Reification of substitutions in the asynchronous pi-calculus

Noël Bernard, Yves Dumond

Reification of substitutions in the asynchronous π-calculus

Noël Bernard[a], Yves Dumond[b]

[a] *LAMA, University of Savoie, Le Bourget-du-Lac, France*
[b] *LISTIC, University of Savoie, Le Bourget-du-Lac, France*

Abstract

In this paper, we bring in a dialect of the π-calculus, namely the π_S-calculus, which involves explicit substitutions. This mechanism has the property to handle substitutions in such a way that it avoids deep parsing of the terms concerned. Then, we show that the π_S-calculus can faithfully simulate the π-calculus, thus putting in evidence the fact that terms of the latter can be interpreted more efficiently.

Key words: concurrency, process algebra, π-calculus, explicit substitutions.

1. Introduction

Name substitution is a basic feature in the semantics of the λ-calculus and of many process calculi. For instance, let us pay attention to the following reduction rule which is part of the semantics of the π-calculus:

$$a(r).P \xrightarrow{a(v)} P\{v/r\}$$

Here, the name v is received on the channel a and it is then substituted for the free occurrences of the placeholder r in the term P. Therefore, this one must be parsed in order to perform this task. Moreover, if the process P is made of a set of processes running in parallel, e.g.:

$$P \equiv P_1 \mid \ldots \mid P_n$$

the substitutions should normally involve several communications, which are not considered in the usual semantics. Consequently, as it stands, this mechanism might appear fairly artificial. Thus, we investigate in this paper the reification of name substitution under the form of dedicated π-processes.

Explicit substitutions are well known in the λ-calculus, starting from [2]. The question of adding a similar construct in the π-calculus has not been so widely studied. The first attempt has been the π_ξ-calculus of [7]. A term in this calculus consists of a pair denoted $\xi :: P$ of an environment ξ, that is a list of equalities between names, and a process P.
A closer proposal to ours can be found in [9, 11], where processes are ordinary processes of the π-calculus with an additional construct $P[s]$, where s is a substitution, i.e. a correspondence between names in P and values.
Explicit fusions [8] adopt a point of view very close to that in this paper: they present an equality $x = y$ as a process in itself. The main difference is the use of structural equivalences such as symmetry and transitivity relations between equalities, which amount to neglect of the computational aspect of explicit substitutions.
The Applied π-calculus [3, 1] also introduces processes $\{M/x\}$ but unlike the present paper these processes act globally on a process P not taking into account the computational content of deep propagation of substitution inside P.

2. Syntax

The syntax of the π-calculus widely differs from one author to another. Our point of view about this calculus is strongly influenced by the Chemical Abstract Machine [5]. This leads us to choose an asynchronous version of the π-calculus [6, 10] involving *guarded sums* as the reference dialect. In addition, we define *explicit substitutions* as a specific kind of process:

Definition 1 (π_S-calculus syntax). *The following is an inductive definition of the syntax of the π_S-calculus:*

$$P ::= \bar{a}\,b \mid S \mid P|P \mid (\nu\, x)P \mid [x = y]P \mid D(\vec{w}) \mid \{\!\{r \leftarrow v\}\!\} \mid \mathbf{0}$$

the syntactic category S denoting "guarded sums":

$$S ::= a(x).P \mid S + S$$

each constant invocation $D(\vec{w})$ being associated to a defining equation $D(\vec{w}) \stackrel{def}{=} P$, such that all the names that occur free in P must appear in $\vec{w}$.

The π_S-terms (also called π_S-processes) have to fulfill the Barendregt's hygiene condition (Bhc) [4] *which prohibits name capture: for any two occurrences of a given name, either both are free or both are bound in the same binding.*

The set of the π_S-terms is denoted Π_S. Moreover, in the explicit substitution $\{\!\{r \leftarrow v\}\!\}$, the name r is called the repository while v is called the value.

With regard to the Bhc, it is important to point out the fact that such a property can only be managed *globally*. Therefore, in this paper we study the derivation of processes considered individually but with the proviso that they are embedded in a syntactical context under the form of a π_S-process that complies with the Bhc.

3. Action of explicit substitutions

The *outer names* are names which, with regard to their location in a given π_S-term, are potentially subject to the action of explicit substitutions.

Definition 2 (outer names). *Let P and P' be two π_S-terms. The set $O(P)$ of outer names of P is defined by induction on P by the following list of equations:*

- $O(\mathbf{0}) = \emptyset$
- $O(\{\!\{r \leftarrow v\}\!\}) = \emptyset$
- $O(\bar{a}b) = \{a, b\}$
- $O(a(x).P) = \{a\}$
- $O(P|P') = O(P) \cup O(P')$
- $O(S + S') = O(S) \cup O(S')$
- $O((\nu\, x)P) = O(P) - \{x\}$
- $O([x = y]P) = O(P) \cup \{x, y\}$
- $O(D(\vec{v})) = \{v_1, \ldots, v_n\}$ *with* $\vec{v} = (v_1, \ldots, v_n)$

Definition 3 (occurrences in outer name position). *Let P be a π_S-term. An occurrence x of an element of $O(P)$ is said to be in outer name position if and only if x is not in the scope of an input prefix.*
We denote $Oc(P)$ the set of occurrences which are in outer name position in P.

Remark 1. For any π_S-term P, $O(P)$ is finite. Moreover, the outer names of P can be found in one step starting from the root of the syntactical tree related to this term and consequently any deep parsing of P is not necessary. This appears as a direct consequence of the syntax chosen for π_S-terms, in particular the fact that sums must be guarded.

