axioms

MDPI

*Article*

# A Hypergraph Model for Communication Patterns

**Gabriel Ciobanu**

Faculty of Computer Science, Alexandru Ioan Cuza University, 700506 Iasi, Romania; gabriel@info.uaic.ro
† In Memoriam Solomon Marcus (1925–2016).

**Abstract:** The article deals with interaction in concurrent systems. A calculus able to express specific communication patterns is defined, together with its abstract control structures. A hypergraph model for these structures is presented. The hypergraphs are able to properly express the communication patterns, providing a fully abstract model for the pattern calculus. It is also proved that the hypergraph model preserves the operational reductions of processes from pattern calculus and of the actions from the control structures.

**Keywords:** process calculus; communication patterns; control structures; hypergraph model

## 1. Introduction

Mathematics has sometimes been called a 'science of patterns' [1], meaning that patterns are at the heart of mathematics. The nice description of mathematics as the "study of patterns" was given by G.H. Hardy in his book *A Mathematician's Apology*: "A mathematician, like a painter or a poet, is a maker of patterns. If his patterns are more permanent than theirs, it is because they are made with ideas". Essentially, patterns are regularities that we can perceive. Regarding the skilful ability of applying a pattern to multiple contexts, Solomon Marcus was a wonderful example of applying knowledge patterns to surprising contexts.

Since mathematics and technology have developed a fruitful relationship over past few decades, patterns have been investigated recently in modern fields. New communication technologies have changed the computing landscape, and the Internet is now a platform for large scale distributed programming. Nowadays, we deal with global computation based on multiple interactions with the environment (instead of isolated systems). Concurrency is essential now in computer science; web servers handle multiple simultaneous clients, and cloud servers allow several simultaneous applications and users. Message-passing represents a way in which concurrent processes communicate (a process is an instance of a running program). In software architecture, a messaging pattern describes how two different processes communicate with each other. In telecommunications, a message exchange pattern describes the messages required by a communications protocol or the message flow between parties involved in communication. For example, when navigating on Internet (representing the channel), a web browser (the communicating party) uses the communication protocol HTTP to request a web page from the server (another communicating party). In general, the interaction between clients and servers follows a specific communication pattern: the client sends a request, the server returns a response, and so on. Such an exchange of messages is only an example of communication patterns. More complicated behaviours appear due to the concurrent interaction of communicating processes; this complexity reveal the necessity to find new ways to describe and build up concurrent systems. The communicating parties in such a concurrent system should have a common language to communicate; moreover, they should follow the rules defined in a communications protocol. There exits already a calculus able to express communication patterns called join calculus; it is used as a basis for some programming languages (JoCaml and C$\omega$), but also as a basis for libraries (embedded in C#, F# and Scala). This calculus is based on 'join patterns', namely rules describing how a certain combination of values

check for updates

**Citation:** Ciobanu, G. A Hypergraph Model for Communication Patterns. *Axioms* **2022**, *11*, 8. https://doi.org/10.3390/axioms11010008

Academic Editor: Cristian S. Calude

Received: 30 October 2021
Accepted: 2 December 2021
Published: 23 December 2021

sent through multiple channels triggers a specific reaction and removes the values from the communication channels. Interaction in such a calculus is provided by sharing the communication channels names.

We introduce the pattern calculus as a weak version of join calculus. After presenting the control structures for pattern calculus, we provide a hypergraph model of these structures. The hypergraphs are able to express properly the communication patterns described by the pattern calculus. It is proved that the hypergraph model is fully abstract for the calculus; a model is fully abstract if all observationally equivalent processes represent the same object in the model, meaning also that processes with different behaviour are not mapped to the same hypergraph. Furthermore, there is a correspondence between the dynamics of the processes of the calculus and their hypergraph representation.

## 2. Pattern Calculus

The pattern calculus is inspired by the join calculus [2], a calculus proposed to underlie programming languages for distributed systems. A presentation of the join calculus can also be found in [3]. The specific construction of the new calculus is the definition of communication channels: $\texttt{def}\ u\langle y\rangle \triangleright P\ \texttt{in}\ Q$. To elucidate the simplicity of this syntactical construction, let us say that it could be expressed in the $\pi$-calculus [4] by using several syntactical constructions: $\texttt{def}\ u\langle y\rangle \triangleright P\ \texttt{in}\ Q = \nu u.(!u(y).P \mid Q)$.

The syntax of the pattern calculus is defined by using a countable set $\mathcal{X}$ of *names* ranging over $u, v, x, \ldots$, together with $\tilde{u}, \tilde{v}, \tilde{x}, \tilde{y} \ldots$ ranging over finite strings of names. We use $P, Q, R, \ldots$ ranging over the set of processes. The set $\mathcal{P}$ of processes contains an *empty process* $0$, as well as an *output message* $u\langle v\rangle$ sending $v$ by using a channel $u$. The process $P \mid Q$ describes the *parallel composition* of processes $P$ and $Q$. The communication between processes is achieved by the *channel definition* $\texttt{def}\ u\langle v\rangle \triangleright P\ \texttt{in}\ Q$ indicating that the interaction of processes $P$ and $Q$ is realized by the channel $u$ (which is created only for the communication between them).

**Definition 1.** *The processes of our calculus are defined by the following syntax:*

$$P \ ::= \ \ 0 \ \mid \ u\langle v\rangle \ \mid \ P \mid Q \ \mid \ \texttt{def}\ u\langle v\rangle \triangleright P\ \texttt{in}\ Q\,.$$

In $\texttt{def}\ u\langle v\rangle \triangleright P\ \texttt{in}\ Q$, both $u$ and $v$ are bound. The scope of $v$ is $P$, while the scope of $u$ is given by the whole definition. It is worth noting that only this definition binds names. The *free names* are defined inductively by $\texttt{fn}(0) = \emptyset$, $\texttt{fn}(u\langle v\rangle) = \{u, v\}$, $\texttt{fn}(P \mid Q) = \texttt{fn}(P) \cup \texttt{fn}(Q)$, and $\texttt{fn}(\texttt{def}\ u\langle v\rangle \triangleright P\ \texttt{in}\ Q) = (\texttt{fn}(Q) \cup (\texttt{fn}(P) - \{v\})) - \{u\}$.

A substitution $\{y/x\}P$ replaces all the free occurrences of name $x$ in $P$ by name $y$; name-capture is avoided by using the $\alpha$-conversion (defined in the standard way).

**Definition 2.** *A* structural congruence $\equiv \subseteq \mathcal{P} \times \mathcal{P}$ *is defined as the smallest congruence satisfying the following axioms:*

- $\texttt{def}\ u\langle v\rangle \triangleright P\ \texttt{in}\ Q \equiv \texttt{def}\ u\langle t\rangle \triangleright \{t/v\}P\ \texttt{in}\ Q$, *if* $t \notin \texttt{fn}(P)$
- $\texttt{def}\ u\langle v\rangle \triangleright P\ \texttt{in}\ Q \equiv \texttt{def}\ w\langle v\rangle \triangleright \{w/u\}P\ \texttt{in}\ \{w/u\}Q$ *if* $v \notin \{u, w\}$ *and* $w \notin \texttt{fn}(P \mid Q)$
- $Q_1 \mid \texttt{def}\ u\langle v\rangle \triangleright P\ \texttt{in}\ Q_2 \equiv \texttt{def}\ u\langle v\rangle \triangleright P\ \texttt{in}\ (Q_1 \mid Q_2)$ *if* $u \notin \texttt{fn}(Q_1)$
- $\texttt{def}\ u\langle v\rangle \triangleright P_1\ \texttt{in}\ \texttt{def}\ w\langle t\rangle \triangleright P_2\ \texttt{in}\ Q \equiv \texttt{def}\ w\langle t\rangle \triangleright P_2\ \texttt{in}\ \texttt{def}\ u\langle v\rangle \triangleright P_1\ \texttt{in}\ Q$
  *if* $u \neq w$, $u \notin \texttt{fn}(P_2)$, *and* $w \notin \texttt{fn}(P_1)$
- $P \mid 0 \equiv P \qquad P \mid Q \equiv Q \mid P \qquad (P \mid Q) \mid R \equiv P \mid (Q \mid R)$.

The evolution of the processes is given by a reduction relation $\rightarrow$.
For the specific construct $\texttt{def}\ u\langle y\rangle \triangleright P\ \texttt{in}\ Q$, the reduction is described mainly by:

$$\texttt{def}\ u\langle y\rangle \triangleright P\ \texttt{in}\ Q \mid u\langle v\rangle \ \rightarrow \ \texttt{def}\ u\langle y\rangle \triangleright P\ \texttt{in}\ Q \mid \{v/y\}P.$$

More exactly, process $Q$ can send a name $v$ along the channel $u$, while process $P$ waits at the other end of channel $u$ to receive certain channel names. When the name $v$ is received, process $Q$ continues its execution in parallel with process $P$ in which all free occurrences

of $y$ are replaced by $v$, i.e., $\{v/y\}P$. Then, channel $u$ remains open to receive other names. The formal definition is given below.

**Definition 3.** *The* reduction relation $\rightarrow \subseteq \mathcal{P} \times \mathcal{P}$ *is defined as the smallest relation satisfying:*

r1: $\texttt{def}\ u_1\langle v_1\rangle \triangleright Q_1\ \texttt{in def}\ u_2\langle v_2\rangle \triangleright Q_2\ \texttt{in}\ \ldots\ \texttt{def}\ u_n\langle v_n\rangle \triangleright Q_n\ \texttt{in}\ (P \mid u_i\langle v\rangle) \rightarrow$
$\texttt{def}\ u_1\langle v_1\rangle \triangleright Q_1\ \texttt{in def}\ u_2\langle v_2\rangle \triangleright Q_2\ \texttt{in}\ \ldots\ \texttt{def}\ u_n\langle v_n\rangle \triangleright Q_n\ \texttt{in}\ (P \mid \{v/v_i\}Q_i)$
*if* $\{u_{i+1}, \ldots, u_n\} \cap (\texttt{fn}(Q_i) \cup \{u_i\}) = \emptyset$ *with* $i \in [n]$ *and* $n \geq 1$

r2: $$\frac{P_1 \rightarrow P_2}{\texttt{def}\ u\langle v\rangle \triangleright Q\ \texttt{in}\ P_1 \rightarrow \texttt{def}\ u\langle v\rangle \triangleright Q\ \texttt{in}\ P_2}$$

r3: $$\frac{P_1 \equiv Q_1,\ Q_1 \rightarrow Q_2 \text{ and } Q_2 \equiv P_2}{P_1 \rightarrow P_2}\ .$$

It is worth noting that there is no rule for parallel composition in the definition of this reduction. The following (easy-to-prove) results show that such a rule for parallel composition is just a consequence.

