# Open bisimulation, revisited

Sébastien Briais ⋆

*School of Computer and Communication Sciences, EPFL, Switzerland*

Uwe Nestmann

*Technical University of Berlin, Germany*

**Abstract**

In the context of the $\pi$-calculus, open bisimulation is prominent and popular due to its congruence properties and its easy implementability. Motivated by the attempt to generalise it to the spi-calculus, we offer a new, more refined definition and show in how far it coincides with the original one.

*Key words:* $\pi$-calculus, spi-calculus, open bisimulation

## 1 Introduction

Open bisimulation, as introduced by Sangiorgi [1] is an attractive candidate notion of bisimulation for the $\pi$-calculus [2–4] for a number of different reasons. First, it constitutes a reasonably full congruence, i.e., it is preserved by all operators including input prefix. Second, it allows for simple axiomatizations (for finite terms). Third, it is rather straightforward to build tools that symbolically check for open bisimilarity (see the MWB [5] or the ABC [6]).

The current paper arose from our attempt to "smoothly" generalise the definition of open bisimulation from the $\pi$-calculus to the spi-calculus [7], an extension of the former by cryptographic primitives used in the description of security protocols. It turns out that this is not easily doable, for reasons that we try to explain in the remainder of this Introduction. Driven by the quest for a meaningful definition of open-style bisimulation for the spi-calculus, we came up with a proposal that we then observed can also be meaningfully projected down to the case of the $\pi$-calculus and compared to the original definition.

⋆ Supported by the Swiss National Science Foundation, grant No. 21-65180.1

*Disclaimer*

Much of the material in this paper addresses the notion of substitution functions of the form $\{^{M}/_{x}\}$. In an application $P\{^{M}/_{x}\}$ of this function to some expression $P$, all (free) occurrences of the *place-holder* $x$ in $P$ are supposed to be replaced by the expression $M$. For clarity of the following explanations, let us use the terms *substitution subject* for $x$ and *substitution object* for $M$.

It is instructive to recall the *type* of substitution functions in different calculi. For instance, in the untyped $\lambda$-calculus [8], the term $P\{^{M}/_{x}\}$ may arise from the $\beta$-reduction of an application term $(\lambda x P)M$. Here, the place-holder $x$ is a *(term) variable*, while both $M$ and $P$ are $\lambda$-terms.

In contrast, in the $\pi$-calculus, the term $P\{^{M}/_{x}\}$ may arise from name-passing communication over some shared channel, e.g., by a sender $\overline{a}\langle M\rangle.Q$ and a concurrent receiver $a(x).P$. Here, both the place-holder $x$ and the object $M$ are usually just *names*. Only few presentations, as by Honda and Yoshida [9] or by Hennessy and Rathke [10] use separate syntactic categories for names and *(input) variables*; the latter would play the role of the above $x$.

Substitutions are also at the core of the many notions of bisimulation for the $\pi$-calculus [1] [3,4], ranging from *ground* over *early* and *late* to *open*. This is due to the different treatments of simulated *symbolic* input transitions, e.g., when

$$\text{simulating} \qquad P \xrightarrow{a(x)} P' \qquad \text{by} \qquad Q \xrightarrow{a(x)} Q'.$$

The problem is that after the execution of a symbolic input on channel $a$, the "input variable" name $x$ becomes free in the resulting continuation processes $P'$ and $Q'$. Considering all possible instantiations of this name $x$ by received name messages can be done either not at all (as in ground), or (as in early) before the simulating transition is chosen, or (as in late) right afterwards—or (as in open) considering all possible substitutions (not only affecting the just freed input variable) even before starting any bisimulation game. The latter case can also be seen as "very late" or "lazy" since all possible instantiations of the input variable will be checked the next time we try to continue with the bisimulation game with $P'$ and $Q'$.

In the current paper, we are not interested in the differences between the just-mentioned variations of bisimulations, but just in the *open* variant [1]. We are neither interested in the study of symbolic variants of bisimulations that were intended to provide finitary representations, amenable to computer-aided verification techniques; this has been studied in depth for the early and late

[1] Luckily, all of these notions collapse in certain sub-calculi, e.g., the asynchronous $\pi$-calculus, that are still expressive enough for most practical purposes.

versions by Hennessy and Lin [11,12] and by Boreale and De Nicola [13], and for the open variant already by Sangiorgi [1]. Finally, our interest was neither to facilitate or improve upon existing reasoning techniques, but just to pursue the goal of finding an open-style definitional scheme for the spi-calculus. The fact that we were then led to revisit open bisimulation within the $\pi$-calculus allows us to offer some potentially interesting slightly philosophical insights.

*Philosophical concerns*

What do we actually mean when we require *all possible instantiations* in a bisimulation game? More precisely: which set of substitutions shall be considered, and how do we characterize it? In other words: which entities are admissible as substitution subjects and objects, respectively? It may be of help to approach an appropriate answer from two different angles: (1) by assuming that names and variables were two distinguished syntactic categories; and (2) by assuming just the single syntactic category of names.

If variables are distinguished from names, then it naturally makes sense to have substitution subjects as a subset of the set of variables. After all, substitutions are only intended to arise from communication, even if the sender role is played by some unknown observer context. On the other hand, it naively makes sense to require that substitutions be *closing* [9]: an application of a substitution to an open term (i.e., possibly containing free occurrences of variables) shall always yield a closed term (i.e., not containing free occurrences of variables).

If, however, variables are *not* distinguished from names, at least not syntactically, then there is some interpretational freedom. We may choose an extreme point of view and consider all names that occur in a term as being potentially replaceable. This is in essence what we see in Sangiorgi's original definition of open bisimulation: all free names are treated as variables. We may however also choose a more refined point of view, based on a process's history.

*Not all substitution* subjects *shall be considered.*

By definition, only free names can ever be affected as substitution subjects. In a process, there are three kinds of free name. A free name may be free because:

(1) either it was already initially free,
(2) or it has become free after having done an input (or been substituted),
(3) or it has become free after having been created as a local name, and afterwards output to some observing process.

In contrast to Sangiorgi, we argue that names of the latter kind are constant, i.e., they should not be considered as substitution *subjects*, because they were created freshly and thus appropriately chosen. In contrast, the first two kinds shall be considered. (We formally support this point of view in Lemma 14, and show that it gives rise to an equivalent *type-aware* [2] notion of bisimulation.) Interestingly, this point of view would correspond precisely to the syntactic separation of variables and names, where names of kind (1) are interpreted as variables, thus operating on open terms.

*Not all substitution* objects *shall be considered.*

An immediately obvious and well-known restriction is that substitutions must avoid the capture of substitution objects by existing bindings.

We may also be too conservative and forbid too many substitutions: if, based on a syntactic separation of variables and names, we were to require substitutions to be closing, then the open bisimulation scheme would essentially collapse with the late bisimulation scheme since substitutions would only apply (formally, only at the next bisimulation step) to the "just freed" input variable and they would not yield further free variables in the resulting term.

Instead, we argue (partially in accordance with Sangiorgi's view) that certain instantiations should be forbidden, again depending on the history of the ongoing bisimulation game. There may be two different reasons for this.

The first reason concerns names of kind (1) or (2), say $a$, that were free in a process *before* another name, say $b$, got freshly created and extruded. Due to the freshness property, any subsequent substitution for subject $a$ must not mention $b$ as substitution object, so not to retrospectively invalidate this freshness property. In Sangiorgi's open bisimulation, represented by an indexed family of binary relations, the indexing component is precisely a structure called *distinction* that keeps track of inequalities like $a \neq b$, as required above. In analogy to type-awareness, we may use the term *freshness-awareness* to characterize bisimulations using this sort of substitutions.

The second reason concerns only names of kind (2) and resides on the intuition that substitution objects represent messages that may be sent from the observer to the observed process. In the $\pi$-calculus, there is no limitation beyond distinctions: the observer may send any name that it may have received earlier, or it may simply invent names on its own. However, it is precisely here that severe difficulties arise when moving to the spi-calculus. The main rea-

[2] Note that, here, we do not refer to the type of names in the sense of typed $\pi$-calculus, but rather to the type of the substitution function. In lack of a better word, we may also have used the term *syntax-aware*.

son there is the presence of complex messages $\mathrm{E}_{k_n}(\cdots \mathrm{E}_{k_1}(M)\cdots)$, which may dispose of some deeply nested structure involving so-called encryption keys $k_1 \ldots k_n$. Substitution objects are then all messages that the observer (potentially a malicious attacker) could possibly have generated at the moment the message was input. This generation is not arbitrary; it is constrained by the *knowledge* that the observer has acquired up to the moment of interaction.

**Example 1** *Consider the spi-calculus process*

$$P \stackrel{\text{def}}{=} (\nu k)\,(\nu m)\,\overline{a}\langle \mathrm{E}_k(m)\rangle.a(x).\overline{a}\langle k\rangle.[x{=}m]\overline{a}\langle a\rangle.\,\mathbf{0}$$

*where $(\nu k)$ denotes the generation of a fresh name, $\overline{a}\langle k\rangle$ the sending of name $k$ over channel name $a$, $a(x)$ the reception of a message over channel name $a$ with input variable $x$, $\mathrm{E}_k(m)$ the previously mentioned encryption of datum $m$ with key $k$, and $[x{=}m]$ a test of equality of names. Intuitively, the output $\overline{a}\langle a\rangle$ is impossible, because it would require that $x$ could have been substituted by $m$, which is itself impossible, because the private datum $m$ was passed on to the observer only within message $\mathrm{E}_k(m)$ encrypted with the private key $k$; however, this key $k$ was unknown to the observer when it sent the message that got received by $a(x)$ — $k$ was published only afterwards.*

*Here, a simple distinction $k \neq m$ is not sufficient to characterise disallowed substitutions because neither $m$, nor $\mathrm{E}_b(m)$, nor $\mathrm{E}_k(\mathrm{E}_b(m))$, etc., are permitted substitution objects, as the observer would have needed to also know the key $k$. In contrast, the cyphertext $\mathrm{E}_k(m)$ that the observer learnt in the first exchange could itself have been sent back to the process without further knowledge.*

Previous studies of notions of bisimulation for the spi-calculus (see an overview in [14]) resulted in careful analyses of observer (attacker) knowledge and various kinds of data structures for the representation of such knowledge.

Typically, all messages that were emitted by an observed process in the course of a bisimulation game are stored. In the above example, when the sub-term $[x{=}m]\overline{a}\langle a\rangle.\,\mathbf{0}$ appears at the top level, the observer has accumulated the messages $\mathrm{E}_k(m)$ and $k$; it has also sent a message to replace $x$.

Likewise, in particular in the proposal of symbolic bisimulation of [15], some timing or ordering information is stored that keeps track of which messages were known to the observer at the moment of the reception of a message by a process. For the above example, this could be represented by pairs $(1, \mathrm{E}_k(m))$, $(2, x)$ and $(3, k)$. Now, we may easily track that at time instant 2, the key $k$ was not yet known to decrypt the message received at time instant 1. Consequently, the name $x$ cannot have been replaced by $m$, which was encrypted using $k$. Thus, we got a technique to exclude the substitution $\{^{m}/_{x}\}$ when playing the bisimulation game on process $[x{=}m]\overline{a}\langle a\rangle.\,\mathbf{0}$. Let us use the term *knowledge-awareness* to characterize the respective bisimulation schemes.