Now, we have to discriminate among the explicit substitutions, those which, due to their location in a given term, are in a position to act on outer names. These are called *active* explicit substitutions:

Definition 4 (active explicit substitutions). *Let P be a π_S-term. We call active explicit substitutions of P the explicit substitutions embedded in P which are not in the scope of an input prefix.*

Definition 5 (set of repositories). *Let P be a π_S-term. We denote $\rho(P)$ the set of the repositories of the active explicit substitutions embedded in P.*

The semantic rules below give the details of implementation of the effect of active explicit substitutions on outer names as a function of the morphology of the terms concerned. This specification involves a meta-operator denoted by ">", the corresponding meta-transitions, which are obviously not performed in the π_S-calculus, being all labeled by the symbol "τ_k".

$$\frac{-}{\{\!\{r \leftarrow v\}\!\} \succ r(x).P \xrightarrow{\tau_k} v(x).P}\ (a_1) \qquad \frac{r \in \{a, b\}}{\{\!\{r \leftarrow v\}\!\} \succ \bar{a}b \xrightarrow{\tau_k} \bar{a}b\{v/r\}}\ (a_2)$$

$$\frac{\{\!\{r \leftarrow v\}\!\} \succ P \xrightarrow{\tau_k} P' \ \wedge\ x \neq r}{\{\!\{r \leftarrow v\}\!\} \succ (\nu\, x)P \xrightarrow{\tau_k} (\nu\, x)P'}\ (a_3) \qquad \frac{r \in \vec{w} \ \wedge\ \vec{w'} = \vec{w}\{v/r\}}{\{\!\{r \leftarrow v\}\!\} \succ D(\vec{w}) \xrightarrow{\tau_k} D(\vec{w'})}\ (a_4)$$

$$\frac{\{\!\{r \leftarrow v\}\!\} \succ S_1 \xrightarrow{\tau_k} S_1' \ \wedge\ r \notin O(S_2)}{\{\!\{r \leftarrow v\}\!\} \succ S_1 + S_2 \xrightarrow{\tau_k} S_1' + S_2}\ (a_5)$$

$$\frac{\{\!\{r \leftarrow v\}\!\} \succ S_1 \xrightarrow{\tau_k} S_1' \ \wedge\ \{\!\{r \leftarrow v\}\!\} \succ S_2 \xrightarrow{\tau_k} S_2'}{\{\!\{r \leftarrow v\}\!\} \succ S_1 + S_2 \xrightarrow{\tau_k} S_1' + S_2'}\ (a_6)$$

$$\frac{\{\!\{r \leftarrow v\}\!\} \succ P \xrightarrow{\tau_k} P'}{\{\!\{r \leftarrow v\}\!\} \succ [x = y]P \xrightarrow{\tau_k} [x = y]\{v/r\}P'}\ (a_7)$$

$$\frac{r \in \{x, y\} \ \wedge\ r \notin O(P)}{\{\!\{r \leftarrow v\}\!\} \succ [x = y]P \xrightarrow{\tau_k} [x = y]\{v/r\}P}\ (a_8)$$

$$\frac{\{\!\{r \leftarrow v\}\!\} \succ P \xrightarrow{\tau_k} P'}{\{\!\{r \leftarrow v\}\!\} \succ Q \,|\, P \,|\, R \xrightarrow{\tau_k} Q \,|\, P' \,|\, R}\ (a_9)$$

These rules deserve the following comments:

(a_1) The process P remains unchanged.

(a_3) Because of the Bhc, the names r and x are supposed to be different.

(a_5) This rule is written up to left-right symmetry.

(a_6) Occurrences of the repository r are substituted both in S_1 and in S_2.

4. Semantics

The operational semantics for the π_S-calculus is given hereafter under the form of a list of conditional term rewriting rules. The rules b_1, b_2 and b_3 are specific to the π_S-calculus. The other rules have the form which they take in usual specifications of the semantics of the asynchronous π-calculus apart from the side conditions. Indeed, while giving the π-calculus a transition semantics, side conditions are associated to some rules with the goal of imposing α-conversions which avoid name capture. There is no need to take such precautions in the π_S-calculus since the Bhc radically prevents this phenomenon:

$$\frac{\{\!\{r \leftarrow v\}\!\} \succ P \xrightarrow{\tau_k} P' \ \wedge \ r \neq v \ \wedge \ r \in O(P)}{\{\!\{r \leftarrow v\}\!\} \mid P \xrightarrow{\tau_s} \{\!\{r \leftarrow v\}\!\} \mid P'} \ (b_1)$$

$$\frac{-}{a(r).P \xrightarrow{a(v)} (\nu\, r)(\{\!\{r \leftarrow v\}\!\} \mid P)} \ (b_2)$$

$$\frac{D(\vec{u}) \stackrel{def}{=} P \ \wedge \ P^{\alpha}\{\vec{v}/\vec{u}\} \xrightarrow{\gamma} P'}{D(\vec{v}) \xrightarrow{\gamma} P'} \ (b_3)$$

$$\frac{-}{\bar{a}b \xrightarrow{\bar{a}b} \mathbf{0}} \ (b_4) \qquad \frac{P \xrightarrow{\gamma} P'}{[a = a]P \xrightarrow{\gamma} P'} \ (b_5)$$

$$\frac{S_1 \xrightarrow{\gamma} S_1'}{S_1 + S \xrightarrow{\gamma} S_1'} \ (b_6) \qquad \frac{P \xrightarrow{\gamma} P'}{P \mid Q \xrightarrow{\gamma} P' \mid Q} \ (b_7)$$

$$\frac{P \xrightarrow{\gamma} P' \ \wedge \ x \notin \gamma}{(\nu\, x)P \xrightarrow{\gamma} (\nu\, x)P'} \ (b_8) \qquad \frac{P \xrightarrow{\bar{a}b} P' \ \wedge \ Q \xrightarrow{a(b)} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'} \ (b_9)$$

$$\frac{P \xrightarrow{\bar{a}x} P' \ \wedge \ a \neq x}{(\nu\, x)P \xrightarrow{\bar{a}(x)} P'} \ (b_{10}) \qquad \frac{P \xrightarrow{\bar{a}(x)} P' \ \wedge \ Q \xrightarrow{a(x)} Q'}{P \mid Q \xrightarrow{\tau} (\nu\, x)(P' \mid Q')} \ (b_{11})$$