**Lemma 1.** *Considering any substitution* $\sigma = \{x/y\}$,
$P \equiv Q$ *implies* $\sigma P \equiv \sigma Q$, *and* $P_1 \rightarrow P_2$ *implies* $\sigma P_1 \rightarrow \sigma P_2$.

**Proposition 1.** $P_1 \rightarrow P_2$ *implies* $Q \mid P_1 \rightarrow Q \mid P_2$.

## 3. Hypergraphs

In the theory of distributed systems, Petri nets [5] and $\pi$-nets [6] provide both algebraic and graphical descriptions for concurrent systems. Compared to Petri nets, our hypergraph model for pattern calculus has a flexible structure; compared to $\pi$-nets, the hypergraph model is simpler, but still able to describe a large class of processes.

Following [7], we present the definitions for hypergraphs and some standard related notions such as isomorphism and contractions (on nodes and on edges). For a set $S$ of *hyperedges* and a set $V$ of *vertices*, it is defined an *incidence* relation $E \subseteq S \times V$. A *rooted hypergraph* is a tuple $H = \langle S, V, E, s\rangle$, where $s \in S$ is the *root hyperedge*. For a hypergraph $H$, we use the notations $S_H$, $V_H$, $E_H$ and $s_H$.

The graphical presentation of a hypergraph H is provided by:

- A hyperedge $t$ represented as an oval with its name $t$ outside.
- A vertex $v$ represented as a point having the name $v$ .
- An element $(t, v)$ of the incidence relation represented as lines from the hyperedge $t$ to the vertex $v$; "v lies on t" whenever $(t, v) \in E_H$.
- The root indicated by an arrow pointing to the hyperedge $s_H$.

**Example 1.** *Considering a hypergraph with* $S_H = \{s, t, t'\}$, $V_H = \{v, w, w'\}$, $E_H = \{(s, v), (t, v), (t, w), (t', v), (t', w')\}$ *and* $s_H = \{t\}$, *various graphical representations are depicted in Figure 1. The left representation uses only lines of non-zero length, while the right one uses only lines of zero length. The representation in the middle is a compromise (between lengths) to provide a reasonable picture.*

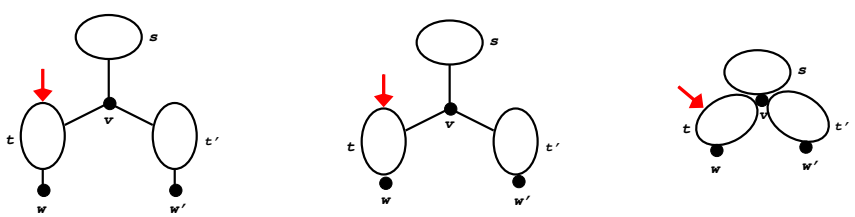



**Figure 1.** Graphical representations of a hypergraph.

For a given rooted hypergraph $H$ and a nonempty subset $W \subseteq V_H$ of nodes, a *contraction on vertices* is specified by the hypergraph $H/W$ with the same root hyperedge

($s_{H/W} = s_H$), the same hyperedges ($S_{H/W} = S_H$), but with $V_{H/W} = (V_H \setminus W) \cup \{v\}$ for a fresh $v \notin V_H$ and $E_{H/W} = (E_H \setminus S_H \times W) \cup \{\ (t, v) \mid \{t\} \times W \cap E_H \neq \emptyset\ \}$.

For a given rooted hypergraph $H$ and a nonempty subset $T \subseteq S_H$ of hyperedges, a *contraction on hyperedges* is specified by the hypergraph $H/T$ with the same set of vertices ($V_{H/T} = V_H$), but with $S_{H/T} = (S_H \setminus T) \cup \{t\}$ for a fresh $t \notin S_H$, and $E_{H/T} = (E_H \setminus T \times V_H) \cup \{\ (t, v) \mid T \times \{v\} \cap E_H \neq \emptyset\ \}$. Regarding the root hyperedge $s_{H/T}$, if $s_H \in T$ then $s_{H/T} = t$, otherwise it remains the same $s_H$.

**Example 2.** *Considering two vertices $v, w \in V_H$, we denote by $H_{v=w}$ the contraction on vertices in $H/\{v, w\}$. For two hyperedges $s, t \in S_H$, we denote by $H_{s=t}$ the contraction on hyperedges in $H/\{s, t\}$. For the hypergraph $H$ used in the previous example, the contraction on vertices $H_{w=w'}$ and the contraction on hyperedges $H_{t=t'}$ are depicted in Figure 2.*

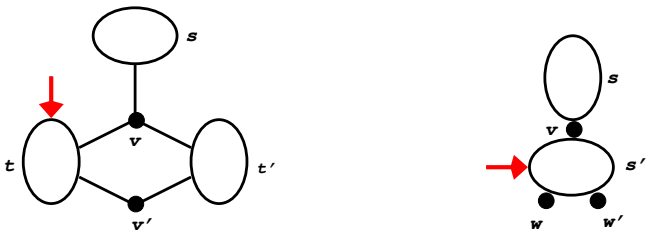



**Figure 2.** A contraction on vertices (**left**), and a contraction on hyperedges (**right**).

The *isomorphism* between two hypergraphs $H$ and $H'$ is defined by two bijections $\phi_S : S_H \to S_{H'}$ and $\phi_V : V_H \to V_{H'}$ which satisfy $\phi_S(s_H) = s_{H'}$ and $(s, v) \in E_H$ if and only if $(\phi_S(s), \phi_V(v)) \in E_{H'}$ for all $s \in S_H$ and $v \in V_H$. In such a situation, we say that hypergraphs $H$ and $H'$ are isomorphic (denoted by $H = H'$).

The isomorphism relation is an equivalence over hypergraphs. The names of hyperedges and vertices do not play any role in the isomorphism of hypergraphs. For the graphical representation of an equivalence class, it can be used any hypergraph (after removing the names of hyperedges and vertices).

## 4. Control Structure for Pattern Calculus

Milner proposed the control structures and action calculi as a unifying framework for the models of concurrent systems in [8]. A control structure defines the static aspects of a process calculus, while the corresponding action calculus describes various models of interactive behaviours. Regarding the behaviour, distinct action calculi differ only in their generators (called controls). Thus, the previously mentioned Petri nets and $\pi$-nets, as well as our hypergraph model, differ only in their generators. Analyzing these generators, it is possible to compare and classify the formal models for concurrent systems. Moreover, by selecting some specific generators, it is possible to combine existing models in order to obtain a new desired model.

The control structures presented here follow the definitions of [9]. Essentially, a control structure is defined by a set of terms, an equational theory, and a reduction relation over terms. This fact is condensed in the following expression:

$$ControlStructure = Actions + EquationalTheory + Reaction.$$

From an algebraic viewpoint, a control structure is a symmetric strict monoidal category with an additional structure [10]. The morphisms of the symmetric strict monoidal category correspond to the terms of the control structure; they are denoted by $a, b, c, \ldots$ and called *actions*.

The control structure uses an enumerable set $\mathcal{X}$ of *names* together with a signature $(P, \mathcal{K})$ in which $P$ is a set of prime arities and $\mathcal{K}$ is a set of control operators. Every name $x \in \mathcal{X}$ has a prime arity $p \in P$, and this is denoted by $x : p$. Each control $K \in \mathcal{K}$ has an arity rule. In addition to the specific control operators, every control structure contains a *datum*

operator $\langle x \rangle : \epsilon \to p$ (where $x : p$) and a *discard* operator $\omega_p : p \to \epsilon$, as well as an *abstractor* operator $\mathtt{ab}_x\, a : p \otimes m \to p \otimes n$ (where $x : p$ and $a : m \to n$). The equational theory is the same for all control structures. To express the evolution, a set $R$ of *reaction rules* is used; reaction rules are ordered pairs of terms with the same arity.

Each action $a$ possesses a *surface* $\mathtt{surf}(a) = \{x \in X \mid \exists p.x : p \text{ and } \mathtt{ab}_x\, a \neq \mathtt{id}_p \otimes a\}$.

The equality $=$ between actions is valid whenever the equation $a = b$ could be proved by using the axioms of the control structure; otherwise $a \neq b$.

We present here some results used later in the proofs of our results.

**Proposition 2.** *The following properties hold in any control structure:*

$$\mathtt{surf}(\langle x \rangle) \subseteq \{x\} \quad \mathtt{surf}(\omega) = \emptyset \quad \mathtt{surf}(a \otimes b) \subseteq \mathtt{surf}(a) \cup \mathtt{surf}(b).$$

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

$$\mathtt{surf}(\mathtt{id}) = \emptyset \quad \mathtt{surf}(\mathtt{p}) = \emptyset \quad \mathtt{surf}(a \cdot b) \subseteq \mathtt{surf}(a) \cup \mathtt{surf}(b).$$

*Additionally,*

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

**Proposition 3.** *The following properties are provable in any control structure:*

1. $[x/y](a \cdot b) = [x/y]a \cdot [x/y]b$;
2. $[x/y](a \otimes b) = [x/y]a \otimes [x/y]b$;
3. $[x/y](z)a = (z)[x/y]a$ $\quad$ *if* $z \notin \{x, y\}$;
4. $[x/y](x)a = (w)[x/y][w/x]a$ $\quad$ *if* $w \notin \mathtt{surf}(a) \cup \{x, y\}$ *and* $x \neq y$.

We define the control structure for our pattern calculus, emphasizing on its actions. We also present a graphical representation for its processes. The monoid $(\mathbb{N}, +, 0)$ of the natural numbers provides the arity monoid of the control structure, with $m, n, k, \ldots$ ranging over natural numbers, and $[n] = \{1, 2, \ldots, n\}$ denoting the first $n$ natural numbers. The (unique) prime arity 1 is associated with each name $x \in \mathcal{X}$. For a number $k$ and a function $f : [n] \to Y$, we define $k \oplus f : \{k+1, \ldots k+n\} \to Y$ by $(k \oplus f)(i) = f(i-k)$. Following [9], the control structure is defined over the set $\mathcal{X} = \{z_i \mid i \in \mathbb{N}\}$ of names using $x, y, u, \ldots$ as meta-variables.