Summing up, let us use the term *history-sensitive substitutions* to refer to substitutions that are admissible w.r.t. the above-motivated principles:

**type-awareness**
Never use once freshly created names as substitution subjects.
**freshness-awareness**
Never fuse once freshly created names with any previously known name.
**knowledge-awareness**
Never use names as substitution objects that cannot yet have been known.

Note that the original notion of open bisimulation is just freshness-aware.

*Contribution*

Recall that our goal was to find an open-style definition of bisimulation for the spi-calculus. In §2, we provide a uniformed presentation of the $\pi$- and the spi-calculus, accompanied by the original notion of open bisimulation for the $\pi$-calculus. As the above example shows, we cannot naively lift the definition of open bisimulation to the $\pi$-calculus. Instead, we proceed as follows.

First, we learn from the phenomena studied in previous bisimulations for the spi-calculus in that we transport the idea of history-sensitive substitutions to the $\pi$-calculus (see §3). We call the resulting notion K-open bisimulation. Along the way, for technical and completeness reasons, we also introduce the notion of T-open bisimulation, which is not knowledge-aware, but only type-aware. We prove all of them equivalent in precise ways.

| bisimulation | open | T-open | K-open |
|---|---|---|---|
| type-aware | − | + | + |
| freshness-aware | + | + | + |
| knowledge-aware | − | − | + |

Second, we show that due to its richer underlying information structures, we may formulate stronger congruence properties for K-open bisimilarity than for the original open bisimilarity (see §3.3); this closes a conjecture we stated in [16]. Third, after recalling *(late) hedged* bisimulations for the spi-calculus [14] (see §4), we develop the lifting of K-open bisimulation to the hedged bisimulation of the spi-calculus resulting in *open hedged* bisimulation (see §5). Forth, we prove that open hedged bisimilarity is sound w.r.t. late hedged bisimilarity. Fifth, we prove a conservative extension result: the projection of open hedged bisimulation to the $\pi$-calculus results in K-open bisimulation. Finally, §6 concludes the paper and gives a brief overview of future studies.

$$P, Q \quad ::= \quad \mathbf{0} \;\Big|\; E(x).P \;\Big|\; \overline{E}\langle F\rangle.P \;\Big|\; \phi P \;\Big|\; P \,|\, Q \;\Big|\; P + Q \;\Big|\; !\, P \;\Big|\; (\nu x)\, P$$

Table 1
Syntax of processes $\mathcal{P}$

$$\begin{array}{rcll} M, N & ::= & a & (\text{messages } \mathcal{M}) \\ E, F & ::= & a & (\text{expressions } \mathcal{E}) \\ \phi, \psi & ::= & tt \;\big|\; \phi \wedge \psi \;\big|\; [E = F] & (\text{formulae } \mathcal{F}) \end{array}$$

Table 2
Syntax of messages, expressions and formulae for the $\pi$-calculus

$$\begin{array}{rcll} M, N & ::= & a \;\big|\; \mathrm{E}_N(M) & (\text{messages } \mathcal{M}) \\ E, F & ::= & a \;\big|\; \mathrm{E}_F(E) \;\big|\; \mathrm{D}_F(E) & (\text{expressions } \mathcal{E}) \\ \phi, \psi & ::= & tt \;\big|\; \phi \wedge \psi \;\big|\; [E = F] \;\big|\; [E : \mathcal{N}] & (\text{formulae } \mathcal{F}) \end{array}$$

Table 3
Syntax of messages, expressions and formulae for the spi-calculus

## 2 Open bisimulation

### *2.1 Syntax of the $\pi$-calculus and the spi-calculus*

A countably infinite set $a, b, c, \ldots, k, l, m, n, \ldots, x, y, z, \ldots$ of *names* $\mathcal{N}$ is presupposed. In the following, we write $\tilde{z}$ for a (possibly empty) finite sequence of names $z_1, z_2, \ldots, z_n$. If $\tilde{z}$ is such a sequence, then we write $\{\tilde{z}\}$ for the set of names appearing in the sequence $\tilde{z}$. In order to unify the presentation of the $\pi$-calculus and the spi-calculus, we have parametrised the syntax of *processes* Table 1 by *messages*, *expressions* and *formulae*. Table 2 read in conjunction with Table 1 gives the syntax of the $\pi$-calculus, whereas for the spi-calculus, Table 3 and Table 1 should be considered.

The main difference between the $\pi$-calculus and the spi-calculus is that it is possible in the latter to send and receive compound *messages*; in particular, a *cyphertext* of the form $\mathrm{E}_N(M)$ denotes the message $M$ encrypted with the shared key $N$ (which might itself be a compound message). The language of *expressions* permits to manipulate compound messages; in particular, one may decrypt a cyphertext with the construction $\mathrm{D}_F(E)$ which succeeds if the expression $F$ evaluates to the key that was used to encrypt the message represented by $E$ (perfect cryptography). Finally, communication can only occur on channels (names); the guard $[E : \mathcal{N}]$ reflects this point of view by allowing syntactically to check that the expression $E$ evaluates to a bare name.

The set of names appearing in a message $M$ is written $\mathrm{n}(M)$. In the case of the $\pi$-calculus, it is simply the singleton set containing $M$ (since $M$ is a name). Similarly, the set of the names appearing in an expression $E$ is written $\mathrm{n}(E)$

| Definition of $[\![\cdot]\!] : \mathcal{E} \to \mathcal{M} \cup \{\bot\}$ | | |
|---|---|---|
| $[\![a]\!]$ | $\stackrel{\text{def}}{=} a$ | |
| $[\![\mathrm{E}_F(E)]\!]$ | $\stackrel{\text{def}}{=} \mathrm{E}_N(M)$ | if $[\![E]\!] = M \in \mathcal{M}$ and $[\![F]\!] = N \in \mathcal{M}$ |
| $[\![\mathrm{D}_F(E)]\!]$ | $\stackrel{\text{def}}{=} M$ | if $[\![E]\!] = \mathrm{E}_N(M) \in \mathcal{M}$ and $[\![F]\!] = N \in \mathcal{M}$ |
| $[\![E]\!]$ | $\stackrel{\text{def}}{=} \bot$ | in all other cases |
| Definition of $[\![\cdot]\!] : \mathcal{F} \to \{\mathbf{true}, \mathbf{false}\}$ | | |
| $[\![tt]\!]$ | $\stackrel{\text{def}}{=} \mathbf{true}$ | |
| $[\![\phi \wedge \psi]\!]$ | $\stackrel{\text{def}}{=} [\![\phi]\!]$ and $[\![\psi]\!]$ | |
| $[\![[E{=}F]]\!]$ | $\stackrel{\text{def}}{=} \mathbf{true}$ | if $[\![E]\!] = [\![F]\!] = M \in \mathcal{M}$ |
| $[\![[E{:}\mathcal{N}]]\!]$ | $\stackrel{\text{def}}{=} \mathbf{true}$ | if $[\![E]\!] = a \in \mathcal{N}$ |
| $[\![\phi]\!]$ | $\stackrel{\text{def}}{=} \mathbf{false}$ | in all other cases |
| Definition of $\mathfrak{c}(\cdot) : \mathcal{F} \to 2^{\mathcal{M} \cup \{\bot\}}$ | | |
| $\mathfrak{c}(tt)$ | $\stackrel{\text{def}}{=} \emptyset$ | |
| $\mathfrak{c}(\phi \wedge \psi)$ | $\stackrel{\text{def}}{=} \mathfrak{c}(\phi) \cup \mathfrak{c}(\psi)$ | |
| $\mathfrak{c}([E{=}F])$ | $\stackrel{\text{def}}{=} \emptyset$ | |
| $\mathfrak{c}([E{:}\mathcal{N}])$ | $\stackrel{\text{def}}{=} \{[\![E]\!]\}$ | |

Table 4
Evaluation of expressions and formulae

and the set of the names appearing in a formula $\phi$ is written $\mathrm{n}(\phi)$. Finally, the set of free names $\mathrm{fn}(P)$ and bound names $\mathrm{bn}(P)$ of a process $P$ are defined as usual taking into account that the name $x$ is bound in $P$ by the constructs $E(x).P$ and $(\nu x)\, P$. These notions are straightforwardly lifted to sets. Finally, we use $=_\alpha$ to relate any two processes that only differ w.r.t. the clash-free renaming of their bound names.

### *2.2 Labelled (late) semantics*

Table 4 defines the straightforward evaluation of expressions and formulae, as well as some name constraints of a given formula. Table 5 defines a labelled transition $P \stackrel{\mu}{\to}_S P'$ where $\mu$ is an action and $S$ is a set of names. The set $S$ collects those free names that are required—either in their role as communication channel or within some guard formula of the spi-calculus—to enable the transition. In the $\pi$-calculus, where only names are considered, it can be simply ignored but it will be used later on for the case of spi-calculus.

Upon this transition system, the late semantics of the $\pi$-calculus and the spi-

$$\textsc{Input}\ \frac{[\![E]\!] = a \in \mathcal{N}}{E(x).P \xrightarrow{a(x)}_{\{a\}} P} \qquad \textsc{Output}\ \frac{[\![E]\!] = a \in \mathcal{N} \qquad [\![F]\!] = M \in \mathcal{M}}{\overline{E}\langle F\rangle.P \xrightarrow{\overline{a}\,M}_{\{a\}} P}$$

$$\textsc{Close-l}\ \frac{P \xrightarrow{a(x)}_S P' \qquad Q \xrightarrow{(\nu\tilde{z})\,\overline{a}\,M}_{S'} Q'}{P \mid Q \xrightarrow{\tau}_{S\cup S'} (\nu\tilde{z})\,(P'\{^{M}/_{x}\} \mid Q')}\ \{\tilde{z}\} \cap \mathrm{fn}(P) = \emptyset$$

$$\textsc{Open}\ \frac{P \xrightarrow{(\nu\tilde{z})\,\overline{a}\,M}_S P'}{(\nu z')\,P \xrightarrow{(\nu z'\tilde{z})\,\overline{a}\,M}_{S\setminus\{z'\}} P'}\ z' \in \mathrm{n}(M) \setminus \{a, \tilde{z}\}$$

$$\textsc{Res}\ \frac{P \xrightarrow{\mu}_S P'}{(\nu z)\,P \xrightarrow{\mu}_{S\setminus\{z\}} (\nu z)\,P'}\ z \notin \mathrm{n}(\mu) \qquad \textsc{Guard}\ \frac{P \xrightarrow{\mu}_S P'}{\phi P \xrightarrow{\mu}_{S\cup \mathfrak{c}(\phi)} P'}\ [\![\phi]\!] = \mathbf{true}$$

$$\textsc{Par-l}\ \frac{P \xrightarrow{\mu}_S P'}{P \mid Q \xrightarrow{\mu}_S P' \mid Q}\ \mathrm{bn}(\mu) \cap \mathrm{fn}(Q) = \emptyset \qquad \textsc{Sum-l}\ \frac{P \xrightarrow{\mu}_S P'}{P + Q \xrightarrow{\mu}_S P'}$$

$$\textsc{Rep}\ \frac{P \mid !\,P \xrightarrow{\mu}_S P'}{!\,P \xrightarrow{\mu}_S P'} \qquad \textsc{Alpha}\ \frac{P =_\alpha P' \qquad P' \xrightarrow{\mu}_S P''}{P \xrightarrow{\mu}_S P''}$$

Table 5
The late semantics of the $\pi$-calculus

calculus is given by: $P \xrightarrow{\mu} P'$ if and only if there is $S$ such that $P \xrightarrow{\mu}_S P'$.

