langle w_n \rangle)$ and

$$\begin{aligned} a' \;=\; & \nu \cdot (w_1)(\mathtt{def}_{w_1}\,(t_1)[R_1] \otimes \\ & \quad \vdots \\ & \qquad \nu \cdot (w_n)(\mathtt{def}_{w_n}\,(t_n)[R_n] \otimes [R \mid \{w_n/t\}P_1] \otimes \mathtt{def}_w\,(t)[P_1]) \ldots) \end{aligned}$$

where $w_i \notin \mathtt{fn}(P_1) \cup \{w\}$ for every $i \in [n]$.

In case (1), by induction, $P_2 \to Q_2$ and $b = [Q_2]$.

Then $P \to \underbrace{\mathtt{def}\ w\langle t \rangle \triangleright P_1\ \mathtt{in}\ Q_2}_{Q}$ and $a = \nu \cdot (w)([Q_2] \otimes \mathtt{def}_w\,(t)[P_1]) = [Q]$.

In case (2) it follows from Lemmas 3.1 and 4.4 that $w_i \notin \mathtt{surf}(\mathtt{def}_w\,(t)[P_1])$.

$P \equiv \mathtt{def}\ w\langle t \rangle \triangleright P_1\ \mathtt{in}\ \mathtt{def}\ w_1\langle t_1 \rangle \triangleright R_1\ \mathtt{in}\ \ldots\ \mathtt{def}\ w_n\langle t_n \rangle \triangleright R_n\ \mathtt{in}\ (R \mid u\langle w_n \rangle)$

$\to \underbrace{\mathtt{def}\ w\langle t \rangle \triangleright P_1\ \mathtt{in}\ \mathtt{def}\ w_1\langle t_1 \rangle \triangleright R_1\ \mathtt{in}\ \ldots\ \mathtt{def}\ w_n\langle t_n \rangle \triangleright R_n\ \mathtt{in}\ (R \mid \{w_n/t\}P_1)}_{Q}$

$$\begin{aligned} a \;=\; & \nu \cdot (w)(&& \text{by Proposition 2.4(5),(2)} \\ & \quad \nu \cdot (w_1)(\mathtt{def}_{w_1}\,(t_1)[R_1] \otimes \\ & \qquad \vdots \\ & \qquad \nu \cdot (w_n)(\mathtt{def}_{w_n}\,(t_n)[R_n] \otimes [R \mid \{w_n/t\}P_1] \otimes \mathtt{def}_w\,(t)[P_1]) \ldots)) \\ =\; & \nu \cdot (w)(\mathtt{def}_w\,(t)[P_1] \otimes \\ & \quad \nu \cdot (w_1)(\mathtt{def}_{w_1}\,(t_1)[R_1] \otimes \\ & \qquad \vdots \\ & \qquad \nu \cdot (w_n)(\mathtt{def}_{w_n}\,(t_n)[R_n] \otimes [R \mid \{w_n/t\}P_1]) \ldots)) \\ =\; & [Q] \end{aligned}$$

□

References

[AL96] A. Asperti, G. Longo. Categories, Types, and Structures. MIT Press, 1996.

[Bar85] H. P. Barendregt. The Lambda Calculus: Its Syntax and Semantics. North Holland, 1985.

[CR00] G. Ciobanu, M. Rotaru. A π-calculus Machine. In *Journal of Universal Computer Science*, Springer Verlag, Vol. 6, 2000.

[CR98] G. Ciobanu, M. Rotaru. Faithful π-nets. In *Electronic Notes of Theoretical Computer Science*, North-Holland, Vol. 18, 1998.

[Eng93] J. Engelfriet. A multiset semantics of the π-calculus with replication, in CONCUR'93, LNCS 715, 1993.

[FG96] C. Fournet, G. Gonthier. The Reflexive CHAM and the Join Calculus. In *Proc. of POPL'96*, ACM Press, 1996.

[Fou99] C. Fournet. The Join-Calculus: A calculus for Distributed Mobile Programming. PhD thesis, INRIA Rocquencourt, 1998.

[Lev98] J.J. Levy. Some results in the Join Calculus.

[MMP95] A. Mifsud, R. Milner, and A. J. Parrow. Control Structures. In *Proc. 10th Symposium on Logic in Computer Science (LICS'95)*, 1995.

[M96] Mifsud. PhD thesis, University of Edinburgh, 1996.

[Mil94] R. Milner. π-nets: a graphical form of π-calculus, ESOP'94, LNCS 788, Springer-Verlag, 1994.

[Mil96] R. Milner. Calculi for interaction. *Acta Informatica*, 33(8), 1996.

[SS93] G. Schmidt, T. Strohlein. *Relations and Graphs. Discrete Mathematics for Computer Scientists.* EATCS Monographs on Theoretical Computer Science, Springer-Verlag, 1993.

Institute of Theoretical Computer Science, Romanian Academy RO-6600 Iasi
E-mail address: mrotaru@iit.tuiasi.ro