These rules deserve the following comments:

(b_1) This rule, which is written up to left-right symmetry, specifies the effect of an active explicit substitution on processes of another kind. No action is possible on explicit substitutions since they have no outer names. The corresponding transition, performed in the π_S-calculus, is labeled by the special action τ_s. The condition $r \neq v$ prevents infinite derivations of the form $\{\!\{r \leftarrow v\}\!\} \mid P \xrightarrow{\tau_s} \{\!\{r \leftarrow v\}\!\} \mid P \xrightarrow{\tau_s} \ldots$ Furthermore, this rule stresses the fact that explicit substitutions are *permanent* processes, i.e. they do not disappear, nor are modified, after an interaction with another process.

(b_2) When the value v is received, the explicit substitution $\{\!\{r \leftarrow v\}\!\}$ is created and this one runs in parallel with the process P. Both the new explicit substitution and the process P are in the scope of the restricted name r which was formerly a placeholder.

(b_3) The Bhc must be preserved by constant unfolding. Hence, all the bound names of P are first replaced by fresh names, the corresponding process, denoted P^{α}, being obviously not unique. Then, actual parameters are substituted for formal ones. Once the corresponding process $P^{\alpha}\{\vec{v}/\vec{u}\}$ has been built up, the transition γ can be performed.

(b_6), (b_7), (b_9), (b_{11}) These rules are written up to left-right symmetry.

(b_{10}) The label $\bar{a}(x)$ denotes the emission of the restricted name x on the channel a.

Proposition 1. *The Bhc property is preserved in the π_S-calculus.*

Proof. Consider a π_S-term P. We reason by cases on the action γ performed by P, i.e. $P \xrightarrow{\gamma} P'$:

– Case $\gamma = \bar{a}b$ and γ is not combined with a constant unfolding. The emitting process $\bar{a}b$ is replaced by $\mathbf{0}$ in P' and the status of the remaining names is unchanged.

– Case $\gamma = \bar{a}(b)$ and γ is not combined with a constant unfolding. Again, the emitting process $\bar{a}b$ is replaced by $\mathbf{0}$ in P'. In addition, the concerned restriction statement $(\nu\, b)$ is removed from P. Therefore, b becomes free in P' but remains different from the bound names of P'. Moreover, the status of the other names of P' is unchanged.

– Case $\gamma = a(v)$ and γ is not combined with a constant unfolding. Then, P encompasses a process of the form $S' + a(r).Q + S''$. This one is turned into $(\nu\, r)(\{\!\{r \leftarrow v\}\!\} \mid Q)$ in P'. Hence, the name r is a placeholder before the transition and becomes a restricted name after it. Therefore, it remains a bound name different from the other bound and free names of P', including the incoming name v (because of the Bhc). Moreover, the status of the other names remaining in P' is unchanged.

– Case $\gamma = \tau$, γ is not combined with a constant unfolding and the exchanged name is free. This case is a simple combination of the first and the third ones.

– Case $\gamma = \tau$, γ is not combined with a constant unfolding and the exchanged name b is bound. The process P is subject to the following modifications:

- The emitting process $\bar{a}b$ is replaced by $\mathbf{0}$.
- The scope of the restricted name b is extended and no α-conversion is required since name capture is not possible in π_S.
- A new explicit substitution is created and a placeholder is turned into a restricted name.

Again, the status of all the names remaining in P' is not modified.

– Case $\gamma = \tau_s$ and γ is not combined with a constant unfolding. The consequence of a τ_s-action consists in substituting one or several occurrences of a given repository which stand in outer name position in P, by the corresponding value. This does not bring in any new name nor modifies the status of the names remaining in P'.

– Case γ is combined with a constant unfolding. The process P encompasses a constant invocation $D(\vec{v})$ and a constant defining equation, i.e. $D(\vec{u}) \stackrel{def}{=} T$, is associated to the process P. First, fresh names are substituted for all the bound names of T: this generates a term T^α. By definition, all the names that occur free in T appear in $\vec{u}$. Then, these ones are replaced in T^α by the names of $\vec{v}$, i.e. by names already present in P. Let us denote Q the process P in which the term $T^\alpha\{\vec{v}/\vec{u}\}$ has been substituted for $D(\vec{v})$. Then, the names of $\vec{v}$ occur free in $T^\alpha\{\vec{v}/\vec{u}\}$ and they have in Q the status, i.e. bound or free, that they had in P. Moreover, the only names that have appeared in Q by the fact of the unfolding process of the constant D are the fresh names. The latter are bound in $T^\alpha\{\vec{v}/\vec{u}\}$, and consequently in Q, and they differ, by construction, from all the other names of Q. Furthermore, the status of the latter, in particular that of the names of $\vec{v}$, has not been modified by the constant unfolding. Thence, Q fulfills the Bhc. At this stage, the action γ can be performed, and we are led to consider one of the previous cases.