The syntax of actions $\mu$ is given by:

$$\mu \quad ::= \quad \tau \;\Big|\; a(x) \;\Big|\; (\nu\tilde{z})\,\overline{a}\,M \quad \text{(actions)}$$

The bound output actions $(\nu\tilde{z})\,\overline{a}\,M$ are such that $\{\tilde{z}\} \subseteq \mathrm{n}(M)$. In the case of the $\pi$-calculus, since messages $M$ are reduced to names, we have two cases: either $\tilde{z}$ is the empty sequence and $(\nu\tilde{z})\,\overline{a}\,M$ is simply written $\overline{a}\,M$ or $\tilde{z} = M$ and the bound output action is simply $(\nu z)\,\overline{a}\,z$ where $z = M$.

The set of names $\mathrm{n}(\mu)$ is defined by:

$$\mathrm{n}(\tau) := \emptyset, \qquad \mathrm{n}(a(x)) := \{a, x\}, \qquad \mathrm{n}((\nu\tilde{z})\,\overline{a}\,M) := \{a, \tilde{z}\} \cup \mathrm{n}(M).$$

The set of bound names $\mathrm{bn}(\mu)$ of $\mu$ is defined by:

$$\mathrm{bn}(\tau) := \emptyset, \qquad \mathrm{bn}(a(x)) := \{x\}, \qquad \mathrm{bn}((\nu\tilde{z})\,\overline{a}\,M) := \{\tilde{z}\}.$$

Moreover, if $\mu = a(x)$ or $\mu = (\nu\tilde{z})\,\overline{a}\,M$, we define $\mathrm{ch}(\mu) \stackrel{\mathrm{def}}{=} a$.

As mentioned in the Introduction, open bisimulation was introduced by Sangiorgi [1]. It relies on the notion of distinction to keep track of inequalities of names in order to constrain the set of substitutions to be considered in the respective bisimulation game.

**Definition 2 (distinction)** *A binary relation $D \subseteq \mathcal{N} \times \mathcal{N}$ on names is called* distinction *if it is finite, symmetric, and irreflexive.*

*By* $\mathrm{n}(D)$ *we denote the set of names contained in* $D$*.*

$X^=$ *denotes the symmetric closure of a binary relation* $X$*.*

*If* $A$*,* $B$ *are two sets of names, we define the distinction* $A \otimes B$ *to be* $\{\, (x,y) \in A \times B \cup B \times A \mid x \neq y \,\}$*. Like* $\times$*, we let* $\otimes$ *have higher precedence than* $\cup$ *or other standard set operators.* $A^{\neq}$ *abbreviates* $A \otimes A$*.*

**Definition 3 (substitution)** *A substitution* $\sigma$ *is a total function* $\mathcal{N} \to \mathcal{M}$ *such that its* support $\mathrm{supp}(\sigma) := \{x \mid x\sigma \neq x\}$ *is a finite set.*

*The* co-support *of* $\sigma$ *is* $\mathrm{cosupp}(\sigma) := \{x\sigma \mid x \in \mathrm{supp}(\sigma)\}$*.*

*The set of* names *of* $\sigma$ *is* $\mathrm{n}(\sigma) := \mathrm{supp}(\sigma) \cup \mathrm{n}(\mathrm{cosupp}(\sigma))$*.*

As said previously, distinctions are to prevent substitutions to fuse two names that were assumed to be different at some point. Hence the definition of so-called respectful substitutions.

**Definition 4 (respectfulness)** *Let* $D$ *be a distinction,* $\sigma$ *a substitution.*

$\sigma$ respects $D$*, written* $\sigma \triangleright D$*, if and only if* $x\sigma \neq y\sigma$ *for all* $(x,y) \in D$*.*

*If* $\sigma$ *respects* $D$*, then* $D\sigma$ *is defined as* $\{(x\sigma, y\sigma) \mid (x,y) \in D\}$*.*

Note that since $\mathcal{M} = \mathcal{N}$ in the case of the $\pi$-calculus, $D\sigma$ is itself a distinction.

An open bisimulation is a distinction-indexed family of symmetric relations between processes that satisfies some condition.

**Definition 5 (open bisimulation)** *The family* $(\mathcal{R}_D)_{D \in \mathcal{D}}$ *(where* $\mathcal{D}$ *is a set of distinctions) of symmetric relations is an open bisimulation if for all* $D \in \mathcal{D}$*, for all substitutions* $\sigma$ *such that* $\sigma \triangleright D$*, for all* $(P,Q) \in \mathcal{R}_D$*, whenever* $P\sigma \xrightarrow{\mu} P'$ *(with* $\mathrm{bn}(\mu)$ *fresh), there exists* $Q'$ *such that* $Q\sigma \xrightarrow{\mu} Q'$ *and*

- *if* $\mu = (\nu z)\,\overline{a}\,z$ *for some* $a$ *and* $z$*,* $D' \in \mathcal{D}$ *and* $(P', Q') \in \mathcal{R}_{D'}$

*where* $D' = D\sigma \cup \{z\} \otimes (\mathrm{fn}((P+Q)\sigma) \cup \mathrm{n}(D\sigma))$
- *otherwise,* $D\sigma \in \mathcal{D}$ *and* $(P', Q') \in \mathcal{R}_{D\sigma}$.

The induced equivalence is defined as usual, modulo the indexing component.

**Definition 6 (open bisimilarity)** *Let* $P, Q \in \mathcal{P}$ *and* $D$ *a distinction. We say that* $P$ *and* $Q$ *are* open $D$-bisimilar*—written* $P \sim_{\mathrm{O}}^{D} Q$*—if there exists an open bisimulation* $(\mathcal{R}_{\underline{D}})_{\underline{D} \in \mathcal{D}}$ *such that* $D \in \mathcal{D}$ *and* $(P, Q) \in \mathcal{R}_D$*; the relation* $\sim_{\mathrm{O}}^{D}$ *itself is then called* open $D$-bisimilarity.

Instead of families of binary relations between processes we may also use ternary relations, which is often done in the context of the spi-calculus. Thus, instead of $(P, Q) \in \mathcal{R}_D$, we then write $(D, P, Q) \in \mathcal{R}$, where $D$ is usually called *environment*, and the ternary relation is called *environment-sensitive*. It is mainly for easier readability that we adopt the ternary style in the following, although a bit of care needs to be taken to lift the three equivalence properties to the ternary format. For example, for non-symmetric environment structures $\mathsf{e}$, i.e., where $\mathsf{e} \neq \mathsf{e}^{-1}$, a (ternary) environment-sensitive relation is called *symmetric* if and only if $(\mathsf{e}, P, Q) \in \mathcal{R} \Leftrightarrow (\mathsf{e}^{-1}, Q, P) \in \mathcal{R}$.

As mentioned in the Introduction, open bisimulation enjoys powerful congruence properties. More precisely, Sangiorgi [1] showed that open $D$-bisimilarity is a so-called *$D$-congruence*: open $D$-bisimilarity is preserved by *$D$-respectful* contexts, i.e., contexts in which the occurrence of the hole is not underneath an input prefix binding a name in $D$. Actually, [1] stated an even more precise result:

**Proposition 7** *Let* $P$*,* $Q$ *two processes and* $D$ *a distinction. We assume that* $P \sim_{\mathrm{O}}^{D} Q$*. Then,*

*(1)* $\forall R : R \mid P \sim_{\mathrm{O}}^{D} R \mid Q$ *and* $P \mid R \sim_{\mathrm{O}}^{D} Q \mid R$
*(2)* $\forall R : R + P \sim_{\mathrm{O}}^{D} R + Q$ *and* $P + R \sim_{\mathrm{O}}^{D} Q + R$
*(3)* $!\,P \sim_{\mathrm{O}}^{D} !\,Q$
*(4)* $\forall \phi : \phi P \sim_{\mathrm{O}}^{D} \phi Q$
*(5)* $\forall a, z : \overline{a}\langle z \rangle.P \sim_{\mathrm{O}}^{D} \overline{a}\langle z \rangle.Q$
*(6)* $\forall x : (\nu x)\,P \sim_{\mathrm{O}}^{D \setminus x} (\nu x)\,Q$
*(7)* $\forall a, x : x \notin D \Rightarrow a(x).P \sim_{\mathrm{O}}^{D} a(x).Q$

## 3 Open bisimulation, reloaded

Before proceeding to our new proposal to define open-style bisimulation, we provide a slightly different, but equivalent variant of the previously given standard notion. This variant will make it easier to relate to our new proposal.

In this section, we define the notion of T-open bisimulation. The simple idea is, as we mentioned already in the Introduction, to prevent names that were previously (in the course of a bisimulation game) created freshly from being considered as permissible substitution subjects.

The knowledgeable reader may be reminded of the notion of *quasi-open* bisimulation, proposed by Sangiorgi and Walker [17], and later on revisited by Fu [18]. There, the use of distinctions as environments was adapted to the use of a simple set of names that were once freshly created and therefore deemed to remain constant. The resulting quasi-open bisimulation was recognised as being strictly weaker than open bisimulation. Sangiorgi and Walker intuitively summarised this difference as: "*In open bisimilarity, when a name $z$ is sent in a bound-output action, the distinction is enlarged to ensure that $z$ is never identified with any name that is free in the processes that send it. In quasi-open bisimilarity, in contrast, at no point after the scope of $z$ is extruded can a substitution be applied that identifies $z$ with any other name.*" [17].

Like quasi-open bisimulation, the following definition also explicitly keeps track of previously freshly created names. However, it does not use this information to prevent the fusion of such fresh names like quasi-open bisimulation does. It only uses this information to implement the idea that fresh names can be considered as constant names once chosen, such that they should afterwards never be used as substitution subjects. In fact, Lemmas 14 and 15 show that this change still faithfully retains the equational power of open bisimulation.

**Definition 8 (T-environment)** *The pair $(D, C)$ where $D$ is a distinction and $C$ is a finite subset of names is a* T-environment *if $C^{\neq} \subseteq D$. The set of all* T*-environments is written $\mathcal{F}$.*

The distinction $D$ plays the same role as in open bisimulation, while the set $C$ indicates which names can be considered as constant names. It is used to refine the notion of respectfulness, as follows.