Regarding the actions of the control structure for our calculus, they are given by enriched hypergraphs called shortly *pattern nets*. An action $a = (H, \Sigma)$ with arity $m \to n$ is given by a hypergraph $H$ together with its decoration $\Sigma = \langle \mathtt{I}, \mathtt{O}, \lambda, \tau, \mu \rangle$ consisting of:

- An *input function* given by an injective function $\mathtt{I} : [m] \to V_H$;
- An *output function* given by a function $\mathtt{O} : [n] \to V_H$;
- A *label function* given by an injective function $\lambda : Z \to V_H$, where $Z \subseteq \mathcal{X}$;
- A *transition relation* $\tau \subseteq V_H \times V_H$;
- A *resource function* $\mu : S_H \to \mathbb{N}^{V_H \times V_H}$.

We can look at these functions as multisets over $S_H \times V_H \times V_H$. We denote by $\{x, y, y\}$ a multiset $\mu$ over $\{x, y, z\}$ such that $\mu(x) = 1$, $\mu(y) = 2$ and $\mu(z) = 0$; we use the standard multiset operations over these functions: $(\uplus, -, \ldots)$.

We extend in a straightforward way the isomorphism and contraction introduced for hypergraphs. Considering $a_i = (H_i, \Sigma_i)$ with $\Sigma_i = \langle \mathtt{I}_i, \mathtt{O}_i, \lambda_i, \tau_i, \mu_i \rangle$ ($i \in [2]$), the nets $a_1$ and $a_2$ are isomorphic if there is a hypergraph isomorphism $(\phi_S, \phi_V)$ 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 $(v, v') \in \tau_1$ if and only if

$(\phi_V(v), \phi_V(v')) \in \tau_2$, together with $\mu_1(s, v, v') = \mu_2(\phi_S(s), \phi_V(v), \phi_V(v'))$ for all $s \in S_{H_1}$ and $v, v' \in V_{H_1}$.

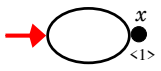
For the graphical representations of the pattern nets, let us consider a generic net $a = (H, \Sigma)$ with $\Sigma = \langle \mathtt{I}, \mathtt{O}, \lambda, \tau, \mu \rangle$; the hypergraph $H$ is presented by assuming that its lines are of length zero (see Figure 1):

- Whenever $\mathtt{I}(i) = v$, $\mathtt{O}(k) = v'$ and $\lambda(x) = w$, an *input label* $(i)$ is assigned to vertex $v$, an *output label* $\langle k \rangle$ to vertex $v'$, and a *name label* $x$ to vertex $w$;
- Whenever $(v, v') \in \tau$, an arc is drawn outside any oval from vertex $v$ to vertex $v'$;
- Whenever $\mu(s, v, v') > 0$, $v$ and $v'$ lie on the same hyperedge $s$: $(s, v), (s, v') \in E_H$; more exactly, whenever $\mu(s, v, v') = k > 0$, we have $k$ arcs inside the oval $s$ from vertex $v$ to vertex $v'$.

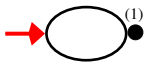
As for hypergraphs, isomorphic nets are not distinguished. The names of vertices and hyperedges do not play any role in the isomorphic nets, and so the graphical representation of an isomorphic (equivalence) class of pattern nets is given by any net of the class after removing the names of vertices and hyperedges.

The *control structure operators* for our pattern nets are:

- *datum* $\langle x \rangle^\gamma = (H, \Sigma) : 0 \to 1$ defined by
$H = \langle \{s\}, \{v\}, \{(s, v)\}, s \rangle$ and
$\Sigma = \langle \emptyset, \{1 \mapsto v\}, \{x \mapsto v\}, \emptyset, \emptyset \rangle$



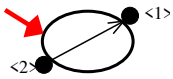

- *discard* $\omega^\gamma = (H, \Sigma) : 1 \to 0$ defined by
$H = \langle \{s\}, \{v\}, \{(s, v)\}, s \rangle$ and
$\Sigma = \langle \{1 \mapsto v\}, \emptyset, \emptyset, \emptyset, \emptyset \rangle$




The three *controls* generating the pattern nets are:

– $\nu^\gamma = (H, \Sigma) : 0 \to 1$ defined by
$H = \langle \{s\}, \{v\}, \{(s, v)\}, s \rangle$ and
$\Sigma = \langle \emptyset, \{1 \mapsto v\}, \emptyset, \emptyset, \emptyset \rangle$


– $\mathtt{out}^\gamma = (H, \Sigma) : 2 \to 0$ defined by
$H = \langle \{s\}, \{v, v'\}, \{(s, v), (s, v')\}, s \rangle$ and
$\Sigma = \langle \emptyset, \{1 \mapsto v, 2 \mapsto v'\}, \emptyset, \emptyset, \{(s, v', v)\} \rangle$



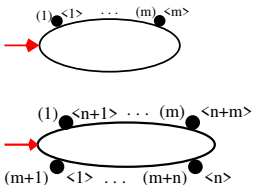

– If $a = (H, \Sigma) : 1 \to 0$ and $\Sigma = \langle \mathtt{I}, \mathtt{O}, \lambda, \tau, \mu \rangle$, then
$\mathtt{def}^\gamma\, a = (H', \Sigma') : 1 \to 0$, where
$H' = \langle S_H \cup \{t\}, V_H \cup \{v\}, E_H \cup \{(t, v)\}, t \rangle$, for fresh
$t \notin S_H$ and $v \notin V_H$, and
$\Sigma' = \langle \{1 \mapsto v\}, \mathtt{O}, \lambda, \tau \cup \{(v, \mathtt{I}(1))\}, \mu \rangle$.

The *equational theory* is defined by the following operators.

Let us consider the nets $a_i = (H_i, \Sigma_i)$ 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 $i \in [2]$. Without loss of generality, we consider $s_{H_1} = s_{H_2} = s$. $(S_{H_1} - \{s_{H_1}\}) \cap (S_{H_2} - \{s_{H_2}\}) = \emptyset$ and $\lambda_1(z) = \lambda_2(z)\ \forall z \in Z_1 \cap Z_2$, as well as $(V_{H_1} - \lambda_1(Z_1 \cap Z_2)) \cap (V_{H_2} - \lambda_2(Z_1 \cap Z_2)) = \emptyset$.

– *Identity* $\mathtt{id}^\gamma_m = (H, \Sigma) : m \to m$ defined by
$H = \langle \{s\}, \{v_i | i \in [m]\}, \{(s, v_i) | i \in [m]\}, s \rangle$ and
$\Sigma = \langle \{i \mapsto v_i | i \in [m]\}, \{i \mapsto v_i | i \in [m]\}, \emptyset, \emptyset, \emptyset \rangle$

– *Symmetry* $\mathtt{p}^\gamma_{m,n} = (H, \Sigma) : m + n \to n + m$ 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 two nets $a_1 : m \to n$ and $a_2 : k \to l$ is obtained by combining them as follows: in $a_2$, the input labels are incremented by $m$ and the output labels are incremented by $n$; then overlap the two roots and the vertices of $a_1$ and of $a_2$ with the same name. Formally, $a_1 \otimes a_2 = (H, \Sigma)$, where

$$H = \langle S_{H_1} \cup S_{H_2}, V_{H_1} \cup V_{H_2}, E_{H_1} \cup E_{H_2}, s \rangle \text{ and}$$

$$\Sigma = \langle \mathtt{I}_1 \cup m \oplus \mathtt{I}_2, \mathtt{O}_1 \cup n \oplus \mathtt{O}_2, \lambda_1 \cup \lambda_2, \tau_1 \cup \tau_2, \mu_1 \uplus \mu_2 \rangle.$$

– *Composition* $a_1 \cdot a_2 : m \to k$ of two nets $a_1 : m \to n$ and $a_2 : n \to k$ is obtained by combining them as follows: overlap the two roots and vertices of $a_1$ and $a_2$ with the same name; for every $i \in [n]$, overlap the vertex labelled $\langle i \rangle$ in $a_1$ with the vertex labelled $(i)$ in $a_2$, and then remove the 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

$$H = \langle S_{H_1} \cup S_{H_2}, V_{H_1} \cup V_{H_2}, E_{H_1} \cup E_{H_2}, s \rangle \text{ and}$$

$$\Sigma = \langle \mathtt{I}_1, \mathtt{O}_2, \lambda_1 \cup \lambda_2, \tau_1 \cup \tau_2, \mu_1 \uplus \mu_2 \rangle.$$

– *Abstractor.* Let us consider a net $a = (H, \Sigma) : m \to n$ with $\Sigma = \langle \mathtt{I}, \mathtt{O}, \lambda, \tau, \mu \rangle$.

Then $\mathtt{ab}^{\gamma}_{x}\, a : 1 + m \to 1 + n$ is obtained from $a$ in the following steps: increment all the input and output labels by 1; assign both the input label $(1)$ and the output label $\langle 1 \rangle$ to vertex $x$, and then remove the label $x$. Formally, $\mathtt{ab}^{\gamma}_{x}\, a = (H, \Sigma')$, where

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

In general, these operators over the pattern nets are well-defined. However, the abstractor $\mathtt{ab}^{\gamma}_{x}\, a$ is not well-defined if a vertex labelled by $x$ is not contained in the net $a$. To avoid such a situation, we adjust the definition of the above operators by

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

where op stands for each operator defined above, and $\mathbf{i} = (H, \Sigma)$ is the pattern net

$$H = \langle \{s\}, \{v_i | i \in \mathbb{N}\}, \{(s, v_i) | i \in \mathbb{N}\}, s \rangle$$

$$\Sigma = \langle \emptyset, \emptyset, \{z_i \mapsto v_i | i \in \mathbb{N}\}, \emptyset, \emptyset \rangle.$$

Following [9], it is not difficult to prove the following result.