5. Reduction of π-terms in π_S

From now on, we focus on derivations carried out in the π_S-calculus under the assumption that they start from π-terms. This criterion characterizes a specific subset among the π_S-terms. In particular, all the explicit substitutions present in these terms are active: this comes from the fact that they are the result of the application of the rule b_2.

Definition 6 (π'_S-calculus). *We denote Π'_S the set of π_S-terms which can be obtained by derivation of π-terms and we call π'_S the calculus made of Π'_S outfitted with the semantic rules of π_S.*

The next definition characterizes terms on which rule b_3 can no longer be applied.

Definition 7 (enacted processes). *Let P be a π'_S-term which encompasses the following explicit substitutions $\{\!\{r_1 \leftarrow v_1\}\!\}, \ldots, \{\!\{r_n \leftarrow v_n\}\!\}$. The process P is said to be enacted if and only if $O(P) \cap \rho(P) = \emptyset$.*

Definition 8 (enactment phase). *Let P be a π'_S-term. Any (possibly empty) sequence of τ_s-actions, due to the action of the explicit substitutions of P, which turns it into an enacted process P' is called an enactment phase. The corresponding derivation is denoted $P \rightsquigarrow P'$.*

We define hereafter a property which is fulfilled by the set of explicit substitutions of π'_S-terms provided the latter are reduced according to a given strategy. This property is subsequently used to establish the termination and the confluence of the enactment phase.

Definition 9 (disjunction property). *Let P be a π'_S-term which encompasses the following explicit substitutions $\{\!\{r_1 \leftarrow v_1\}\!\}, \ldots, \{\!\{r_n \leftarrow v_n\}\!\}$. The term P is said to fulfill the disjunction property on repositories and values if and only if:*

- *all the repositories r_i are distinct,*
- *no value v_i is equal to a repository r_j .*

We denote $\mathcal{D}(P)$ the fact that P abides by this property.

Lemma 1. *Let P be a π'_S-term that derives from a π-term P_0. Then, $\mathcal{D}(P)$ holds provided along the derivation that leads from P_0 to P, all the actions different from τ_s have been performed by enacted processes.*

Proof. Consider a π'_S-term P that encompasses the list of explicit substitutions $\{\!\{r_1 \leftarrow v_1\}\!\}, \ldots, \{\!\{r_n \leftarrow v_n\}\!\}$. We reason by induction on the length of a derivation path that complies with the aforementioned constraints.

Base case : no explicit substitution is present in P_0 and consequently $\mathcal{D}(P_0)$ is verified.

Induction step : by induction hypothesis, $\mathcal{D}(P)$ holds. According to the semantics of the π'_S-calculus, active explicit substitutions are permanent processes, i.e. they cannot disappear nor be modified. Therefore, the only transitions which are worth considering are those that generate a new explicit substitution: this expels the actions $\bar{a}v$, $\bar{a}(v)$ and τ_s. The mechanism of constant unfolding does not generate new explicit substitutions, either. Hence, in the proof below, the two actions, i.e. $a(v)$ and τ, are considered with the proviso that they are not combined with a constant unfolding. Thus, if we denote $\{\!\{r \leftarrow v\}\!\}$ the explicit substitution which has appeared in P', then for all explicit substitution $\{\!\{r_i \leftarrow v_i\}\!\}$ of P, we have to prove that $r \neq r_i$, $r \neq v_i$, $r \neq v$ and $r_i \neq v$. We proceed by cases on the action γ performed by P, i.e. $P \stackrel{\gamma}{\longrightarrow} P'$:

– Case $\gamma = a(v)$:

- $r \neq r_i$. The repository r_i either is a placeholder present in the original term P_0 or a fresh name that has appeared due to a constant unfolding. In both cases, because of the Bhc, $r \neq r_i$.

- $r \neq v_i$. We reason *ad absurdum*. Assume that for some i, $r = v_i$. Then, P necessarily encompasses:
  - a process $S' + a(r).Q + S''$, which becomes $(\nu\, r)(\{\!\{r \leftarrow v\}\!\} \mid Q)$ in P',
  - an explicit substitution $\{\!\{r_i \leftarrow r\}\!\}$ which, by hypothesis, cannot be embedded in Q since in π'_S-terms, all the explicit substitutions are active.

  Here, Q is the scope of the placeholder r and this name also occurs outside Q, namely in the explicit substitution $\{\!\{r_i \leftarrow r\}\!\}$. This is clearly in contradiction with the Bhc.
- $r \neq v$. Again, we reason *ad absurdum*. Assume that $r = v$. Then, P necessarily encompasses a process $S' + a(v).Q + S''$ which becomes $(\nu\, v)(\{\!\{v \leftarrow v\}\!\} \mid Q)$ in P'. Here, the incoming name v appears in P as a placeholder and outside P in an emitting process of the form $\bar{a}v$. This is in contradiction with the Bhc, to which the context encompassing P is supposed to comply.
- $r_i \neq v$. Here, we have to prove that the incoming name v cannot be equal to one of the repositories r_i. Each of them either was a placeholder in the term P_0 or has appeared as a fresh name during a constant unfolding. In both cases, it could not occur anywhere else. So, it can be received by P only if it has been previously sent out by a process from which P derives, and subsequently sent back to P. Consider the explicit substitution $\{\!\{r_i \leftarrow v_i\}\!\}$: because of

  the disjunction property, $r_i \neq v_i$. Under this assumption, any action $\bar{a}r_i$ or $\bar{a}(r_i)$ is prohibited because the encompassing process is not enacted: the value v_i must be substituted for r_i before the emission. Therefore, the received value v cannot be equal to one of the repositories of P.