**Definition 9 (respectful substitution)**
*Let $(D, C)$ be a* T*-environment and $\sigma$ a substitution. We say that $\sigma$ respects $(D, C)$ – written $\sigma \blacktriangleright (D, C)$ – if $\sigma \triangleright D$ and* $\mathrm{supp}(\sigma) \cap C = \emptyset$.

The following lemma states the link between the two previously seen notion of respectfulness.

**Lemma 10** *Let $D$ be a distinction and $\sigma$ a substitution with $\sigma \triangleright D$. Let $C$ be a finite set of names such that $C^{\neq} \subseteq D$. Then there exists a substitution $\sigma'$ and a bijective substitution $\theta$ such that $\sigma' \blacktriangleright (D, C)$ and $\sigma = \sigma'\theta$ and* $\mathrm{n}(\theta) \subseteq C \cup C\sigma$.

**Proof.**

We first prove that $\sigma$ is injective on the finite set $C$.

Indeed, let $x, y \in C$ such that $x \neq y$. Since $C^{\neq} \subseteq D$, we have $(x, y) \in D$. Moreover, we have $\sigma \triangleright D$, so we have $x\sigma \neq y\sigma$. This proves that $\sigma$ is injective on $C$.

According to Lemma 1.4.11 of [4], there exists a bijective substitution $\theta$ such that $\sigma$ and $\theta$ agree on $C$. By construction, we also have that $\mathrm{n}(\theta) \subseteq C \cup C\sigma$.

Let $\sigma' = \sigma\theta^{-1}$. Then $\sigma'$ is a substitution such that $\sigma = \sigma'\theta$.

It remains now to prove that $\sigma' \blacktriangleright (D, C)$.

We first show that $\sigma' \triangleright D$. Let $x, y \in D$. Since $\sigma \triangleright D$, we have that $x\sigma \neq y\sigma$. Now, since $\theta^{-1}$ is bijective, we get $x\sigma\theta^{-1} \neq y\sigma\theta^{-1}$, hence $x\sigma' \neq y\sigma'$ and $\sigma' \triangleright D$.

Now, we show that $\mathrm{supp}(\sigma') \cap C = \emptyset$. Let $x \in C$. Since $\sigma$ and $\theta$ agree on $C$, we have $x\sigma = x\theta$. So $x\sigma' = x\sigma\theta^{-1} = x\theta\theta^{-1} = x$ and $x \notin \mathrm{supp}(\sigma')$. Hence $\mathrm{supp}(\sigma') \cap C = \emptyset$.

Finally, we have proven that $\sigma' \blacktriangleright (D, C)$. ■

**Definition 11 (T-relation)** *A* T-relation $\mathcal{R}$ *is a subset of* $\mathcal{F} \times \mathcal{P} \times \mathcal{P}$.

**Definition 12 (T-open bisimulation)** *A symmetric* T*-relation* $\mathcal{R}$ *is a* T-open bisimulation*, if for all* $((D, C), P, Q) \in \mathcal{R}$ *and for all substitutions* $\sigma$ *such that* $\sigma \blacktriangleright (D, C)$*, whenever* $P\sigma \xrightarrow{\mu} P'$ *(with* $\mathrm{bn}(\mu)$ *fresh), there exists* $Q'$ *such that* $Q\sigma \xrightarrow{\mu} Q'$ *and*

- *if* $\mu = (\nu z)\,\overline{a}\,z$ *for some* $a$ *and* $z$, $((D', C \cup \{z\}), P', Q') \in \mathcal{R}$ *where* $D' = D\sigma \cup \{z\} \otimes (\mathrm{fn}((P{+}Q)\sigma) \cup \mathrm{n}(D\sigma))$
- *otherwise,* $((D\sigma, C), P', Q') \in \mathcal{R}$

The only two differences compared to open bisimulation are, first, that the notion of respectfulness is slightly modified such that it takes into account the constant names of a T-environment and, second, that the extruded names are being accumulated in the pool of constant names of T-environments.

**Definition 13 (T-open bisimilarity)** *Let* $P, Q \in \mathcal{P}$ *and* $(D, C) \in \mathcal{F}$.

$P$ *and* $Q$ *are* T*-open* $(D, C)$*-bisimilar, written* $P \sim_{\mathrm{T}}^{(D,C)} Q$*, if there is a* T*-open bisimulation* $\mathcal{R}$ *such that* $((D, C), P, Q) \in \mathcal{R}$.

Open and T-open bisimilarity are equivalent in the following sense, as expressed by the combination of the statements of the Lemmas 14 and 15.

**Lemma 14** *Let $P, Q \in \mathcal{P}$ and $(D, C) \in \mathcal{F}$.*

*If $P \sim_{\mathrm{T}}^{(D,C)} Q$, then $P \sim_{\mathrm{O}}^{D} Q$.*

**Proof.**

Let $\mathcal{R}$ be a T-open bisimulation such that $((D, C), P, Q) \in \mathcal{R}$.

Let $\mathcal{D} = \{D \mid \exists C, P, Q : ((D, C), P, Q) \in \mathcal{R}\}$

For $D \in \mathcal{D}$ and $\theta$ a bijective substitution, let

$$\mathcal{R}'_{D\theta} = \{\ (P\theta, Q\theta) \mid \exists C : ((D, C), P, Q) \in \mathcal{R}\ \}$$

Let $\mathcal{D}' = \{D\theta \mid D \in \mathcal{D} \wedge \theta \text{ bijective substitution}\}$.

We have that $(\mathcal{R}'_D)_{D\in\mathcal{D}'}$ is an open bisimulation.

Indeed, let $D' \in \mathcal{D}'$, $\sigma$ a substitution with $\sigma \triangleright D'$ and $(P_0, Q_0) \in \mathcal{R}'_{D'}$. By definition, there is $D \in \mathcal{D}$ and $\theta$ a bijective substitution such that $D' = D\theta$. Moreover, there exists $C$ with $((D, C), P, Q) \in \mathcal{R}$ and $P_0 = P\theta$ and $Q_0 = Q\theta$.

Since $\sigma \triangleright D\theta$, we have $\theta\sigma \triangleright D$. We then use Lemma 10 with $\theta\sigma$ and $C$. We have the existence of a substitution $\sigma'$ and a bijective substitution $\theta'$ such that $\theta\sigma = \sigma'\theta'$, $\sigma' \blacktriangleright (D, C)$ and $\mathrm{n}(\theta') \subseteq C \cup C\theta$.

Assume now that $P_0\sigma \xrightarrow{\mu} P'_0$ (with $\mathrm{bn}(\mu)$ fresh), i.e. $P\theta\sigma \xrightarrow{\mu} P'_0$, i.e. $P\sigma'\theta' \xrightarrow{\mu} P'_0$. Since $\theta'$ is bijective, we have $P\sigma' \xrightarrow{\mu\theta'^{-1}} P'_0\theta'^{-1}$.

Since $((D, C), P, Q) \in \mathcal{R}$ and $\sigma' \blacktriangleright (D, C)$, by definition, there exists $Q'$ such that $Q\sigma' \xrightarrow{\mu\theta'^{-1}} Q'$ and

- if $\mu\theta'^{-1} = (\nu z)\,\overline{a}\,z$ then $((D'', C \cup \{z\}), P'_0\theta'^{-1}, Q') \in \mathcal{R}$ where $D'' = D\sigma' \cup \{z\} \otimes (\mathrm{fn}((P + Q)\sigma') \cup \mathrm{n}(D\sigma'))$
- otherwise $((D\sigma', C), P'_0\theta'^{-1}, Q') \in \mathcal{R}$

Let $Q'_0 = Q'\theta'$, then we have $Q' = Q'_0\theta'^{-1}$ and $Q\sigma' \xrightarrow{\mu\theta'^{-1}} Q'_0\theta'^{-1}$.

Since $\theta'^{-1}$ is bijective, we get $Q\sigma'\theta' \xrightarrow{\mu} Q'_0$, i.e., $Q\theta\sigma \xrightarrow{\mu} Q'_0$, i.e. $Q_0\sigma \xrightarrow{\mu} Q'_0$.

- if $\mu = (\nu z)\,\overline{a}\,z$, then $\mu\theta'^{-1} = (\nu z)\,\overline{a}\,z$ and we have by assumption $((D'', C \cup \{z\}), P'_0\theta'^{-1}, Q'_0\theta'^{-1}) \in \mathcal{R}$ where $D'' = D\sigma' \cup \{z\} \otimes (\mathrm{fn}((P+Q)\sigma') \cup \mathrm{n}(D\sigma'))$. So, by definition, we have $(P'_0, Q'_0) \in \mathcal{R}'_{D''\theta'}$. But $D''\theta' = D\sigma'\theta' \cup \{z\theta'\} \otimes (\mathrm{fn}((P + Q)\sigma')\theta' \cup \mathrm{n}(D\sigma'\theta'))$. So $D''\theta' = D\theta\sigma \cup \{z\theta\} \otimes (\mathrm{fn}((P + Q)\theta\sigma) \cup \mathrm{n}(D\theta\sigma))$, i.e. $D''\theta' = D'\sigma \cup \{z\} \otimes (\mathrm{fn}((P_0 + Q_0)\sigma) \cup \mathrm{n}(D'\sigma))$ (because $z$ is fresh and thus $z \notin \mathrm{n}(\theta')$).

- otherwise $((D\sigma', C), P'_0\theta'^{-1}, Q'_0\theta'^{-1}) \in \mathcal{R}$ so $(P'_0, Q'_0) \in \mathcal{R}'_{D\sigma'\theta'}$ and $D\sigma'\theta' = D\theta\sigma = D'\sigma$.

Hence, $(\mathcal{R}'_D)_{D\in\mathcal{D}'}$ is an open bisimulation. ■

**Lemma 15** *Let $P, Q \in \mathcal{P}$ and $D$ a distinction.*

*If $P \sim_{\mathrm{O}}^{D} Q$, then $\forall C : C^{\neq} \subseteq D \Rightarrow P \sim_{\mathrm{T}}^{(D,C)} Q$.*

**Proof.**

This result is obvious because $\sigma \blacktriangleright (D, C)$ implies $\sigma \triangleright D$. ■

### *3.2 A knowledge-aware variant of open bisimulation*

As motivated in the Introduction, we propose a bisimulation that makes explicit an attacker who plays against the two players $P$ and $Q$ involved in the bisimulation game. The knowledge of the attacker is stored in K-environments of the form $(O, V, \prec)$. The set of names $V$ represents all the substitutable free names (those that were initially free or have become free after an input action). The set of messages $O$ contains all the messages that were emitted by $P$ and $Q$, except the names of $V$. Finally, the relation $\prec$ indicates for each substitutable name $x$ the available knowledge $\{\, n \in O \mid n \prec x \,\}$ that had possibly been acquired by the attacker at the moment the name $x$ was input. Thus, the relation $\prec$ constrains the messages that may possibly be or have been received at a particular moment from the attacker.