**Proposition 4.** *The operators* $\langle x \rangle$, $\omega$, $\nu$, $\mathtt{out}$, $\mathtt{def}$, $\mathtt{id}$, $\mathtt{p}$, $\cdot$, $\otimes$ *and* $\mathtt{ab}_x$ *define a control structure.*

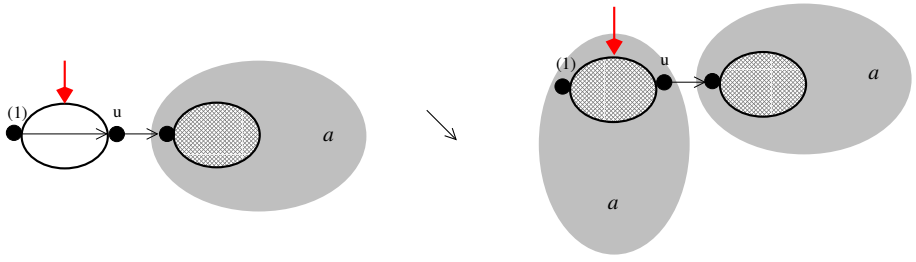
The actions of this control structure determine the *hypergraph model* for the pattern calculus. We actually use the derived control operators:

$$\begin{array}{rcl} \mathtt{out}_u & \stackrel{def}{=} & (\langle u \rangle \otimes \mathtt{id}_1) \cdot \mathtt{out}; \\ \mathtt{def}_u\, a & \stackrel{def}{=} & \langle u \rangle \cdot \mathtt{def}\, a. \end{array}$$

**The reaction** $\searrow$ is the smallest relation over the pattern nets closed under equality, composition, tensorial product and abstraction which satisfies the control rule

$$\mathtt{out}_u \otimes \mathtt{def}_u\, a \searrow a \otimes \mathtt{def}_u\, a.$$

The corresponding graphical description of the reaction rule is given by:



In this diagram, the scope of the $\mathtt{def}$ operator is represented as a gray patch; due to the properties derived from the syntax of our calculus, this patch can actually be determined from the hypergraph structure.

The operators, actions and reaction complete the definition of our nets.

It is worth noting that the $\mathtt{def}$ operator can be generalized, namely we can have a more general control operator $\mathtt{def}_{u_1 \dots u_m}$ by $\mathtt{def}_{u_1 \dots u_m}\, a = (\langle u_1 \rangle \otimes \dots \otimes \langle u_m \rangle) \cdot \mathtt{def}\, a$. Moreover, the corresponding graphical representation is extended by using $m$ external arcs to connect the new root hyperedge to the old one. The corresponding reaction is generalized in the following way: $\mathtt{out}_{u_1} \otimes \dots \otimes \mathtt{out}_{u_m} \otimes \mathtt{def}_{u_1 \dots u_m}\, a \searrow a \otimes \mathtt{def}_{u_1 \dots u_m}\, a$.

We present some proprieties of the pattern nets. The proofs of these properties are tedious (but easy), based mainly on definitions and the structure of the nets.

**Lemma 2.** *We have the following properties:*

1. $\mathtt{surf}(\mathtt{out}_u) \subseteq \{u\}$ *and* $\mathtt{surf}(\mathtt{def}_u\, a) \subseteq \{u\} \cup \mathtt{surf}(a)$.
2. *For any substitution* $\sigma = \{x/y\}$, $[x/y]\mathtt{out}_u = \mathtt{out}_{\sigma u}$ *and* $[x/y]\mathtt{def}_u\, a = \mathtt{def}_{\sigma u}\, [x/y]a$.
3. *If* $a \searrow b$, *then there exists* $b'$ *such that* $b = b'$ *and* $\mathtt{surf}(b') \subseteq \mathtt{surf}(a)$.
4. $\langle v \rangle \cdot \mathtt{out}_w \otimes \mathtt{def}_u\, (y)a \searrow b$ *iff* $u = w$ *and* $b = [v/y]a \otimes \mathtt{def}_u\, (y)a$.
5. $a_1 \otimes a_2 \otimes a_3 \searrow c$ *iff*
   - *either there exists* $i \in [3]$ *such that* $a_i \searrow b$ *and* $c = b \otimes a_j \otimes a_k$, *or*
   - *there exist* $i, j \in [3]$ *such that* $a_i \otimes a_j \searrow b$ *and* $c = b \otimes a_k$, *where* $[3] = \{i, j, k\}$.
6. *Whenever* $u \notin \mathtt{surf}(b)$, $b \otimes \mathtt{def}_u\, a \searrow c$ *iff* $b \searrow b'$ *and* $c = b' \otimes \mathtt{def}_u\, a$.
7. $\nu \cdot (x)a \searrow b$ *iff* $a \searrow a'$ *and* $b = \nu \cdot (x)a'$.

## 5. Fully Abstract Hypergraph Model of the Pattern Calculus

This section presents the main results of the paper. These results reveal the hypergraphs as a fully abstract model for the pattern calculus. According to [11], a model is fully abstract if all observationally equivalent terms in the object language represent the same object in the model. This means that processes with different behaviour are not mapped to the same hypergraph. Moreover, we prove a correspondence between the reduction of the processes and the reduction of their hypergraph representation.

**Definition 4.** *The semantic relationship* $[\![ - ]\!]$ *between the pattern calculus processes and the pattern nets is defined by structural induction as follows:*

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. $[\![ \mathtt{def}\ u\langle y \rangle \triangleright P\ \mathtt{in}\ Q ]\!] = \nu \cdot (u)(\, [\![ Q ]\!] \otimes \mathtt{def}_u\, (y)\, [\![ P ]\!]\, )$ .

We prove some results involving this semantic relationship $[\![ - ]\!]$.

**Lemma 3.** *For every process* $P \in \mathcal{P}$, *we have* $[\![ P ]\!] : 0 \to 0$ .

**Proof.** A simple induction on the structure of $P$. In the case of our nets, 0 is the neutral element of the arity monoid $(\mathbb{N}, +, 0)$. For case (4) of the previous definition, we use the discard operator $\omega$ instead of $\omega_p$. Since 1 is the only prime arity $p$ of the monoid $(\mathbb{N}, +, 0)$, we omit the index without any risk of confusion. □

**Lemma 4.** *For every process* $P \in \mathcal{P}$, *we have* $\mathtt{surf}(\, [\![ P ]\!]\, ) \subseteq \mathtt{fn}(P)$.

**Proof.** By induction on the structure of $P$ (the proof uses Lemma 2). □

**Lemma 5.** *For two names* $x, y \in X$ *and a process* $P \in \mathcal{P}$, *we have* $[\![ \{x/y\}P ]\!] = [x/y]\, [\![ P ]\!]$ .

**Proof.** Induction on the definition of the substitution over processes (and Lemma 5). □

**Proposition 5.** *If* $P \equiv Q$, *then* $[\![ P ]\!] = [\![ Q ]\!]$.

**Proof.** Induction on the definition of structural congruence. Let us consider the relation

$$\sim = \{(P, Q) \in \mathcal{P} \mid P \equiv Q \text{ and } [\![ P ]\!] = [\![ Q ]\!]\}.$$

Proof is reduced to the equality between $\sim$ and $\equiv$. Obviously, $\sim \subseteq \equiv$. We show that $\sim$ satisfies the axioms from the definition of $\equiv$. Since $\equiv$ is the smallest relation satisfying these axioms, it follows that $\equiv \subseteq \sim$, and so $\sim = \equiv$. Thus, to prove that $\equiv \subseteq \sim$, it is enough to verify that $\sim$ satisfies the axioms from the definition of $\equiv$.

The cases $P \mid 0 \equiv P$, $P \mid Q \equiv Q \mid P$ and $(P \mid Q) \mid R \equiv P \mid (Q \mid R)$ are rather trivial, based on the fact that $\mathtt{id}_0$ is neutral for tensor product, together with the commutativity and associativity of tensor product $\otimes$ in the equational theory (of the control structures).

Let us consider the other cases.

– $\mathtt{def}\ u\langle v\rangle \triangleright P\ \mathtt{in}\ Q \equiv \mathtt{def}\ u\langle t\rangle \triangleright \{t/v\}P\ \mathtt{in}\ Q$, if $t \notin \mathtt{fn}(P)$.

Assume $t \notin \mathtt{fn}(P)$. By Lemma 4, it follows that $t \notin \mathtt{surf}([\![ P ]\!])$. Then,

$[\![\ \mathtt{def}\ u\langle t\rangle \triangleright \{t/v\}P\ \mathtt{in}\ Q\ ]\!] =$

$= \nu \cdot (u)([\![ Q ]\!] \otimes \mathtt{def}_u\,(t)\,[\![ \{t/v\}P ]\!])$ by Lemma 5

$= \nu \cdot (u)([\![ Q ]\!] \otimes \mathtt{def}_u\,(t)[t/v]\,[\![ P ]\!])$

$= [\![\ \mathtt{def}\ u\langle v\rangle \triangleright P\ \mathtt{in}\ Q\ ]\!]$.

– $\mathtt{def}\ u\langle v\rangle \triangleright P\ \mathtt{in}\ Q \equiv \mathtt{def}\ w\langle v\rangle \triangleright \{w/u\}P\ \mathtt{in}\ \{w/u\}Q$ if $v \notin \{u, w\}$, $w \notin \mathtt{fn}(P \mid Q)$.

Assume $v \notin \{u, w\}$ and $w \notin \mathtt{fn}(P \mid Q)$; then, $u \neq v$ and $w \notin \mathtt{fn}(P) \cup \mathtt{fn}(Q) \cup \{v\}$. By Lemma 4, $w \notin \mathtt{surf}([\![ P ]\!]) \cup \mathtt{surf}([\![ Q ]\!])$. If $u = w$, then the result is trivial.

Let us assume that $u \neq w$.

$[\![\ \mathtt{def}\ w\langle v\rangle \triangleright \{w/u\}P\ \mathtt{in}\ \{w/u\}Q\ ]\!] =$

$= \nu \cdot (w)([\![ \{w/u\}Q ]\!] \otimes \mathtt{def}_w\,(v)\,[\![ \{w/u\}P ]\!])$ by Lemma 5

$= \nu \cdot (w)([w/u]\,[\![ Q ]\!] \otimes \mathtt{def}_w\,(v)[w/u]\,[\![ P ]\!])$ by Proposition 3 and Lemma 2

$= \nu \cdot (w)[w/u]([\![ Q ]\!] \otimes \mathtt{def}_u\,(v)\,[\![ P ]\!])$ by Lemma 2

$= [\![\ \mathtt{def}\ u\langle v\rangle \triangleright P\ \mathtt{in}\ Q\ ]\!]$.

– $Q_1 \mid \mathtt{def}\ u\langle v\rangle \triangleright P\ \mathtt{in}\ Q_2 \equiv \mathtt{def}\ u\langle v\rangle \triangleright P\ \mathtt{in}\ (Q_1 \mid Q_2)$ if $u \notin \mathtt{fn}(Q_1)$.