– Case $\gamma = \tau$ and the emitted name is free:
- $r \neq r_i$ and $r \neq v_i$. We can reason in the same way as in case $\gamma = a(v)$: the only difference is that the emitting process is embedded in P.
- $r \neq v$. We reason *ad absurdum*. Assume that $r = v$. Then, P necessarily encompasses:
  - a process $\bar{a}v$,
  - a process $S' + a(v).Q + S''$ which becomes $(\nu\, v)(\{\!\{v \leftarrow v\}\!\} \mid Q)$ in P'.

  Moreover, the emitting process $\bar{a}v$ is obviously not embedded in Q. Hence, the name v occurs in Q as a placeholder and outside Q in the emitting process, which is in contradiction with the Bhc.
- $r_i \neq v$. Again, we reason *ad absurdum*. Assume that for some i, $r_i = v$. Then, P necessarily encompasses:
  - a process $\bar{a}v$,
  - a process $S' + a(r).Q + S''$, which becomes $(\nu\, r)(\{\!\{r \leftarrow v\}\!\} \mid Q)$ in P',
  - an active explicit substitution $\{\!\{v \leftarrow v_i\}\!\}$ which is not embedded in Q.

  Since $\mathcal{D}(P)$ holds, we have $v \neq v_i$. Here, the contradiction comes from the fact that the action τ cannot be performed because P is not enacted: v_i must be substituted for v in the process $\bar{a}v$ prior to this action. As a result, the repository v cannot be emitted and ergo cannot become the value of the newly created explicit substitution.

– Case $\gamma = \tau$ and the emitted name is bound, i.e. is a restricted name:
- $r \neq r_i$, $r \neq v_i$ and $r \neq v$. We can reason as we did in the previous case.
- $r_i \neq v$. We reason *ad absurdum*. Assume that for some i, $r_i = v$. Then, P necessarily encompasses:
  - a process $(\nu\, v)(R' \mid \bar{a}v \mid R'')$,
  - a process $S' + a(r).Q + S''$, which becomes $(\nu\, r)(\{\!\{r \leftarrow v\}\!\} \mid Q)$ in P',
  - an explicit substitution $\{\!\{v \leftarrow v_i\}\!\}$ which, because of the Bhc, is necessarily embedded in R' or R'' and such that $v \neq v_i$.

  Again, we can argue that the τ-action cannot be performed if P is not enacted: the name v_i has to be substituted for v before the emission.

Remark 2. If the π'_S-term P_0 has been derived up to P according to the hypotheses made in the terms of Lemma 1, then $\mathcal{D}(P)$ is verified whether P is enacted or not.

Lemma 2 (finiteness of enactment phase). *Let P be a π'_S-term such that $\mathcal{D}(P)$ holds. Then, any enactment phase of P is finite.*

Proof. If P is enacted, then the enactment phase is empty. Now, assume that P is a non enacted π'_S-process. The outer names which are affected during the enactment phase are the elements of the finite set $\mathcal{E} = O(P) \cap \rho(P)$. By the fact of the disjunction property, each element of $\mathcal{E}$ can be subject to the action of one, and one only, explicit substitution. Once the substitution(s) has (have) been performed, the names(s) appearing at the corresponding location(s) cannot be modified

anymore since no value is equal to a repository. Therefore, if $nb{-}occ(x, P)$ denotes the number of occurrences of x in outer names position in P, we can argue that the integer value $\sum_{e\in\mathcal{E}} nb-occ(e, P)$ strictly decreases, up to zero, at each τ_s-step.

Lemma 3 (confluence of enactment phase). *Let P be a π'_S-term such that $\mathcal{D}(P)$ holds. Then, the enactment phase of P has the property of confluence.*

Proof. Obvious: due to the disjunction property, any occurrence of an element of the set $\mathcal{E} = O(P)\cap\rho(P)$is subject to the action of one, and one only, explicit substitution. Hence, any τ_s-action is performed in total causal independency towards the other.

6. Simulation of π by π'_S

We show in this section that any π-term can be derived, respectively in π and in π'_S, in such a way that, at each step, the terms reached are closely related, thus putting in evidence a relation of simulation between the derivations they respectively belong to. This clearly shows the possibility to interpret π-terms in an efficient way, avoiding the parsing phases of terms made necessary by *in-depth* substitutions.
To begin with, we need to define the two calculi, one compared to the other:

- The Bhc is assumed to be implemented both in π and in π'_S.
- The only difference between the two syntaxes resides in the fact that in π'_S, explicit substitutions do not exist.
- From a semantic point of view the relationship between the two calculi is specified hereafter. Thus, for what concerns the π-calculus:
  – Substitutions are performed *in-depth* after each reception, so there is no need for a meta-operation handling them. Consequently, the rule b_1 must be removed and the input rule b_2 has to be written under its usual form:
  $$a(r).P \xrightarrow{a(v)} P\{v/r\}$$
  – All the other semantic rules, in particular b_3 which preserves the Bhc, are unchanged.

At this point, we introduce a notion that aims at defining a morphological relation between π- and π'_S-terms. The latter are characterized by the presence of explicit substitutions and of restriction statements $(\nu\, r_i)$ related to the repositories. Hence, these features have to be removed in order to make the comparisons possible between terms of the two calculi.