**Definition 16 (K-environment)** *A* K*-environment is a triple $(O, V, \prec)$ such that $O \cup V$ is a finite subset of $\mathcal{N}$, $O \cap V = \emptyset$ and $\prec \subseteq O \times V$. The set of all* K*-environments is $\mathcal{K}$.*

If $\mathsf{pe}$ is a K-environment, and $n \in \mathcal{N}$, it is possible to extend $\mathsf{pe}$ with $n$ in two ways. Either $n$ is meant to be an emitted name and it is added to the constant part of $\mathsf{pe}$, or $n$ is meant to be a received name and it is added to the variable part of $\mathsf{pe}$ and put in relation with all already emitted names. If $n$ is already contained in $\mathsf{pe}$, its addition to $\mathsf{pe}$ has no effect.

**Definition 17 (Extension of a K-environment)** *Let $\mathsf{pe} = (O, V, \prec)$ be a* K*-environment and $n \in \mathcal{N}$. We define*

*(1)* $\mathsf{pe} \oplus_{\mathrm{O}} n \stackrel{\text{def}}{=} (O', V, \prec)$ *where* $O' \stackrel{\text{def}}{=} O \cup \{n\}$ *if* $n \notin V$ *and* $O' \stackrel{\text{def}}{=} O$ *otherwise.*
*(2) if* $n \notin O \cup V$, $\mathsf{pe} \oplus_{\mathrm{V}} n \stackrel{\text{def}}{=} (O, V \cup \{n\}, \prec')$ *where* $\prec' \stackrel{\text{def}}{=} \prec \cup\, O \times \{n\}$.

Keeping in mind that a substitution represents the potential inputs the attacker could have generated, we define the set of respectful substitutions. A substitution $\sigma$ respects a K-environment $\mathsf{pe} = (O, V, \prec)$ if it affects only substitutable names (those in $V$) and if for each $x \in V$, it takes only values that were generatable at the moment when $x$ was input. This means that such a name $x$ can use any name in $V$ (this corresponds to fusing two substitutable names), or use any name in $O$ that was known by the attacker when $x$ was input (this is indicated by the relation $\prec$) or use any new fresh name not contained in $\mathsf{pe}$ (this corresponds to the creation of free names by the attacker). In the $\pi$-calculus, since a substitution replaces a name by a name, this can be easily and concisely expressed by:

**Definition 18 (respectful substitution)**
*A substitution $\sigma$ respects a* K*-environment $\mathsf{pe} = (O, V, \prec)$, written $\sigma \blacktriangleright\!\blacktriangleright \mathsf{pe}$, if:*

*(1)* $\mathrm{supp}(\sigma) \subseteq V$
*(2)* $\forall x \in V : x\sigma \in O \Rightarrow x\sigma \prec x$

Roughly speaking, in spi-calculus, $x\sigma$ is built using names from $V$, the messages from $O$ that are permitted by $\prec$ and some freshly generated names. In $\pi$-calculus, this is simplified to $x\sigma \prec x$ because $x\sigma \in \mathcal{N}$.

Any K-environment $\mathsf{pe} = (O, V, \prec)$ may, under the impact of some respectful substitution $\sigma$, be straightforwardly updated to $\mathsf{pe}^\sigma$. In general, the knowledge contained in $O$ should be updated to $O\sigma$. However, in the $\pi$-calculus, substitution deals only with names, and since $O \cap V = \emptyset$ and $\mathrm{supp}(\sigma) \subseteq V$ we have $O\sigma = O$. The set $V$ of substitutable names should keep all the names that were not affected by $\sigma$, and in addition list all the new names that were created by the attacker, as visible in the substitution objects. [3] Particular care must be taken when computing the new relation $\prec'$ because of the possibility that $\sigma$ fuses two names of $V$. Fusing two names $x$ and $y$ (by $x\sigma = y\sigma$) corresponds to a voluntary loss of power of the attacker: the only admissible values for the fused name are those that were admissible for *both $x$ and $y$*.

**Definition 19 (K-environment updating)**
*Let $\mathsf{pe} = (O, V, \prec)$ be a* K*-environment and $\sigma$ a substitution such that $\sigma \blacktriangleright\!\blacktriangleright \mathsf{pe}$. The updated environment is $\mathsf{pe}^\sigma \stackrel{\text{def}}{=} (O, V', \prec')$ of $\mathsf{pe}$ by $\sigma$ where*

$$V' \stackrel{\text{def}}{=} (V \setminus \mathrm{supp}(\sigma)) \cup \{\, x\sigma \mid x \in \mathrm{supp}(\sigma) \wedge x\sigma \notin O \,\}$$
$$\prec' \stackrel{\text{def}}{=} \{\, (n, x') \mid \forall x \in V : x' \in \mathrm{n}(x\sigma) \Rightarrow n \prec x \,\}$$

[3] The fact that we put the names created by the environment in the substitutable part gives a "lazy" flavour to our definition, because it allows the attacker to uncover itself gradually.

**Definition 20 (K-relation)** *A* K*-relation* $\mathcal{R}$ *is a subset of* $\mathcal{K} \times \mathcal{P} \times \mathcal{P}$ *such that* $\forall((O, V, \prec), P, Q) \in \mathcal{R} : \mathrm{fn}(P{+}Q) \subseteq O \cup V$.

The new variant of open bisimulation now simply keeps track of whether dynamically freed names are substitutable or not. If they are, then we explicitly state that previously created names may be used in future substitutions. Names that will be created later on—by the process—will not be permitted.

**Definition 21 (K-open bisimulation)** *A symmetric* K*-relation* $\mathcal{R}$ *is a* K-open bisimulation*, if for all* $(\mathsf{pe}, P, Q) \in \mathcal{R}$ *and for all substitutions* $\sigma$ *such that* $\sigma \blacktriangleright\!\blacktriangleright \mathsf{pe}$*, whenever* $P\sigma \xrightarrow{\mu} P'$ *(with* $\mathrm{bn}(\mu)$ *fresh), there exists* $Q'$ *such that* $Q\sigma \xrightarrow{\mu} Q'$ *and*

- *if* $\mu = \tau$*, then* $(\mathsf{pe}^\sigma, P', Q') \in \mathcal{R}$
- *if* $\mu = a(x)$ *then* $(\mathsf{pe}^\sigma \oplus_{\mathrm{V}} x, P', Q') \in \mathcal{R}$
- *if* $\mu = (\nu z)\,\overline{a}\,z$ *or* $\mu = \overline{a}\,z$ *then* $(\mathsf{pe}^\sigma \oplus_{\mathrm{O}} z, P', Q') \in \mathcal{R}$

We see in this definition that indeed $O$ collects all the messages emitted by $P$ and $Q$ (but the addition $\mathsf{pe}^\sigma \oplus_{\mathrm{O}} z$ has only effect when $\mu = (\nu z)\,\overline{a}\,z$ because $\mathsf{pe}$ contains all free names of $P$ and $Q$) and $V$ collects all substitutable names.

**Definition 22 (K-open bisimilarity)** *Let* $P, Q \in \mathcal{P}$ *and* $E \in \mathcal{K}$.

$P$ *and* $Q$ *are* K*-open* $\mathsf{pe}$*-bisimilar, written* $P \sim^{\mathsf{pe}}_{\mathrm{K}} Q$*, if there is a* K*-open bisimulation* $\mathcal{R}$ *such that* $(E, P, Q) \in \mathcal{R}$.

In the $\pi$-calculus, it is possible to represent any K-environment by some T-environment. The idea is that all names in $O$ should be kept pairwise distinct (they were fresh names) and for all $(n, x) \in O \times V$, if $n$ cannot be used to generate $x$ (i.e. $\neg n \prec x$), then $n$ and $x$ should be distinct ($n \neq x$).

**Definition 23 (T-environment of a K-environment)** *Let* $\mathsf{pe} = (O, V, \prec)$ *be a* K*-environment.*
*Then, we define* $\mathrm{f}(\mathsf{pe}) \stackrel{\mathrm{def}}{=} (D, O)$ *where* $D \stackrel{\mathrm{def}}{=} O^{\neq} \cup \bigcup_{n \in O \wedge x \in V \wedge \neg n \prec x} \{(n, x)\}^{=}$.

Note that if $\mathsf{pe} \in \mathcal{K}$, then $\mathrm{f}(\mathsf{pe}) \in \mathcal{F}$.

The next lemma gives a precise correspondence between respectfulness of a T-environment and respectfulness of a K-environment.

**Lemma 24** *Let* $\mathsf{pe} = (O, V, \prec)$ *be a* K*-environment and* $\sigma$ *a substitution. Then*

$$\sigma \blacktriangleright\!\blacktriangleright \mathsf{pe} \quad \Longleftrightarrow \quad \mathrm{supp}(\sigma) \subseteq V \wedge \sigma \blacktriangleright \mathrm{f}(\mathsf{pe})$$

**Proof.**

Let $D$ such that $\mathrm{f}(\mathsf{pe}) = (D, O)$.

- First assume that $\sigma \blacktriangleright\!\blacktriangleright \mathsf{pe}$.
  By definition, we have $\mathrm{supp}(\sigma) \subseteq V$ and $\forall x \in V : x\sigma \in O \Rightarrow x\sigma \prec x$.
  Since $\mathrm{supp}(\sigma) \subseteq V$ and $O \cap V = \emptyset$, we have $\mathrm{supp}(\sigma) \cap O = \emptyset$.
  Let $(x, y) \in D$. We have to show that $x\sigma \neq y\sigma$. There are four cases (according to the definition of $D$): either $x, y \in O$ with $x \neq y$, or $x \in O$, $y \in V$ and $\neq x \prec y$ or the two other symmetric cases.
  By case distinction, assume that $x, y \in O$ and $x \neq y$. Since $\mathrm{supp}(\sigma) \cap O = \emptyset$, we have $x\sigma = x$, $y\sigma = y$, hence $x\sigma \neq y\sigma$.
  Now assume that $x \in O$, $y \in V$ and $\neg x \prec y$. Since $\mathrm{supp}(\sigma) \cap O = \emptyset$, we have $x\sigma = x$. Assume by contradiction that $y\sigma = x\sigma = x$, then we have $y\sigma \in O$. Thus, we have $y\sigma \prec y$ which is equivalent to $x \prec y$ and thus leading to a contradiction. So $x\sigma \neq y\sigma$.
  The two other symmetric cases are treated in the same way.
  Hence $\sigma \blacktriangleright \mathrm{f}(\mathsf{pe})$.
- Assume now that $\mathrm{supp}(\sigma) \subseteq V \wedge \sigma \blacktriangleright \mathrm{f}(\mathsf{pe})$.
  We have then that $\sigma \triangleright D$.
  By hypothesis, $\mathrm{supp}(\sigma) \subseteq V$.
  Let $x \in V$ and assume that $x\sigma \in O$. We have to show that $x\sigma \prec x$. Assume by contradiction that $\neg x\sigma \prec x$. Then, by definition of $D$, we have that $(x\sigma, x) \in D$. Since $\sigma$ respects $D$, we have $x\sigma\sigma \neq x\sigma$, but since $x\sigma \in O$ and $\mathrm{supp}(\sigma) \cap O = \emptyset$, we have $x\sigma\sigma = x\sigma$, obtaining a contradiction.
  Hence $\sigma \blacktriangleright\!\blacktriangleright \mathsf{pe}$.

■

The next lemma studies the updating of a K-environment.