Assume $u \notin \mathtt{fn}(Q_1)$. By Lemma 4, $u \notin \mathtt{surf}([\![ Q_1 ]\!])$. Then,

$[\![\ \mathtt{def}\ u\langle v\rangle \triangleright P\ \mathtt{in}\ (Q_1 \mid Q_2)\ ]\!] =$

$= \nu \cdot (u)([\![ Q_1 ]\!] \otimes [\![ Q_2 ]\!] \otimes \mathtt{def}_u\,(v)\,[\![ P ]\!])$ by Lemma 3 and Proposition 2

$= [\![\ Q_1 \mid \mathtt{def}\ u\langle v\rangle \triangleright P\ \mathtt{in}\ Q_2\ ]\!]$.

– $\mathtt{def}\ u\langle v\rangle \triangleright P_1\ \mathtt{in}\ \mathtt{def}\ w\langle t\rangle \triangleright P_2\ \mathtt{in}\ Q \equiv \mathtt{def}\ w\langle t\rangle \triangleright P_2\ \mathtt{in}\ \mathtt{def}\ u\langle v\rangle \triangleright P_1\ \mathtt{in}\ Q$
if $u \neq w$, $u \notin \mathtt{fn}(P_2)$, and $w \notin \mathtt{fn}(P_1)$.

Assume $u \neq w$, $u \notin \mathtt{fn}(P_2)$ and $w \notin \mathtt{fn}(P_1)$. By Lemma 4, it follows that $u \notin \mathtt{surf}([\![ P_2 ]\!])$ and $w \notin \mathtt{surf}([\![ P_1 ]\!])$. Furthermore, by Lemma 2, $w \notin \mathtt{surf}(\mathtt{def}_u\,(v)\,[\![ P_1 ]\!]$.

$[\![\ \mathtt{def}\ u\langle v\rangle \triangleright P_1\ \mathtt{in}\ \mathtt{def}\ w\langle t\rangle \triangleright P_2\ \mathtt{in}\ Q\ ]\!] =$

$= \nu \cdot (u)(\nu \cdot (w)([\![ Q ]\!] \otimes \mathtt{def}_w\,(t)\,[\![ P_2 ]\!]) \otimes \mathtt{def}_u\,(v)\,[\![ P_1 ]\!])$ by Proposition 2

$= \nu \cdot (u)(\nu \cdot (w)([\![ Q ]\!] \otimes \mathtt{def}_w\,(t)\,[\![ P_2 ]\!] \otimes \mathtt{def}_u\,(v)\,[\![ P_1 ]\!]))$ by Proposition 2

$= (\nu \otimes \nu) \cdot (u)(w)([\![ Q ]\!] \otimes \mathtt{def}_w\,(t)\,[\![ P_2 ]\!] \otimes \mathtt{def}_u\,(v)\,[\![ P_1 ]\!])$ **= X**.

In a similar way, we obtain $[\![\ \mathtt{def}\ w\langle t\rangle \triangleright P_2\ \mathtt{in}\ \mathtt{def}\ u\langle v\rangle \triangleright P_1\ \mathtt{in}\ Q\ ]\!] =$

$= (\nu \otimes \nu) \cdot (w)(u)([\![ Q ]\!] \otimes \mathtt{def}_u\,(v)\,[\![ P_1 ]\!] \otimes \mathtt{def}_w\,(t)\,[\![ P_2 ]\!])$ **= Y**.

To complete the proof, it remains to prove that **X** = **Y**.

**X** = by Proposition 2

$= (\nu \otimes \nu) \cdot \mathtt{p}_{1,1} \cdot (w)(u)([\![ Q ]\!] \otimes \mathtt{def}_w\,(t)\,[\![ P_2 ]\!] \otimes \mathtt{def}_u\,(v)\,[\![ P_1 ]\!])$

$= \mathtt{p}_{0,0} \cdot (\nu \otimes \nu) \cdot (w)(u)([\![ Q ]\!] \otimes \mathtt{def}_u\,(v)\,[\![ P_1 ]\!] \otimes \mathtt{def}_w\,(t)\,[\![ P_2 ]\!])$ □

$=$ **Y**.

**Theorem 1.** *If* $P \to Q$, *then* $[\![ P ]\!] \searrow [\![ Q ]\!]$.

**Proof.** By induction on the definition of $P \to Q$.
* r1: $P \to Q$ is

$$\texttt{def}\ u_1\langle y_1\rangle \triangleright Q_1\ \texttt{in}\ \texttt{def}\ u_2\langle y_2\rangle \triangleright Q_2\ \texttt{in}\ \ldots\ \texttt{def}\ u_n\langle y_n\rangle \triangleright Q_n\ \texttt{in}\ R \mid u_i\langle v\rangle \ \to$$

$$\texttt{def}\ u_1\langle y_1\rangle \triangleright Q_1\ \texttt{in}\ \texttt{def}\ u_2\langle y_2\rangle \triangleright Q_2\ \texttt{in}\ \ldots\ \texttt{def}\ u_n\langle y_n\rangle \triangleright Q_n\ \texttt{in}\ R \mid \{v/y_i\}Q_i, \quad \text{where}$$

$\{u_{i+1}, \ldots, u_n\} \cap (\texttt{fn}(Q_i) \cup \{u_i\}) = \emptyset$, $i \in [n]$, and $n \geq 1$. According to Lemma 2 and Lemma 4, $\{u_{i+1}, \ldots, u_n\} \cap \texttt{surf}(\texttt{def}_{u_i}(y_i)\ [\![ Q_i ]\!]\ ) = \emptyset$. According to Proposition 2 and using the compatibility of $\searrow$ with composition, tensorial product and abstraction,
$[\![ P ]\!] = \nu \cdot (u_1)(\texttt{def}_{u_1}(y_1)\ [\![ Q_1 ]\!]\ \otimes$

$$\begin{aligned}
&= \quad \vdots \\
&\quad \nu \cdot (u_i)(\texttt{def}_{u_i}(y_i)\ [\![ Q_i ]\!]\ \otimes \\
&\qquad \vdots \\
&\qquad \nu \cdot (u_n)(\texttt{def}_{u_n}(y_n)\ [\![ Q_n ]\!]\ \otimes\ [\![ R ]\!]\ \otimes \langle v\rangle \cdot \texttt{out}_{u_i}) \ldots) \ldots) \\
&= \quad \nu \cdot (u_1)(\texttt{def}_{u_1}(y_1)\ [\![ Q_1 ]\!]\ \otimes \qquad \text{by Lemma 2} \\
&\qquad \vdots \\
&\qquad \nu \cdot (u_{i-1})(\texttt{def}_{u_{i-1}}(y_{i-1})\ [\![ Q_{i-1} ]\!]\ \otimes \\
&\qquad \nu \cdot (u_i)( \\
&\qquad \nu \cdot (u_{i+1})(\texttt{def}_{u_{i+1}}(y_{i+1})\ [\![ Q_{i+1} ]\!]\ \otimes \\
&\qquad\qquad \vdots \\
&\qquad\qquad \nu \cdot (u_n)(\texttt{def}_{u_n}(y_n)\ [\![ Q_n ]\!]\ \otimes\ [\![ R ]\!]\ \otimes \langle v\rangle \cdot \texttt{out}_{u_i} \otimes \texttt{def}_{u_i}(y_i)\ [\![ Q_i ]\!]\ ) \ldots))))\ldots) \\
&\searrow \quad \nu \cdot (u_1)(\texttt{def}_{u_1}(y_1)\ [\![ Q_1 ]\!]\ \otimes \qquad \text{by Lemma 5} \\
&\qquad \vdots \\
&\qquad \nu \cdot (u_i)(\texttt{def}_{u_i}(y_i)\ [\![ Q_i ]\!]\ \otimes \\
&\qquad\qquad \vdots \\
&\qquad\qquad \nu \cdot (u_n)(\texttt{def}_{u_n}(y_n)\ [\![ Q_n ]\!]\ \otimes\ [\![ R ]\!]\ \otimes [v/y_i]\ [\![ Q_i ]\!]\ ) \ldots) \ldots) \\
&= \quad [\![ Q ]\!]\,.
\end{aligned}$$

* r2: $P \to Q$ is $\texttt{def}\ u\langle v\rangle \triangleright R\ \texttt{in}\ P' \to \texttt{def}\ u\langle v\rangle \triangleright R\ \texttt{in}\ Q'$ with $P' \to Q'$.
By induction, $[\![ P' ]\!] \searrow [\![ Q' ]\!]$. Since $\searrow$ is closed under composition, tensor and abstraction, it follows that

$$[\![ P ]\!] = \nu \cdot (u)(\ [\![ P' ]\!] \otimes \texttt{def}_u(v)\ [\![ R ]\!]\ ) \searrow \nu \cdot (u)(\ [\![ Q' ]\!] \otimes \texttt{def}_u(v)\ [\![ R ]\!]\ ) = [\![ Q ]\!]\,.$$

* r3: $P \to Q$ with $P \equiv P'$, $P' \to Q'$ and $Q' \equiv Q$.
By the induction hypothesis, $[\![ P' ]\!] \searrow [\![ Q' ]\!]$. By Proposition 5, we have $[\![ P ]\!] = [\![ P' ]\!]$ and $[\![ Q' ]\!] = [\![ Q ]\!]$. Since $\searrow$ is closed under equality, then $[\![ P ]\!] \searrow [\![ Q ]\!]$. □

**Lemma 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.** ($\Leftarrow$) A consequence of the fact that the reaction is closed under tensorial product and equality.
($\Rightarrow$) Induction on the structure of $P$.

– If $P$ is the empty process or a message, then $\langle v\rangle \cdot \texttt{out}_u \otimes [\![ P ]\!] \not\searrow$. Therefore, the statement of the lemma is obviously true because its premise is not satisfied.
– If $P$ is a parallel composition $P_1 \mid P_2$, then $\langle v\rangle \cdot \texttt{out}_u \otimes [\![ P_1 ]\!] \otimes [\![ P_2 ]\!] \searrow a$. Since $\langle v\rangle \cdot \texttt{out}_u \not\searrow$, it follows (Lemma 2) that one of the following cases remains possible:
  (1) $[\![ P_i ]\!] \searrow b'$ and $a = \langle v\rangle \cdot \texttt{out}_u \otimes b' \otimes [\![ P_j ]\!]$;
  (2) $[\![ P_i ]\!] \otimes [\![ P_j ]\!] \searrow b'$ and $a = \langle v\rangle \cdot \texttt{out}_u \otimes b'$;

(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, we have $[\![ P ]\!] = [\![ P_i ]\!] \otimes [\![ P_j ]\!]$.
In case (1), $[\![ P ]\!] \searrow b' \otimes [\![ P_j ]\!]$, and we consider $b = b' \otimes [\![ P_j ]\!]$. In case (2), we take $b = b'$. In case (3), by induction, we have $[\![ P_i ]\!] \searrow b'$ and $a' = \langle v \rangle \cdot \mathtt{out}_u \otimes b'$. Thus, $[\![ P ]\!] \searrow b' \otimes [\![ P_j ]\!]$, considering $b = b' \otimes [\![ P_j ]\!]$.