Definition 10 (concrete term). *If P is a π'_S-term, we denote $\langle P\rangle$ the π-term defined by the following set of axioms, where $\equiv$ is the syntactic identity between terms:*

- $\langle P\,|\ldots|\,Q\rangle \equiv \langle P\rangle|\ldots|\langle Q\rangle$
- $\langle a_1(x_1).P_1 + \ldots + a_n(x_n).P_n\rangle$
  $\equiv\ a_1(x_1).\langle P_1\rangle + \ldots + a_n(x_n).\langle P_n\rangle$
- $\exists v,\ \{\!\{r\leftarrow v\}\!\}$ *is a subprocess of* P
  $\Rightarrow \langle(\nu\, r)P\rangle \equiv \langle P\rangle$
- $\nexists v,\ \{\!\{r\leftarrow v\}\!\}$ *is a subprocess of* P
  $\Rightarrow \langle(\nu\, r)P\rangle \equiv (\nu\, r)\langle P\rangle$
- $\langle(\{\!\{r\leftarrow v\}\!\}\,|\,P)\rangle \equiv \langle P\rangle$
- $\langle[x = y]P\rangle \equiv [x = y]\langle P\rangle$
- $\langle\bar{a}b\rangle \equiv \bar{a}b$
- $\langle D(\vec{w})\rangle \equiv D(\vec{w})$
- $\langle\boldsymbol{0}\rangle \equiv \boldsymbol{0}$

We bring in below the simulation relation considered in this paper, as well as some related notions:

Definition 11 (name occurrence location). *Let x be an occurrence of a name n in the π'_S-term P. We call location of x and denote $loc(x, P)$ the number of symbols preceding x in P, the names and the constant identifiers being counted as 1.*

Definition 12 ($\hookrightarrow$-transition). *If P and Q are two π'_S-terms and γ an action different from τ_s, any derivation of the form $P\xrightarrow{\gamma} P' \rightsquigarrow Q$ is denoted $P\overset{\gamma}{\hookrightarrow} Q$.*

Corollary 1. *Let P_0 be a π-term and consider the following derivation $P_0\overset{\gamma_0}{\hookrightarrow} \ldots \overset{\gamma_{n-1}}{\hookrightarrow} P_n$ performed in π'_S. Then, for all i, P_i is enacted and $\mathcal{D}(P_i)$ holds.*

Proof. Obvious since the above derivation of P_0 is in conformity with the reduction strategy defined by the terms of Lemma 1.

Definition 13 ($\mathcal{M}$-equivalence). *Let P be a π-term and Q an enacted π'_S-term such that $\mathcal{D}(Q)$ holds. Consider the list $\{\!\{r_1 \leftarrow v_1\}\!\}, \ldots, \{\!\{r_n \leftarrow v_n\}\!\}$ of explicit substitutions encompassed by Q. Then, the terms P and Q are said to be $\mathcal{M}$-equivalent if and only if:*

$\mathcal{M}$**1** $P \equiv \langle Q\rangle\{v_1/r_1\}\ldots\{v_n/r_n\}$

$\mathcal{M}$**2** $(x \in Oc(P)\ \wedge\ x$ *is an occurrence of* n
$\Rightarrow \exists x' \in Oc(Q)$ *s.t.* :
$(x'$ *is an occurrence of* $n\ \wedge loc(x, P) = loc(x', \langle Q\rangle)))$
and conversely

The $\mathcal{M}$-equivalence is denoted $\sim$.

Definition 14 (σ-simulation). *A relation $\asymp$ is said to be a σ-simulation of π by π'_S if for any derivation $\Delta : P \equiv P_0 \xrightarrow{\gamma_0} P_1 \xrightarrow{\gamma_1} \ldots \xrightarrow{\gamma_{n-1}} P_n$ in π there exists a derivation $\Delta' : P \equiv Q_0 \xhookrightarrow{\gamma_0} Q_1 \xhookrightarrow{\gamma_1} \ldots \xhookrightarrow{\gamma_{n-1}} Q_n$ in π'_S such that for all i, $P_i \asymp Q_i$.*

Remark 3. At first sight, one may be surprised that we consider a simulation relation between derivations and not between terms as it is usually done. In fact, by explicitly considering derivations of the process P as a whole, we deal with terms which encompass all the required explicit substitutions.

Proposition 2. *The relation $\sim$ is a σ-simulation of π by π'_S.*

Proof. We reason by induction on the length of a derivation path. Moreover, in the proof below, any expression of the form $[\vec{x} = \vec{x}]$ denotes a list of adjacent matching statements $[x_1 = x_1] \ldots [x_n = x_n]$.

Base case : The π-terms P, P_0 and Q_0 are identical, i.e. $P \equiv P_0 \equiv Q_0$ and the set of explicit substitutions encompassed by Q_0 is empty. Thence $P_0 \sim Q_0$.

Induction step : We reason by cases on the transition γ_n undergone by the π-term P_n, i.e. $P_n \xrightarrow{\gamma_n} P_{n+1}$:

– Case $\gamma_n = \bar{a}b$ and γ_n is not combined with a constant unfolding. By induction hypothesis, $P_n \sim Q_n$. Therefore, the processes P_n and Q_n are such that they both encompass an emitting process of the form $[\vec{x} = \vec{x}]\bar{a}b$, the location of the corresponding occurrences of a being respectively the same in P_n and in $\langle Q_n \rangle$. Hence, there exists a π'_S-term Q_{n+1} for which $Q_n \xrightarrow{\bar{a}b} Q_{n+1}$. Moreover, P_{n+1} and Q_{n+1} can be respectively deduced from P_n and Q_n by replacing the process $[\vec{x} = \vec{x}]\bar{a}b$ by $\mathbf{0}$. By induction hypothesis, Q_n is enacted and since no new explicit substitution has appeared in Q_{n+1}, this one is enacted too. Consequently, the π'_S-term Q_{n+1} is such that $Q_n \xhookrightarrow{\bar{a}b} Q_{n+1}$. Furthermore, under the assumption that $P_n \equiv \langle Q_n \rangle \{v_1/r_1\} \ldots \{v_n/r_n\}$, if we respectively replace $[\vec{x} = \vec{x}]\bar{a}b$ by $\mathbf{0}$ in P_n and in Q_n, the location of all the occurrences of the remaining names is either unchanged or shifted symmetrically in P_n and in $\langle Q_n \rangle$. This is in particular true for:

- the occurrences of the different repositories r_i and ergo $P_{n+1} \equiv \langle Q_{n+1} \rangle \{v_1/r_1\} \ldots \{v_n/r_n\}$,
- the occurrences in outer name position which remain identical, at any concerned location, respectively in P_{n+1} and $\langle Q_{n+1} \rangle$.