**Lemma 25** *Let* $\mathsf{pe} = (O, V, \prec)$ *be a* K*-environment,* $D$ *such that* $\mathrm{f}(\mathsf{pe}) = (D, O)$ *and* $\sigma$ *a substitution such that* $\sigma \blacktriangleright\!\blacktriangleright \mathsf{pe}$*. Then* $\mathrm{f}(\mathsf{pe}^\sigma) = (D\sigma, O)$*.*

**Proof.**

Let $(D', O) = \mathrm{f}(\mathsf{pe}^\sigma)$. We have to show that $D' = D\sigma$.

By definition, $D' = O^{\neq} \cup \bigcup_{n \in O \wedge x' \in V' \wedge \neg n \prec' x'} \{(n, x'), (x', n)\}$ where $V' = (V \setminus \mathrm{supp}(\sigma)) \cup \{x\sigma \mid x \in \mathrm{supp}(\sigma) \wedge x\sigma \notin O\}$ and $\prec'$ is defined by

$$n \prec' x' \Leftrightarrow \bigwedge_{x \in V \wedge x' \in \mathrm{n}(x\sigma)} n \prec x$$

Let $(x', y') \in D'$. If $(x', y') \in O \otimes O$ then $(x', y') \in D\sigma$ since $\mathrm{supp}(\sigma) \cap O = \emptyset$. So, assume that $x' \in O$, $y' \in V'$ and $\neg x' \prec' y'$. By definition, we have that there exists in $y \in V$ such that $y' \in \mathrm{n}(y\sigma)$ and $\neg x' \prec y$. So, we have, by definition of $D$, $(x', y) \in D$ and since $x'\sigma = x'$ and $y\sigma = y'$, we have thus $(x', y') \in D\sigma$. So $D' \subseteq D\sigma$.

Let $(x', y') \in D\sigma$. By definition, there exists $(x, y) \in D$ such that $x' = x\sigma$ and $y' = y\sigma$. If $(x, y) \in O \otimes O$, then $x' = x$ and $y' = y$ and thus $(x', y') \in D'$. Now assume that $x \in O$, $y \in V$ and $\neg x \prec y$. Since $\mathrm{supp}(\sigma) \cap O = \emptyset$, we have $x' = x$. If $y' \in O$ then $(x', y') \in O \otimes O$ and $(x', y') \in D'$. Assume that $y' \notin O$. Then, by definition of $V'$, $y' \in V'$. We have, since $y' = y\sigma$, $y' \in \mathrm{n}(y\sigma)$ and since $\neg x' \prec y$, we have, by definition of $\prec'$, $\neg x' \prec' y'$ and thus $(x', y') \in D'$. So $D\sigma \subseteq D'$. ■

Finally, the following lemma studies how evolves the distinction corresponding to an environment when adding a fresh name to the constant part.

**Lemma 26** *Let $\mathsf{pe} = (O, V, \prec)$ be a $\mathrm{K}-$environment and $z$ a fresh name (i.e. neither in $O$, nor in $V$) and let $(D, O) = \mathrm{f}(\mathsf{pe})$.*

*Then $\mathrm{f}(\mathsf{pe} \oplus_{\mathrm{O}} z) = (D \ \cup \ \{z\} \otimes (O \cup V), O \cup \{z\})$.*

**Proof.**

Since $z$ is fresh, we have $\mathsf{pe} \oplus_{\mathrm{O}} z = (O \ \cup \{z\}, V, \prec)$.

So, by definition, we have $\mathrm{f}(\mathsf{pe} \oplus_{\mathrm{O}} z) = (D', O \cup \{z\})$ where the distinction $D'$ has been defined to be $D' = O \cup \{z\}^{\neq} \cup \bigcup_{n \in O \cup \{z\} \wedge x \in V \wedge \neg n \prec x} \{(n, x)\}^{=}$.

Thus $D' = O^{\neq} \cup \{z\} \otimes O \cup \bigcup_{n \in O \wedge x \in V \wedge \neg n \prec x} \{(n, x)\}^{=} \cup \bigcup_{x \in V} \{(z, x)\}^{=}$ because $z$ does not appear in $\prec$ and so for every $x \in V$ we have $\neg z \prec x$.

Hence $D' = D \ \cup \ \{z\} \otimes (O \cup V)$. ■

The K-open bisimilarity is sound with respect to T-open bisimilarity.

**Lemma 27** *Let $P, Q \in \mathcal{P}$ and $(O, V, \prec) \in \mathcal{K}$ such that $\mathrm{fn}(P{+}Q) \subseteq O \cup V$. Then we have:*

$$P \sim_{\mathrm{K}}^{(O,V,\prec)} Q \quad \Longrightarrow \quad P \sim_{\mathrm{T}}^{\mathrm{f}((O,V,\prec))} Q$$

Under the condition that the T-environment $(D, C)$ is representable by a K-environment $\mathsf{pe}$, T-open $(D, C)$-bisimilarity is sound with respect to K-open $\mathsf{pe}$-bisimilarity.

**Lemma 28** *Let $P, Q \in \mathcal{P}$ and $(D, C) \in \mathcal{F}$. Then we have*

$$P \sim_{\mathrm{T}}^{(D,C)} Q \quad \Longrightarrow \quad \forall V, \prec : \left( \begin{array}{ll} & C \cap V = \emptyset \\ \wedge & \mathrm{fn}(P{+}Q) \subseteq C \cup V \\ \wedge & (D, C) = \mathrm{f}((C, V, \prec)) \end{array} \Rightarrow P \sim_{\mathrm{K}}^{(C,V,\prec)} Q \right)$$

**Proof.**

Let $\mathcal{R} = \left\{ ((C,V,\prec),P,Q) \mid \begin{array}{c} P \sim_{\mathrm{T}}^{(D,C)} Q \ \wedge\ \mathrm{fn}(P+Q) \subseteq C \cup V \\ \wedge\ C \cap V = \emptyset \ \ \wedge\ (D,C) = \mathrm{f}((C,V,\prec)) \end{array} \right\}$.

We show that $\mathcal{R}$ is a K-open bisimulation.

Let $((C,V,\prec),P,Q) \in \mathcal{R}$. Let $\sigma$ such that $\sigma \blacktriangleright\!\blacktriangleright (C,V,\prec)$ and $P\sigma \xrightarrow{\mu} P'$.

First, let $\mathsf{pe} = (C,V,\prec)$.

By definition, there exists $D$ such that $P \sim_{\mathrm{T}}^{(D,C)} Q$ and $(D,C) = \mathrm{f}((C,V,\prec))$.

By Lemma 24, we have $\sigma \blacktriangleright \mathrm{f}(\mathsf{pe})$, i.e. $\sigma \blacktriangleright (D,C)$.

Since $P \sim_{\mathrm{T}}^{(D,C)} Q$ and $P\sigma \xrightarrow{\mu} P'$, we have that $Q\sigma \xrightarrow{\mu} Q'$ and

- if $\mu = (\nu z)\,\overline{a}\,z$ (with $z$ fresh), then $P' \sim_{\mathrm{T}}^{(D',C\cup\{z\})} Q'$ where $D' = D\sigma \ \cup\ \{z\} \otimes (\mathrm{fn}((P+Q)\sigma) \cup \mathrm{n}(D\sigma))$
- otherwise $P \sim_{\mathrm{T}}^{(D\sigma,C)} Q'$

So,

- if $\mu = \tau$, we have by Lemma 25, $(\mathsf{pe}^\sigma, P', Q') \in \mathcal{R}$
- if $\mu = a(x)$, still by Lemma 25 and because $\mathrm{f}(\mathsf{pe}^\sigma \oplus_{\mathrm{V}} x) = \mathrm{f}(\mathsf{pe}^\sigma)$, we have $(\mathsf{pe}^\sigma \oplus_{\mathrm{V}} x, P', Q') \in \mathcal{R}$
- if $\mu = \overline{a}\,z$, then by Lemma 25 and because $\mathsf{pe}^\sigma \oplus_{\mathrm{O}} z = \mathsf{pe}^\sigma$ (since $z$ is not fresh), we have $(\mathsf{pe}^\sigma \oplus_{\mathrm{O}} z, P', Q') \in \mathcal{R}$
- if $\mu = (\nu z)\,\overline{a}\,z$ (with $z$ fresh), by Lemma 25 and Lemma 26, we have $(\mathsf{pe}^\sigma \oplus_{\mathrm{O}} z, P', Q') \in \mathcal{R}$ (because the only difference between the updated distinction above and the distinction generated by $\mathrm{f}(\mathsf{pe}^\sigma \oplus_{\mathrm{O}} z)$ is the presence of some irrelevant names for the bisimilarity; the important fact is that $\mathrm{fn}(P+Q) \subseteq C \cup V$).

Hence $\mathcal{R}$ is a K-open bisimulation. ∎

### *3.3 About congruence properties*

In the following, we prove a conjecture we formulated in [16]. We prove with the help of K-open bisimilarity that, under some conditions, open $D$-bisimilarity is a congruence for a bigger class of contexts than just only $D$-respectful contexts. The idea is, if $(D,O) = \mathrm{f}((O,V,\prec))$, (1) to admit contexts that are $D$-respectful, and furthermore (2) to admit contexts where the hole occurs underneath an input prefix that binds a name $x$ of $V$, but only if, in addition, every name of $\{\ n \in O \mid \neg\, n \prec x\ \}$ appears underneath a respective restric-

tion on the "path" from the hole-binding input prefix for $x$ to the hole. This corresponds to the fact that, in the bisimulation, a name $n$ in $O$ comes from a restriction and a name $x$ from $V$ comes from an input prefix and we have $n \prec x$ if $n$ was disclosed before $x$ was input. Before going deeper into the formal details, let us understand the intuition by means of a simple example.

**Example 29** *Let $P = \overline{x} \,|\, y$ and $Q = \overline{x}.y + y.\overline{x}$.*
*It is known and easily verifiable that $P \sim_{\mathrm{O}}^{D} Q$ with $D = \{(x,y),(y,x)\}$.*

*Let $C = \{y\}$ and $V = \{x\}$, and note that $(D, C) = \mathrm{f}((C, V, \emptyset))$.*
*Observe that $P \sim_{\mathrm{K}}^{(C,V,\prec)} Q$.*

*Now, let us regard the context $X[\cdot] = a(x).(\nu y)\,[\cdot]$.*
*Then $X[P] \sim_{\mathrm{O}}^{\emptyset} X[Q]$, although $X[\cdot]$ is not considered by D-congruence.*
*However, $X[\cdot]$ follows our above informal rule of admissible contexts.*

*Finally, just note in passing that also $X[P] \sim_{\mathrm{K}}^{(\emptyset,\{a\},\emptyset)} X[Q]$.*

**Definition 30** *Let $D$ be a distinction and $x \in \mathcal{N}$.*
*We write $x \in D$ if there exists $y$ such that $(x,y) \in D$.*
*We write $D \setminus x$ for the distinction $\{(y,z) \in D \mid y \neq x \wedge z \neq x\}$.*

**Definition 31** *Let $\mathsf{pe} = (O, V, \prec) \in \mathcal{K}$ and $n \in \mathcal{N}$.*
*We define $\mathsf{pe}\backslash n \stackrel{\mathrm{def}}{=} \big( O\backslash\{n\}, V\backslash\{n\}, \prec \setminus (\{n\}\times\mathcal{N})^{=} \big)$.*

Note that if $\mathsf{pe}$ is a K-environment, then $\mathsf{pe}\backslash n$ is also a K-environment.