– If $P$ is a definition $\mathtt{def}\ w\langle t \rangle \triangleright P_1\ \mathtt{in}\ P_2$, then we may assume without losing generality that $w \notin \{u, v\}$. It follows from Lemma 2 together with Proposition 2 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 2, $\langle v \rangle \cdot \mathtt{out}_u \otimes [\![ P_2 ]\!] \otimes \mathtt{def}_w\,(t)\,[\![ P_1 ]\!] \searrow a'$ and $a = \nu \cdot (w)a'$. Since $\langle v \rangle \cdot \mathtt{out}_u \not\searrow$, $\mathtt{def}_w\,(t)\,[\![ P_1 ]\!] \not\searrow$ and $\langle v \rangle \cdot \mathtt{out}_u \otimes \mathtt{def}_w\,(t)\,[\![ P_1 ]\!] \not\searrow$, it follows (according to Lemma 2) 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 $a' = \langle v \rangle \cdot \mathtt{out}_u \otimes b'$;
(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), $[\![ P ]\!] = \nu \cdot (w)([\![ P_2 ]\!] \otimes \mathtt{def}_w\,(t)\,[\![ P_1 ]\!]) \searrow \nu \cdot (w)(b' \otimes \mathtt{def}_w\,(t)\,[\![ P_1 ]\!])$. Considering $b = \nu \cdot (w)(b' \otimes \mathtt{def}_w\,(t)\,[\![ P_1 ]\!])$, it satisfies the requirements (according to Proposition 2). In case (2), we have $[\![ P ]\!] \searrow \nu \cdot (w)b'$, and consider $b = \nu \cdot (w)b'$. In case (3), by induction hypothesis, $[\![ P_2 ]\!] \searrow b'$ and $a'' = \langle v \rangle \cdot \mathtt{out}_u \otimes b'$. Thus, $[\![ P ]\!] \searrow \nu \cdot (w)(b' \otimes \mathtt{def}_w\,(t)\,[\![ P_1 ]\!])$, and consider $b = \nu \cdot (w)(b' \otimes \mathtt{def}_w\,(t)\,[\![ P_1 ]\!])$. □

**Lemma 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.** ($\Leftarrow$) A consequence of the fact that the reaction is closed under tensorial product and equality.
($\Rightarrow$) Induction on the structure of $P$.

– If $P$ is the empty process 0, then condition 2 holds obviously.
– If $P$ is a message, then condition 2 holds by Lemma 6.
– If $P$ is a parallel composition $P_1 \mid P_2$, then $[\![ P_1 ]\!] \otimes [\![ P_2 ]\!] \otimes [\![ Q ]\!] \searrow a$.

By Lemma 2, it follows that one of the following cases is possible:

($i$) $[\![ Q ]\!] \searrow b'$ and $a = [\![ P_1 ]\!] \otimes [\![ P_2 ]\!] \otimes b'$;
($ii$) $[\![ P_i ]\!] \searrow b'$ and $a = b' \otimes [\![ P_j ]\!] \otimes [\![ Q ]\!]$;
($iii$) $[\![ P_i ]\!] \otimes [\![ P_j ]\!] \searrow b'$ and $a = b' \otimes [\![ Q ]\!]$;
($iv$) $[\![ P_i ]\!] \otimes [\![ Q ]\!] \searrow a'$ and $a = a' \otimes [\![ P_j ]\!]$, where $\{i, j\} = [2]$.

According to Proposition 2, $[\![ P ]\!] = [\![ P_i ]\!] \otimes [\![ P_j ]\!]$. In case ($i$), condition 2 holds by taking $b = b'$. In case ($ii$), we have $[\![ P ]\!] \searrow b' \otimes [\![ P_j ]\!]$. Then, condition 1 holds by taking $b = b' \otimes [\![ P_j ]\!]$. In case ($iii$), condition 1 holds by taking $b = b'$.
In case ($iv$), by induction, we distinguish two sub-cases:

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

For ($a$), we obtain $[\![ P ]\!] \searrow b' \otimes [\![ P_j ]\!]$, and condition 1 holds for $b = b' \otimes [\![ P_j ]\!]$. For ($b$), condition 2 holds for $b = b'$. In both sub-cases, some action commutations are required; they are possible according to Proposition 2.

– If $P$ is a definition $\mathtt{def}\ w\langle t \rangle \triangleright P_1\ \mathtt{in}\ P_2$, then we may assume without losing generality that $w \notin \mathtt{fn}(Q)$. By Lemma 4, $w \notin \mathtt{surf}([\![ Q ]\!])$. According to Proposition 2, we have $\nu \cdot (w)([\![ P_2 ]\!] \otimes \mathtt{def}_w\,(t)\,[\![ P_1 ]\!] \otimes [\![ Q ]\!]) \searrow a$. By Lemma 2, $[\![ P_2 ]\!] \otimes \mathtt{def}_w\,(t)\,[\![ P_1 ]\!] \otimes [\![ Q ]\!] \searrow a'$ and $a = \nu \cdot (w)a'$. Since $\mathtt{def}_w\,(t)\,[\![ P_1 ]\!] \not\searrow$, it follows from Lemma 2 that one of the following cases remains possible:

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

$(ii)$ $[\![ Q ]\!] \searrow b'$ and $a' = [\![ P_2 ]\!] \otimes \mathtt{def}_w\,(t)\,[\![ P_1 ]\!] \otimes b'$;
$(iii)$ $[\![ P_2 ]\!] \otimes \mathtt{def}_w\,(t)\,[\![ P_1 ]\!] \searrow b'$ and $a' = b' \otimes [\![ Q ]\!]$;
$(iv)$ $[\![ P_2 ]\!] \otimes [\![ Q ]\!] \searrow a''$ and $a' = a'' \otimes \mathtt{def}_w\,(t)\,[\![ P_1 ]\!]$.

In case $(i)$, condition *1* holds for $b = \nu \cdot (w)(b' \otimes \mathtt{def}_w\,(t)\,[\![ P_1 ]\!])$.

In case $(ii)$, by Lemma 2, there exists $b$ such that $\mathtt{surf}(b) \subseteq \mathtt{surf}([\![ Q ]\!])$ and $b = b'$; condition *2* holds for this $b$. In case $(iii)$, condition *1* holds for $b = \nu \cdot (w)b'$.

In case $(iv)$, we distinguish two sub-cases:

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

For $(a)$, condition *1* holds for $b = \nu \cdot (w)(b' \otimes \mathtt{def}_w\,(t)\,[\![ P_1 ]\!])$. For $(b)$, there exists $b$ such that $\mathtt{surf}(b) \subseteq \mathtt{surf}([\![ Q ]\!])$ and $b = b'$ (by Lemma 2); condition *2* holds for this $b$. Proposition 2 is used in all cases and sub-cases. □

**Lemma 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. $P \equiv \mathtt{def}\ v_1\langle t_1 \rangle \triangleright R_1\ \mathtt{in}\ \mathtt{def}\ v_2\langle t_2 \rangle \triangleright R_2\ \mathtt{in} \ldots \mathtt{def}\ v_n\langle t_n \rangle \triangleright R_n\ \mathtt{in}\ (R \mid u\langle v_n \rangle)$, *and*

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

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

**Proof.** $(\Leftarrow)$ If condition *1* holds, then the implication follows as a consequence of the fact that the reaction is closed under tensorial product and equality. If condition *2* holds, then we have

$$\begin{aligned} & [\![ P ]\!] \otimes \mathtt{def}_u\,(y)\,[\![ Q ]\!] = && \text{by Proposition 5} \\ = \;& [\![ R ]\!] \otimes \langle v \rangle \cdot \mathtt{out}_u \otimes \mathtt{def}_u\,(y)\,[\![ Q ]\!] && \text{by Lemma 2} \\ \searrow \;& [\![ R ]\!] \otimes [v/y]\,[\![ Q ]\!] \otimes \mathtt{def}_u\,(y)\,[\![ Q ]\!] && \text{by Lemma 5} \\ = \;& a. \end{aligned}$$

If condition *3* holds, it follows by Lemma 2 and Lemma 4 that $v_i \notin \mathtt{surf}(\mathtt{def}_u\,(y)\,[\![ Q ]\!])$ for every $i \in [n]$. Then,

$$\begin{aligned} & [\![ P ]\!] \otimes \mathtt{def}_u\,(y)\,[\![ Q ]\!] = && \text{by Propositions 5 and 2} \\ = \;& \nu \cdot (v_1)(\mathtt{def}_{v_1}\,(t_1)\,[\![ R_1 ]\!] \otimes && \text{by Lemma 2} \\ & \quad \vdots \\ & \quad \nu \cdot (v_n)(\mathtt{def}_{v_n}\,(t_n)\,[\![ R_n ]\!] \otimes [\![ R ]\!] \otimes \langle v_n \rangle \cdot \mathtt{out}_u \otimes \mathtt{def}_u\,(y)\,[\![ Q ]\!])\ldots) \\ \searrow \;& \nu \cdot (v_1)(\mathtt{def}_{v_1}\,(t_1)\,[\![ R_1 ]\!] \otimes && \text{by Lemma 5} \\ & \quad \vdots \\ & \quad \nu \cdot (v_n)(\mathtt{def}_{v_n}\,(t_n)\,[\![ R_n ]\!] \otimes [\![ R ]\!] \otimes [v_n/y]\,[\![ Q ]\!] \otimes \mathtt{def}_u\,(y)\,[\![ Q ]\!])\ldots) \\ = \;& a. \end{aligned}$$

$(\Rightarrow)$ 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 $[\![ P ]\!] \otimes \mathtt{def}_u\,(y)\,[\![ Q ]\!] \not\searrow$. 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

$$\begin{aligned} & [\![ P ]\!] \otimes \mathtt{def}_u\,(y)\,[\![ Q ]\!] \searrow && \text{by Lemma 2} \\ \searrow \;& [v/y]\,[\![ Q ]\!] \otimes \mathtt{def}_u\,(y)\,[\![ Q ]\!] \searrow a && \text{by Lemma 5} \\ = \;& 0 \mid \{v/y\}Q \otimes \mathtt{def}_u\,(y)\,[\![ Q ]\!]. \end{aligned}$$

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

– If $P$ is a parallel composition $P_1 \mid P_2$, then $[\![ P_1 ]\!] \otimes [\![ P_2 ]\!] \otimes \mathtt{def}_u\,(y)\,[\![ Q ]\!] \searrow a$. Since $\mathtt{def}_u\,(y)\,[\![ Q ]\!] \not\searrow$, it follows from Lemma 2 that one of the following cases remains possible:

$(i)$ $[\![ P_i ]\!] \searrow b'$ and $a = b' \otimes [\![ P_j ]\!] \otimes \mathtt{def}_u\,(y)\,[\![ Q ]\!]$,
$(ii)$ $[\![ P_i ]\!] \otimes [\![ P_j ]\!] \searrow b'$ and $a = b' \otimes \mathtt{def}_u\,(y)\,[\![ Q ]\!]$,
$(iii)$ $[\![ P_i ]\!] \otimes \mathtt{def}_u\,(y)\,[\![ Q ]\!] \searrow a'$ and $a = a' \otimes [\![ P_j ]\!]$, where $\{i, j\} = [2]$.