In conclusion, $\mathcal{M}.1$ and $\mathcal{M}.2$ hold and thence $P_{n+1} \sim Q_{n+1}$.

– Case $\gamma_n = \bar{a}(b)$ and γ_n is not combined with a constant unfolding. Here, the reasoning is basically the same as that of the previous case. The only difference lies in the fact that in addition to the emitting process, a restriction statement $(\nu\, b)$ disappears from P_n and Q_n as well.

– Case $\gamma_n = a(v)$ and γ_n is not combined with a constant unfolding. The process P_n encompasses a process of the form $[\vec{y} = \vec{y}](S' + a(\rho_i).P_n^i + S'')$. By induction hypothesis, $P_n \sim Q_n$ and consequently Q_n encompasses a process of the form $[\vec{y} = \vec{y}](T' + a(\rho_i).Q_n^i + T'')$ for which:

- Q_n^i is a π-term such that $P_n^i \equiv Q_n^i\{v_1/r_1\} \ldots \{v_n/r_n\}$,
- the location of the corresponding occurrences of a is the same in P_n and in $\langle Q_n \rangle$,
- $S' \sim T'$ and $S'' \sim T''$.

After the reception of the value v on the channel a, the two receiving processes are respectively replaced by:

- $P_n^i\{v/\rho_i\}$ in P_{n+1}, i.e. the substitution $\{v/\rho_i\}$ is performed *in-depth* in the process P_n^i,
- $(\nu\, \rho_i)(\{\!\{\rho_i \leftarrow v\}\!\} \mid Q_n^i)$ in the immediate derivative of Q_n here denoted Q, i.e. a new explicit substitution $\{\!\{\rho_i \leftarrow v\}\!\}$ is created that runs in parallel with the process Q_n^i, the whole being in the scope of the restricted name ρ_i.

By induction hypothesis, $\mathcal{D}(Q_n)$ holds. Thence, after Lemma 1 $\mathcal{D}(Q)$ does too. Moreover, after Lemma 2 and Lemma 3 the enactment phase of Q is finite and confluent. Therefore, there exists a unique enacted π'_S-term Q_{n+1} for which $Q \rightsquigarrow Q_{n+1}$ and $Q_n \xhookrightarrow{a(v)} Q_{n+1}$. From a morphological point of view, the processes Q and Q_{n+1} do not differ since the latter can be deduced from the former by a finite set of τ_s-actions. In fact, Q is subject to the action of the previously existing explicit substitutions, i.e. $\{\!\{r_1 \leftarrow v_1\}\!\}, \ldots, \{\!\{r_n \leftarrow v_n\}\!\}$ plus that of the new created one $\{\!\{\rho_i \leftarrow v\}\!\}$. Consequently, at the end of the enactment phase, the process Q_n^i has been turned into a process Q_{n+1}^i such that $P_n^i\{v/\rho_i\} \equiv Q_{n+1}^i\{v_1/r_1\} \ldots \{v_n/r_n\}\{v/\rho_i\}$, which, by the fact that $\langle (\nu\, \rho_i)(\{\!\{\rho_i \leftarrow v\}\!\} \mid Q_{n+1}^i) \rangle \equiv Q_{n+1}^i$, leads to:

$$P_n^i\{v/\rho_i\} \equiv \langle (\nu\, \rho_i)(\{\!\{\rho_i \leftarrow v\}\!\} \mid Q_{n+1}^i) \rangle \{v_1/r_1\} \ldots \{v_n/r_n\}\{v/\rho_i\}$$

By hypothesis, Q_n is an enacted process and because of the scope definition of ρ_i, occurrences of its own cannot be found outside Q_n^i. Hence, the enactment phase of Q only affects the process Q_n^i. From this, we deduce:

$$P_{n+1} \equiv \langle Q_{n+1} \rangle \{v_1/r_1\} \ldots \{v_n/r_n\}\{v/\rho_i\}$$

So, $\mathcal{M}.1$ holds.

From $P_n^i\{v/\rho_i\} \equiv Q_{n+1}^i\{v_1/r_1\} \ldots \{v_n/r_n\}\{v/\rho_i\}$ and by the fact that Q_{n+1}^i is no longer subject to the action of any explicit

substitution, we can deduce that for any concerned location, the occurrences of names in outer name position are identical in $P_n^i\{v/\rho_i\}$ and in Q_{n+1}^i. Furthermore, the processes that appear in the following couples, i.e. (S', T'), (S'', T'') and $(P_n^i\{v/\rho_i\},\ Q_{n+1}^i)$ being all respectively made of the same number of symbols, the location of the occurrences of outer names present in P_{n+1} and in $\langle Q_{n+1}\rangle$ outside $P_n^i\{v/\rho_i\}$ and Q_{n+1}^i are either unchanged or shifted symmetrically. Therefore, $\mathcal{M}.2$ holds.