The following lemma states that, as for open bisimulation, only free names of processes are relevant in the sense that their consideration in environments matters.

**Lemma 32** *Let $P, Q \in \mathcal{P}$ and $\mathsf{pe} = (O, V, \prec) \in \mathcal{K}$ such that $P \sim_{\mathrm{K}}^{\mathsf{pe}} Q$ and $n_0 \notin \mathrm{fn}(P + Q)$. Then $P \sim_{\mathrm{K}}^{\mathsf{pe}\backslash n_0} Q$.*

**Proof.**

Assume that $P \sim_{\mathrm{K}}^{\mathsf{pe}} Q$ with $\mathsf{pe} = (O, V, \prec)$.

By Lemma 27, we have $P \sim_{\mathrm{T}}^{\mathrm{f}(\mathsf{pe})} Q$ where $\mathrm{f}(\mathsf{pe}) = (D, O)$ and with $D$ such that $D = O^{\neq} \cup \bigcup_{(n,x)\in O\times V} \{(n,x),(x,n) \mid \neg\, n \prec x\}$.

By Lemma 14, we then have that $P \sim_{\mathrm{O}}^{D} Q$.

Since $n_0 \notin \mathrm{fn}(P + Q)$, we have that $P \sim_{\mathrm{O}}^{D\backslash n_0} Q$.

By definition, $D \setminus n_0 = D \setminus (\{n_0\} \times \mathcal{N} \cup \mathcal{N} \times \{n_0\})$.

Let $O' = O \setminus \{n_0\}$, $V' = V \setminus \{n_0\}$ and $\prec' = \prec \setminus (\{n_0\} \times \mathcal{N} \cup \mathcal{N} \times \{n_0\})$.

We then have that $D \setminus n_0 = O'^{\neq} \cup \bigcup_{(n,x) \in O' \times V'} \{(n,x),(x,n) \mid \neg n \prec' x\}$.

Thus, by Lemma 15, we have that $P \sim_{\mathrm{T}}^{(D \setminus n_0, O')} Q$.

Moreover, since $(D \setminus n_0, O') = \mathrm{f}(\mathsf{pe} \setminus n_0)$, we have by Lemma 28 that $P \sim_{\mathrm{K}}^{\mathsf{pe} \setminus n_0} Q$. Hence the result. ■

The next lemma states that an adversary with less knowledge—in the sense that it explicitly disposes of fewer known names at the moment of a process input—distinguishes fewer processes.

**Lemma 33** *Let $P, Q \in \mathcal{P}$ and $\mathsf{pe} = (O, V, \prec) \in \mathcal{K}$ such that $P \sim_{\mathrm{K}}^{\mathsf{pe}} Q$ and $(n_0, x_0) \in \prec$. Then $P \sim_{\mathrm{K}}^{(O, V, \prec \setminus \{(n_0,x_0)\})} Q$.*

**Proof.**

Assume that $P \sim_{\mathrm{K}}^{\mathsf{pe}} Q$ with $\mathsf{pe} = (O, V, \prec)$.

By Lemma 27, we have $P \sim_{\mathrm{T}}^{\mathrm{f}(\mathsf{pe})} Q$ where $\mathrm{f}(\mathsf{pe}) = (D, O)$ and with $D$ such that $D = O^{\neq} \cup \bigcup_{(n,x) \in O \times V} \{(n,x),(x,n) \mid \neg n \prec x\}$.

By Lemma 14, we then have that $P \sim_{\mathrm{O}}^{D} Q$.

Let $D' = D \cup \{(n_0, x_0), (x_0, n_0)\}$.

Since $D \subseteq D'$, we also have that $P \sim_{\mathrm{O}}^{D'} Q$.

Then, by Lemma 15, we have $P \sim_{\mathrm{T}}^{(D', O)} Q$ (because $O$ is such that $O^{\neq} \subseteq D'$).

Moreover, we have $(D', O) = \mathrm{f}((O, V, \prec \setminus \{(n_0, x_0)\}))$, thus by Lemma 28, we have that $P \sim_{\mathrm{K}}^{(O, V, \prec \setminus \{(n_0,x_0)\})} Q$. Hence the result. ■

Before rephrasing Proposition 7 in terms of K-open bisimulation, we extend the second part of Definition 17 to a finite set of names (this is because K-open bisimulation requires the *initial* environment to mention every free name).

**Definition 34** *Let $\mathsf{pe} = (O, V, \prec)$ be a* K*-environment and $N = \{n_1, \ldots, n_k\}$ a finite set of names.*

*We define $\mathsf{pe} \oplus_{\mathrm{V}} N$ to be $\mathsf{pe}_k$ where*

- $\mathsf{pe}_0 = \mathsf{pe}$

- $\mathsf{pe}_{i+1} = \begin{cases} \mathsf{pe}_i \oplus_{\mathrm{V}} n_i & \text{if } n_i \notin (O \cup V) \\ \mathsf{pe}_i & \text{otherwise} \end{cases}$

Note that the previous definition is independent of the order in which we add the elements of $N$.

The following result is obviously true:

**Lemma 35** *Let $\mathsf{pe} = (O, V, \prec)$ be a* K*-environment, $(D, O) \stackrel{\text{def}}{=} \mathrm{f}(\mathsf{pe})$ and $N$ a finite set of names. Then $\mathrm{f}(\mathsf{pe} \oplus_{\mathrm{V}} N) = (D, O)$.*

In analogy with Sangiorgi's congruence results for standard open bisimulation as of Proposition 7, we state one for K-open bisimulation.

**Proposition 36** *Let $P$, $Q$ two processes and $\mathsf{pe} = (O, V, \prec)$ a* K*-environmentwith $P \sim_{\mathrm{K}}^{\mathsf{pe}} Q$. Then,*

*(1)* $\forall R : R \,|\, P \sim_{\mathrm{K}}^{\mathsf{pe} \oplus_{\mathrm{V}} \mathrm{fn}(R)} R \,|\, Q$ *and* $P \,|\, R \sim_{\mathrm{K}}^{\mathsf{pe} \oplus_{\mathrm{V}} \mathrm{fn}(R)} Q \,|\, R$
*(2)* $\forall R : R + P \sim_{\mathrm{K}}^{\mathsf{pe} \oplus_{\mathrm{V}} \mathrm{fn}(R)} R + Q$ *and* $P + R \sim_{\mathrm{K}}^{\mathsf{pe} \oplus_{\mathrm{V}} \mathrm{fn}(R)} Q + R$
*(3)* $! P \sim_{\mathrm{K}}^{\mathsf{pe}} ! Q$
*(4)* $\forall \phi : \phi P \sim_{\mathrm{K}}^{\mathsf{pe} \oplus_{\mathrm{V}} \mathrm{n}(\phi)} \phi Q$
*(5)* $\forall a, z : \overline{a}\langle z \rangle.P \sim_{\mathrm{K}}^{\mathsf{pe} \oplus_{\mathrm{V}} \{a,z\}} \overline{a}\langle z \rangle.Q$
*(6)* $\forall x : (\nu x)\, P \sim_{\mathrm{K}}^{\mathsf{pe} \setminus x} (\nu x)\, Q$
*(7)* $\forall a, x : x \notin (O \cup V) \implies a(x).P \sim_{\mathrm{K}}^{\mathsf{pe} \oplus_{\mathrm{V}} \{a\}} a(x).Q$
*(8)* $\forall a, x : x \in V \wedge \{n \in O \mid \neg\, n \prec x\} = \emptyset \implies a(x).P \sim_{\mathrm{K}}^{\mathsf{pe} \oplus_{\mathrm{V}} \{a\}} a(x).Q$

**Proof.**

The proof is easy and mainly uses Lemmas 27, 14, 15, 28, 32 and 35.

The condition for the input context is the translation in terms of K-environments of the condition of Proposition 7. Indeed, if $(D, O) = \mathrm{f}(\mathsf{pe})$ we have, by definition of $\mathrm{f}(\mathsf{pe})$, that

$$x \in D \quad \Longrightarrow \quad x \in O \vee (x \in V \wedge \{n \in O \mid \neg\, n \prec x\} \neq \emptyset)$$

■

From the previous proposition, we can deduce a set of contexts that are safe concerning K-open bisimulation; for such contexts $C[\cdot]$ we have that if $P \sim_{\mathrm{K}}^{\mathsf{pe}} Q$, then there exists $\mathsf{pe}'$ such that $C[P] \sim_{\mathrm{K}}^{\mathsf{pe}'} C[Q]$.

**Definition 37** *Let $\mathsf{pe} = (O, V, \prec)$ be a* K*-environment. We define the set of* $\mathsf{pe}$-respectful *contexts as the language generated by the grammar defined as follows. For each subset $N \subseteq O$, we define a non-terminal symbol $C_N[\cdot]$. The start symbol is $C_\emptyset[\cdot]$. The production rules are of the form:*

$$
\begin{array}{rcll}
C_N[\cdot] & ::= & [\cdot] & \text{if } N = \emptyset \\
& \Big| & P \,|\, C_N[\cdot] \,\Big|\, C_N[\cdot] \,|\, P & \\
& \Big| & P + C_N[\cdot] \,\Big|\, C_N[\cdot] + P & \\
& \Big| & !\, C_N[\cdot] & \\
& \Big| & \phi C_N[\cdot] & \\
& \Big| & (\nu x)\, C_{N\setminus\{x\}}[\cdot] & \\
& \Big| & \overline{a}\langle z\rangle.C_N[\cdot] & \\
& \Big| & a(x).C_N[\cdot] & \text{if } x \notin O \cup V \\
& \Big| & a(x).C_{N\cup N'}[\cdot] & \text{if } x \in V \text{ and } N' = \{n \in O \mid \neg\, n \prec x\}
\end{array}
$$

The idea is simply that when a name $x$ of $V$ is bound by an input prefix, then according to Proposition 36, it is sufficient that every name $n \in O$ such that $\neg\, n \prec x$ is removed from the environment, which is done via restrictions. The index $N$ of each non-terminal $C_N[\cdot]$ simply remembers all such names.