According to Proposition 2, we obtain $[\![ P ]\!] = [\![ P_i ]\!] \otimes [\![ P_j ]\!]$. In case $(i)$, we obtain $[\![ P ]\!] \searrow b' \otimes [\![ P_j ]\!]$, and condition *1* holds for $b = b' \otimes [\![ P_j ]\!]$. In case $(ii)$, condition *1* holds for $b = b'$. In case $(iii)$, we distinguish three sub-cases:

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

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

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

In sub-case $(a)$, we obtain $[\![ P ]\!] \searrow b' \otimes [\![ P_j ]\!]$. Therefore, condition *1* holds for $b = b' \otimes [\![ P_j ]\!]$. For $(b)$, we have $P \equiv R' \mid P_j \mid u\langle v \rangle$. By Proposition 2, $a = [\![ R' \mid P_j \mid \{v/y\}Q ]\!] \otimes \mathtt{def}_u\,(y)\,[\![ Q ]\!]$. Thus, condition *2* holds. For sub-case $(c)$, we may assume (without losing generality) that $v_k \notin \mathtt{fn}(P_j)$ for every $k \in [n]$. By Lemma 4, it follows that $v_k \notin \mathtt{surf}([\![ P_j ]\!])$ for every $k \in [n]$. Then $P \equiv \mathtt{def}\ v_1\langle t_1 \rangle \triangleright R_1\ \mathtt{in}\ \ldots\ \mathtt{def}\ v_n\langle t_n \rangle \triangleright R_n\ \mathtt{in}\ (R' \mid P_j \mid u\langle v_n \rangle)$, and

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

Thus, condition *3* holds.

– If $P$ is a definition $\mathtt{def}\ w\langle t \rangle \triangleright P_1\ \mathtt{in}\ P_2$, then we can assume without losing generality that $w \notin \mathtt{fn}(Q) \cup \{u\}$. By Lemma 4 and Lemma 2, $w \notin \mathtt{surf}(\mathtt{def}_u\,(y)\,[\![ Q ]\!])$. It follows from Proposition 2 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 2, $[\![ P_2 ]\!] \otimes \mathtt{def}_w\,(t)\,[\![ P_1 ]\!] \otimes \mathtt{def}_u\,(y)\,[\![ Q ]\!] \searrow a'$ and $a = \nu \cdot (w)a'$. Since $\mathtt{def}_w\,(t)\,[\![ P_1 ]\!] \not\searrow$ and $\mathtt{def}_u\,(y)\,[\![ Q ]\!] \not\searrow$, then $\mathtt{def}_u\,(y)\,[\![ Q ]\!] \otimes \mathtt{def}_w\,(t)\,[\![ P_1 ]\!] \not\searrow$ (according to Lemma 2). It follows that one of the following cases remains possible:

$(i)$ $[\![ P_2 ]\!] \searrow b'$ and $a' = b' \otimes \mathtt{def}_w\,(t)\,[\![ P_1 ]\!] \otimes \mathtt{def}_u\,(y)\,[\![ Q ]\!]$;
$(ii)$ $[\![ P_2 ]\!] \otimes \mathtt{def}_w\,(t)\,[\![ P_1 ]\!] \searrow b'$ and $a' = b' \otimes \mathtt{def}_u\,(y)\,[\![ Q ]\!]$;
$(iii)$ $[\![ P_2 ]\!] \otimes \mathtt{def}_u\,(y)\,[\![ Q ]\!] \searrow a''$ and $a' = a'' \otimes \mathtt{def}_w\,(t)\,[\![ P_1 ]\!]$.

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

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

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

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

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

- $v \neq w$. Then $P \equiv \mathtt{def}\ w\langle t \rangle \triangleright P_1\ \mathtt{in}\ R' \mid u\langle v \rangle$. It is easy to see that $\mathtt{surf}([\![ \{v/y\}Q ]\!]) \subseteq \{v\} \cup \mathtt{surf}([\![ Q ]\!])$, and so $w \notin \mathtt{surf}([\![ \{v/y\}Q ]\!])$. By Proposition 2, $a = [\![ \mathtt{def}\ w\langle t \rangle \triangleright P_1\ \mathtt{in}\ R' \mid \{v/y\}Q ]\!] \otimes \mathtt{def}_u\,(y)\,[\![ Q ]\!]$. Thus, condition *2* holds.
- $v = w$. Then $P \equiv \mathtt{def}\ v\langle t \rangle \triangleright P_1\ \mathtt{in}\ (R' \mid u\langle v \rangle)$. Moreover, $a = \nu \cdot (v)(\mathtt{def}_v\,(t)\,[\![ P_1 ]\!] \otimes [\![ R' \mid \{v/y\}Q ]\!] \otimes \mathtt{def}_u\,(y)\,[\![ Q ]\!])$. Thus, condition *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, we obtain

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

Thus, condition *3* holds. □

**Theorem 2.** *If* $[\![\,P\,]\!] \searrow a$, *then there exists a process* $Q$ *such that* $P \to Q$ *and* $[\![\,Q\,]\!] = a$.

**Proof.** Induction on the structure of $P$.

– If $P$ is the empty process or a message, then $[\![\,P\,]\!] \not\searrow$. Therefore, the statement of the theorem is obviously true because 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 7, one of the following cases holds:

(1) $[\![\,P_1\,]\!] \searrow a_1$ and $a = a_1 \otimes [\![\,P_2\,]\!]$;
(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 similar (symmetric).

By induction, we have $P_1 \to Q_1$ and $a_1 = [\![\,Q_1\,]\!]$. According to Proposition 1, $P \to Q_1 \mid P_2$ and $a = [\![\,Q_1\,]\!] \otimes [\![\,P_2\,]\!]$. Thus, the result of the theorem holds for $Q = Q_1 \mid P_2$.

– 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 2 that $[\![\,P_2\,]\!] \otimes \mathtt{def}_w\,(t)\,[\![\,P_1\,]\!] \searrow a'$ and $a = \nu\cdot(w)a'$. By Lemma 8, only one of the following cases holds:

(i) $[\![\,P_2\,]\!] \searrow b$ and $a' = b \otimes \mathtt{def}_w\,(t)\,[\![\,P_1\,]\!]$;
(ii) $P_2 \equiv R \mid w\langle v\rangle$ and $a' = [\![\,R \mid \{v/t\}P_1\,]\!] \otimes \mathtt{def}_w\,(t)\,[\![\,P_1\,]\!]$;
(iii) $P_2 \equiv \mathtt{def}\ w_1\langle t_1\rangle \triangleright R_1\ \mathtt{in}\ \ldots\ \mathtt{def}\ w_n\langle t_n\rangle \triangleright R_n\ \mathtt{in}\ (R \mid w\langle w_n\rangle)$ and

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

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

In case $(i)$, by the induction hypothesis, $P_2 \to Q_2$ and $b = [\![\,Q_2\,]\!]$. It follows that $P \to \mathtt{def}\ w\langle t\rangle \triangleright P_1\ \mathtt{in}\ Q_2$ and $a = \nu\cdot(w)([\![\,Q_2\,]\!] \otimes \mathtt{def}_w\,(t)\,[\![\,P_1\,]\!])$ Thus, the result of the theorem holds for $Q = \mathtt{def}\ w\langle t\rangle \triangleright P_1\ \mathtt{in}\ Q_2$.

In case $(ii)$, we have $P \to \mathtt{def}\ w\langle t\rangle \triangleright P_1\ \mathtt{in}\ (R \mid \{v/t\}P_1)$ and $a = \nu\cdot(w)([\![\,R \mid \{v/t\}P_1\,]\!] \otimes \mathtt{def}_w\,(t)\,[\![\,P_1\,]\!]) = [\![\,Q\,]\!]$. Thus, the result of the theorem holds for $Q = \mathtt{def}\ w\langle t\rangle \triangleright P_1\ \mathtt{in}\ (R \mid \{v/t\}P_1)$.

In case $(iii)$, it follows that $w_i \notin \mathtt{surf}(\mathtt{def}_w\,(t)\,[\![\,P_1\,]\!])$ for any $i \in [n]$ (Lemmas 2 and 4).

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

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

□

## 6. Describing Communication Patterns by Using the Hypergraph Model

In the Unix operating system, interprocess communications based on message queues allow exchange of information between processes. The processes exchange information by accessing a common message queue. Essentially, one process produces a message queue (via a message-passing module) that other processes may access; often a server places a message onto a queue which can be read by multiple clients. The sending process may specify its type when placing the message in a queue such that the reading processes can select the appropriate message; thus, message queues provide a way of multiplexing information from one producer to more consumers.