Again, we can assert that $P_{n+1} \sim Q_{n+1}$.

– Case $\gamma_n = \tau$ and γ_n is not combined with a constant unfolding. If the name subject to the communication is free, the present case is a simple combination of the cases $\gamma_n = \bar{a}b$ and $\gamma_n = a(b)$. If the emitted name is bound, i.e. is a restricted name, the reasoning is globally similar, the only difference lying in the fact that a scope extrusion must be considered, the restriction statement $(\nu\, b)$ remaining in P_{n+1} and Q_{n+1}.

– Case γ_n is combined with a constant unfolding. By induction hypothesis, P_n and Q_n respectively encompass, at the same location in P_n and in $\langle Q_n\rangle$, an occurrence of the same process $D(\vec{u})$. This assumes the existence of a constant defining equation $D(\vec{w}) \stackrel{def}{=} T$. At this point, consider that:

- During the constant unfolding phase, we are free to assume that $D(\vec{u})$ is replaced by the same term $T^\alpha\{\vec{u}/\vec{w}\}$ both in P_n and in Q_n. Remember that this term is obtained by respectively substituting fresh names for all the bound names of T and the names of $\vec{u}$ for those of $\vec{w}$. Thus, the process Q_n being enacted, neither $D(\vec{u})$ nor $T^\alpha\{\vec{u}/\vec{w}\}$ is subject to the action of any explicit substitution. Now, if we denote P_n^δ and Q_n^δ the terms P_n and Q_n in which $T^\alpha\{\vec{u}/\vec{w}\}$ has been substituted at the same location for $D(\vec{u})$, then under the assumption that $P_n \equiv Q_n\{v_1/r_1\}\ldots\{v_n/r_n\}$, we can state that $P_n^\delta \equiv Q_n^\delta\{v_1/r_1\}\ldots\{v_n/r_n\}$, i.e. that $\mathcal{M}.1$ holds for P_n^δ and Q_n^δ.
- As mentioned above, the same term $T^\alpha\{\vec{u}/\vec{w}\}$ is substituted for $D(\vec{u})$ both in P_n and in Q_n. Therefore, the respective locations of the occurrences of outer names of P_n^δ and Q_n^δ appearing respectively outside $T^\alpha\{\vec{u}/\vec{w}\}$ are either unchanged or shifted symmetrically. Consequently, $\mathcal{M}.2$ holds for P_n^δ and Q_n^δ.

We can deduce from what precedes that the $\mathcal{M}$-equivalence is preserved by any constant unfolding phase, i.e. $P_n^\delta \sim Q_n^\delta$. Note that as constant unfolding is not an actual transition, P_n^δ and Q_n^δ are not, strictly speaking, P-derivatives. Nevertheless, $\mathcal{D}(Q_n^\delta)$ holds since Q_n^δ and Q_n encompass the same set of explicit substitutions. Therefore, while considering the transition γ_n starting from P_n^δ and Q_n^δ, the same reasonings as in the previous cases can be applied, which leads to $P_{n+1} \sim Q_{n+1}$.

References

[1] Abadi, M., Blanchet, B., Fournet, C., 2007. Just fast keying in the pi calculus. ACM Trans. Inf. Syst. Secur. 10 (3), 9.

[2] Abadi, M., Cardelli, L., Curien, P.-L., Lvy, J.-J., 1990. Explicit substitutions. In: Conference Record of the Seventeenth Annual ACM Symposium on Principles of Programming Languages. ACM SIGACT and SIGPLAN, ACM Press, pp. 31–46.

[3] Abadi, M., Fournet, C., 2001. Mobile values, new names, and secure communication. In: POPL '01: Proceedings of the 28th ACM SIGPLAN-SIGACT symposium on Principles of programming languages. ACM, New York, NY, USA, pp. 104–115.

[4] Barendregt, H. P., 1984. The Lambda Calculus: Its Syntax and Semantics, revised Edition. No. 103 in Studies in Logic and the Foundations of Mathematics. North Holland, Amsterdam.

[5] Berry, G., Boudol, G., April 1992. The chemical abstract machine. Theoretical Computer Science 96 (1), 217–248.

[6] Boudol, G., May 1992. Asynchrony and the π-calculus (note). RR 1702, Institut National de Recherche en Informatique et en Automatique, Sofia-Antipolis, Rocquencourt.

[7] Ferrari, G. L., Montanari, U., Quaglia, P., 1994. A π-calculus with explicit substitutions: the late semantics. Lecture Notes in Computer Science 841, 342–351.

[8] Gardner, P., Wischik, L., 2000. Explicit fusions. In: Nielsen, M., Rovan, B. (Eds.), Mathematical Foundations of Computer Science , 25th International Symposium, MFCS 2000 (Bratislava, Slovakia). Vol. 1893 of LNCS. Springer, pp. 373–382.

[9] Hirschkoff, D., 1999. Handling substitutions explicitely in the π-calculus. In: Proceedings of the Second International Workshop on Explicit Substitutions.

[10] Honda, K., Tokoro, M., 1992. On asynchronous communication semantics. In: M. Tokoro, O. Nierstrasz, P. W. (Ed.), Proceedings of the Workshop on Object-Based Concurrent Computing (1991 European Conference for Object-Oriented Programming). Vol. 612 of LNCS 612. Springer-Verlag, pp. 21–51.

[11] Stehr, M.-O., Sep. 2000. CINNI – A generic calculus of explicit substitutions and its application to λ-, σ- and π-calculi. In: International Workshop on Rewriting Logic and its Applications (WRLA). Vol. 36 of Electronic Notes in Theoretical Computer Science. Elsevier Science.