As example, we consider a simple system in which only one channel is used to exchange messages between the server and clients, and any message at the input of any client must appear at the output of all the clients (this is a requirement for several social networks including a chat messaging system). A type associated to each message allows a client to access the (unique) message queue for selectively reading only specific messages (in a first-in–first-out manner). We simplify the system, and consider a process $S$ working as a server and two clients $A$ and $B$. The channels `idA` and `idB` are used to indicate the type of messages from $S$ to $A$ and $B$, respectively; a channel `idS` indicates the type of messages from the clients to the server. Client $A$ uses an input channel `inA` and an output channel **outA**; client $B$ uses channels `inB` and **outB**). For a message `m` sent along the input channel `inA`, the pattern calculus process corresponding to this system is:

$$
\begin{array}{lllll}
CommSyst = & \texttt{def} & q\langle x\rangle \mid idS\langle y\rangle & \triangleright & q\langle x\rangle \mid idA\langle\_\rangle \mid q\langle x\rangle \mid idB\langle\_\rangle \ \wedge \\
 & & q\langle x\rangle \mid idA\langle y\rangle & \triangleright & outA\langle x\rangle \ \wedge \\
 & & q\langle x\rangle \mid idB\langle y\rangle & \triangleright & outB\langle x\rangle \ \wedge \\
 & & inA\langle x\rangle & \triangleright & q\langle x\rangle \mid idS\langle\_\rangle \ \wedge \\
 & & inB\langle x\rangle & \triangleright & q\langle x\rangle \mid idS\langle\_\rangle \\
 & \texttt{in} & inA\langle m\rangle\ . & &
\end{array}
$$

Using the hypergraph model, in Figure 3 is presented the net corresponding to this process.

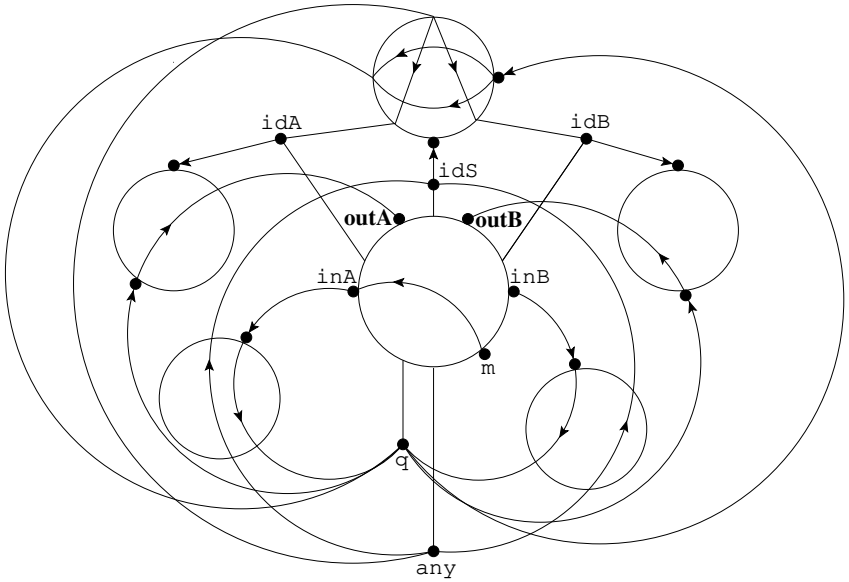



**Figure 3.** The net of a simple communication system described previously in pattern calculus.

Except the root hyperedge, the structure of this net does not change during the evolution. Therefore, the evolution of the system could be described graphically focusing only on the root hyperedge; this evolution is depicted in Figure 4.

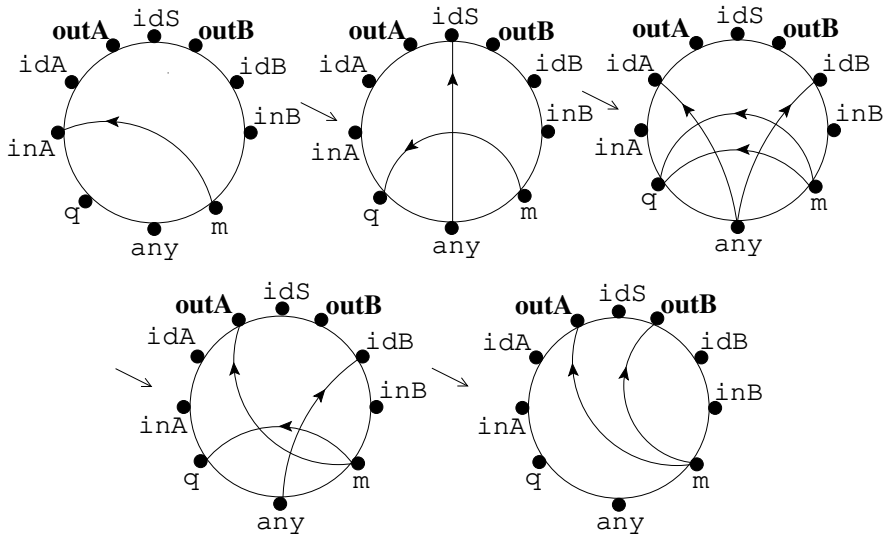



**Figure 4.** The evolution of the system (as it appears in the root hyperedge).

In Figure 4 it is not difficult to check visually the requirement that a message appearing at the input of a client appears also at the output of all the clients. In our case, the message `m` on the input channel `inA` (the initial step) appears at the output channels **outA** and **outB** in the final step described in Figure 4.

## 7. Conclusions and Related Work

In this paper we introduce a hypergraph model (given by the pattern nets) for the communication patterns. These nets provide a fully abstract model for the pattern calculus. In this way, a new sound graphical model for concurrency is introduced. We present a semantic interpretation of the pattern calculus in the framework of control structures, creating a graphical representation for the pattern calculus given by a new hypergraph model given by the pattern nets. By introducing a mapping from the control structure of pattern calculus into a set of hypergraphs, we provide a graphical model for communication patterns. It is also proved that the hypergraph model preserves the operational reductions of processes from pattern calculus and of the actions from the control structures. As an example, simple interprocess communications based on message queues inspired by the social networks are described by using our pattern nets. This example could be a first step towards more realistic scenarios in which the proposed model can be used to identify control structures supporting specific communication patterns. Future work will investigate realistic autonomic networking, mobility management, multiaccess selection, wireless and mobile networks (as they are presented in [12], for instance).

Graphical representations for process calculi highlight a new perception, providing a visual approach of concurrency and networks. According to our knowledge, just a few papers are devoted to the graphical presentations of the process calculi. We mention our previous attempts, namely the faithful $\pi$-nets [13], a graphical representation of the $\pi$-calculus machine [14], and a related approach by using jc-nets [15]. There exist also the graphical representations introduced by Robin Milner, namely action graphs and $\pi$-nets. Action graphs [16] are the graphical presentation of action calculi; they are very general, and so they are not able to describe specific features of certain action calculi. In the graphical presentation of the $\pi$-calculus given by the $\pi$-nets [6], channels are represented as rather complicated nodes called torpedos together with boxes representing guards, and messages are represented as directed arcs. The boxes obscure the internal nodes representing channels; to ensure access to the hidden channels, a rather complex additional mechanism of links is used. To avoid such a mechanism, in [13,14] the channels are represented by nodes, messages are represented by boxes of arcs and guards are represented by arcs between boxes. This approach simplified the graphical representation of the $\pi$-calculus; unfortunately, it provided

identical representations for processes with different behaviours. Fortunately, this deficiency was overtaken in the pattern calculus hypergraph model: processes with different behaviours are not mapped to the same hypergraph. The hypergraph model is presented in the same formal framework used for the $\pi$-nets (it is worth noting that hypergraph model avoids certain irrelevant aspects of $\pi$-nets). It is simpler than the $\pi$-nets, preserving much of their expressive power (according to [2], the join calculus has the same expressive power as the $\pi$-calculus). Compared with all of them, the pattern nets represent a simple but sound graphical model for concurrency, providing a fully abstract model for the pattern calculus.

**Funding:** This research received no external funding.

**Acknowledgments:** Many thanks to Mihai Rotaru for his contributions in our past collaboration.

**Conflicts of Interest:** The author declares no conflict of interest.

## References

1. Resnik, M.D. *Mathematics as a Science of Patterns*; Oxford University Press: Oxford, UK, 1999.
2. Fournet, C; Gonthier, G. The reflexive CHAM and the join calculus. In Proceedings of the 23rd ACM Symposium on Principles of Programming Languages (POPL'96), St. Petersburg, FL, USA; 21–24 January 1996; Association for Computing Machinery: New York, NY, USA, 1996; pp. 372–385. [CrossRef]
3. Levy, J.J. Some results in the join calculus. *Lect. Notes Comput. Sci.* **1997**, *1281*, 233–249.
4. Milner, R. *Communicating and Mobile Systems: The $\pi$-Calculus*; Cambridge University Press: Cambridge, UK, 1999.
5. Reisig, W. *Understanding Petri Nets*; Springer: Berlin, Germany, 2013.
6. Milner, R. $\pi$-nets: A graphical form of $\pi$-calculus. *Lect. Notes Comput. Sci.* **1994**, *788*, 26–42.
7. Schmidt, G.; Strohlein, T. *Relations and Graphs*; EATCS Monographs on Theor. Comput. Sci. Springer: Berlin, Germany, 1993.
8. Milner, R. Action calculi for syntactic action structures. *Lect. Notes Comput. Sci.* **1993**, *711*, 105–121.
9. Mifsud, A.; Milner, R.; Power, J. Control structures. In Proceedings of the 10th IEEE Symposium on Logic in Computer Science (LICS'95), San Diego, CA, USA, 26–29 June 1995.
10. Asperti, A.; Longo, G. *Categories, Types and Structures*; MIT Press: Cambridge, MA, USA, 1996.
11. Milner, R. Fully abstract models of typed $\lambda$-calculi. *Theor. Comput. Sci.* **1977**, *4*, 1–22. [CrossRef]
12. Pentikousis, K.; Blume, O.; Aguero, R.; Papavassiliou, S. *Mobile Networks and Management*; Springer: Berlin, Germany, 2010.
13. Ciobanu G.; Rotaru, M. Faithful $\pi$-nets. A graphical representation of the asynchronous $\pi$-calculus. *Electron. Notes Theor. Comput. Sci.* **1998**, *18*, 24–45. [CrossRef]
14. Ciobanu G.; Rotaru, M. A $\pi$-calculus machine. *J. Univers. Comput. Sci.* **2000**, *6*, 39–59.
15. Ciobanu G.; Rotaru, M. JC-nets. *Lect. Notes Comput. Sci.* **2001**, *2055*, 190–201.
16. Milner, R. Calculi for interaction. *Acta Inform.* **1996**, *33*, 707–737. [CrossRef]