**Example 38** *Back to Example 29, we have in this case* $\mathsf{pe} = (\{y\}, \{x\}, \emptyset)$.

*The context* $X[\cdot] = a(x).(\nu y)\,[\cdot]$ *is obtained by applying the second rule of formation for input prefix (since* $x \in \{x\}$*) and at this point the name* $y$ *is pushed in the set of names* $N$*. Then the rule for the restriction is used to remove* $y$ *from the set* $N$*. Finally the hole is placed.*

*The derivation path for obtaining* $X[\cdot]$ *via the grammar of Definition 37 is*

$$C_\emptyset[\cdot] \to a(x).C_{\{y\}}[\cdot] \to a(x).(\nu y)\, C_\emptyset[\cdot] \to a(x).(\nu y)\,[\cdot]$$

**Lemma 39** *Let* $\mathsf{pe}$ *be a* K*-environment,* $P$ *and* $Q$ *two processes and* $C[\cdot]$ *an* $\mathsf{pe}$*-respectful context. Assume that* $P \sim_{\mathrm{K}}^{\mathsf{pe}} Q$*. Then there exists a* K*-environment* $\mathsf{pe}'$ *such that* $C[P] \sim_{\mathrm{K}}^{\mathsf{pe}'} C[Q]$ *(and* $\mathsf{pe}'$ *is built according to rules given in Proposition 36).*

**Proof.**

This is a simple corollary of the previous observations/results. ∎

**Definition 40** *Let* $\mathsf{pe}$ *be a* K*-environment.*
*A relation* $\mathcal{R} \subset \mathcal{P} \times \mathcal{P}$ *is an* $\mathsf{pe}$-congruence *if for all* $P, Q$ *with* $(P, Q) \in \mathcal{R}$ *and for all* $\mathsf{pe}$*-respectful contexts* $C[\cdot]$ *we have* $(C[P], C[Q]) \in \mathcal{R}$*.*

**Lemma 41** *Let* $\mathsf{pe}$ *be a* K*-environment and* $(D, O) = \mathrm{f}(\mathsf{pe})$*.*
*Then, every* $\mathsf{pe}$*-congruence is also a* $D$*-congruence.*

The following theorem states that open $D$-bisimilarity has better congruence properties than those expressed by $D$-congruence.

**Theorem 42** *Let* $\mathsf{pe}$ *be a* K*-environment and* $(D, O) = \mathrm{f}(\mathsf{pe})$.
*Then, open* $D$*-bisimilarity is an* $\mathsf{pe}$*-congruence.*

**Proof.**

Again, a simple corollary of the previously stated results. ∎

## 4 Bisimulation in the spi-calculus

### *4.1 Syntax and semantics*

The spi-calculus is a process calculus that was introduced by Abadi and Gordon [7] to model and study cryptographic protocols.

The syntax of the spi-calculus is given by Table 1 and Table 3. We have chosen to focus the study of this paper on a shared-key cryptosystem but the message language can be extended to deal with public/private key, pairing and/or hashing (see [15] or [19] for more details).

The so-called late semantics of the spi-calculus has already been defined in Section 2.2.

### *4.2 Late hedged bisimulation*

Abadi and Gordon first noticed that the classical notion of bisimulation as commonly used in the $\pi$-calculus was not adequate for the spi-calculus. The reason is that the latter required an explicit treatment of the knowledge that an observer—in the spi-calculus: an attacker—has possibly acquired over time. Therefore, Abadi and Gordon proposed in [20] an "environment-sensitive" notion of bisimulation that they called *framed bisimulation* . Based on it, *hedged bisimulation* is a variant of environment-sensitive bisimulation that has been shown in [14] to coincide with barbed equivalence (contrary to framed bisimulation). The definition of hedged bisimulation that we use in this paper, as presented in this section, is the *late* variant of the *early* version that was studied in [14]. Here, we briefly review and discuss the main concepts.

*4.2.1 Environments: Hedges*

In this kind of environment, we list pairs of messages that are supposed to be indistinguishable for the attacker. Roughly, one may understand these pairs as being received from two processes in a respective bisimulation game, so a hedge represents an attacker's current knowledge.

Compared to the frame-theory pair used in framed bisimulation, a hedge consist of just keeping the theory and including corresponding names as part of the theory (see [14] for a more detailed comparison).

**Definition 43 (hedge)** *A* hedge *is a finite subset of $\mathcal{M} \times \mathcal{M}$. The set of all hedges is denoted by* $\mathbf{H}$.

We need some further standard algebraic notation to work with hedges.

**Definition 44** *If $C \subseteq A \times B$ for some sets $A$ and $B$, we define*

- $\pi_1(C) \stackrel{\text{def}}{=} \{a \in A \mid \exists b \in B : (a, b) \in C\}$,
- $\pi_2(C) \stackrel{\text{def}}{=} \{b \in B \mid \exists a \in A : (a, b) \in C\}$, *and*
- $C^{-1} = \{(b, a) \in A \times B \mid (a, b) \in C\}$.

In the above definition, we prefer to speak of first projection and second projection instead of domain and range because we see the set $C$ as a flat set of pairs rather than a relation.

If $h$ is a hedge, we define in straightforward manner the *synthesis* $\mathcal{S}(h)$ of $h$ (i.e., whatever message pairs can be constructed from the knowledge contained in $h$), the *analysis* $\mathcal{A}(h)$ of $h$ (i.e., whatever message pairs can be found out by decomposing the knowledge contained in $h$) and the *irreducibles* $\mathcal{I}(h)$ of $h$ (i.e., reducing the knowledge contained in $h$ to its seeds).

**Definition 45 (synthesis, analysis, irreducibles)** *Let $h$ be a hedge.*

*The synthesis $\mathcal{S}(h)$ of $h$ is the smallest subset of $\mathcal{M} \times \mathcal{M}$ containing $h$ and satisfying:*

$$(\text{SYN-ENC})\ \frac{(M, N) \in \mathcal{S}(h) \qquad (K, L) \in \mathcal{S}(h)}{(\mathrm{E}_K(M), \mathrm{E}_L(N)) \in \mathcal{S}(h)}$$

*The analysis $\mathcal{A}(h)$ of $h$ is the smallest hedge containing $h$ and satisfying:*

$$(\text{ANA-DEC})\ \frac{(\mathrm{E}_K(M), \mathrm{E}_L(N)) \in \mathcal{A}(h) \qquad (K, L) \in \mathcal{S}(\mathcal{A}(h))}{(M, N) \in \mathcal{A}(h)}$$

*Finally, the irreducibles $\mathcal{I}(h)$ of $h$ is defined by:*

$$\mathcal{I}(h) \stackrel{\text{def}}{=} \mathcal{A}(h) \setminus \{(\mathrm{E}_K(M), \mathrm{E}_L(N)) \in \mathcal{A}(h) \mid (K, L) \in \mathcal{S}(\mathcal{A}(h))\}$$

**Example 46** *(1) Consider the hedge $h_1 = \{(m, n), (k, k)\}$.*
*Then, we have $(\mathrm{E}_k(m), \mathrm{E}_k(n)) \in \mathcal{S}(h_1)$.*
*Moreover, we have $\mathcal{I}(h_1) = \mathcal{A}(h_1) = h_1$.*
*(2) Consider the hedge $h_2 = \{(k, k), (\mathrm{E}_k(m), \mathrm{E}_k(n))\}$.*
*Then we have $\mathcal{A}(h_2) = \{(k, k), (\mathrm{E}_k(m), \mathrm{E}_k(n)), (m, n)\}$.*
*And we have $\mathcal{I}(h_2) = \{(k, k), (m, n)\}$.*
*(3) Consider $h_3 = \Big\{(k, k), (\mathrm{E}_{\mathrm{E}_k(a)}(m), \mathrm{E}_{\mathrm{E}_k(a)}(n)), (a, a)\Big\}$.*
*Then we have $\mathcal{A}(h_3) = \Big\{(k, k), (\mathrm{E}_{\mathrm{E}_k(a)}(m), \mathrm{E}_{\mathrm{E}_k(a)}(n)), (a, a), (m, n)\Big\}$.*
*And we have $\mathcal{I}(h_3) = \{(k, k), (m, n), (a, a)\}$.*

We now give some results relating hedges and operations on them.

**Proposition 47** *Let $g, h$ be two hedges. We have*

- *if $h \neq \emptyset$ then $\mathcal{S}(h)$ is infinite*
- $\mathcal{S}(h) \subset \mathcal{S}(\mathcal{A}(h)) = \mathcal{S}(\mathcal{I}(h))$
- $\mathcal{A}(\mathcal{A}(h)) = \mathcal{A}(h)$
- $\mathcal{I}(\mathcal{I}(h)) = \mathcal{I}(h)$
- *if $\mathcal{S}(g) \subset \mathcal{S}(h)$ then $\mathcal{S}(\mathcal{A}(g)) \subset \mathcal{S}(\mathcal{A}(h))$ and $\mathcal{S}(\mathcal{I}(g)) \subset \mathcal{S}(\mathcal{I}(h))$*
- $\mathcal{I}(\mathcal{I}(h) \cup g) = \mathcal{I}(h \cup g)$

**Proof.**

The proofs can be found for example in [19]. ∎

The notion of *consistency* can be seen essentially as a characterisation of hedges in which the decryption power on both sides of the pairs coincides, together with the fact that a message on one side cannot be related to two different messages on the other side, as well as an irreducibility condition. Here, we build it up from an asymmetric version.

**Definition 48 (consistency)** *A hedge $h$ is* left-consistent *if for all $(M, N) \in h$, we have*

*(1)* $M \in \mathcal{N} \Rightarrow N \in \mathcal{N}$
*(2)* $\forall (M', N') \in h : M = M' \Rightarrow N = N'$
*(3) if $M = \mathrm{E}_K(M')$ then $K \notin \pi_1(\mathcal{S}(h))$*

*A hedge $h$ is* consistent *if both $h$ and $h^{-1}$ are left-consistent.*

**Example 49** *In this example, we illustrate briefly the three clauses in the above definition of consistency. In the three following examples, we assume*

*that the channel $a$ is public.*

*(1) First consider*

$$P_1 \stackrel{\text{def}}{=} \overline{a}\langle b\rangle.\,\mathbf{0}$$

$$Q_1 \stackrel{\text{def}}{=} \overline{a}\langle \mathrm{E}_k(m)\rangle.\,\mathbf{0}$$

*Then, these two processes can be distinguished by*

$$R_1 \stackrel{\text{def}}{=} a(x).(\overline{x}\langle z\rangle.\,\mathbf{0} \mid x(z).\overline{\omega}\langle\omega\rangle.\,\mathbf{0})$$

*(or even simpler by $R_1' \stackrel{\text{def}}{=} a(x).[x{:}\mathcal{N}]\overline{\omega}\langle\omega\rangle.\,\mathbf{0}$).*

*Indeed, it is possible for the adversary to detect that in the first case, the output message is just a name whereas in the second case it is a complex message. The first clause of consistency detects this situation.*

*The corresponding hedge would be*

$$h_1 \stackrel{\text{def}}{=} \{(a,a),(b,\mathrm{E}_k(m))\}$$

*which is not consistent because it violates the first clause.*

*(2) Now consider*

$$P_2 \stackrel{\text{def}}{=} (\nu k,m)\,\overline{a}\langle \mathrm{E}_k(m)\rangle.\overline{a}\langle \mathrm{E}_k(m)\rangle.\,\mathbf{0}$$

$$Q_2 \stackrel{\text{def}}{=} (\nu k,m,n)\,\overline{a}\langle \mathrm{E}_k(m)\rangle.\overline{a}\langle \mathrm{E}_k(n)\rangle.\,\mathbf{0}$$

*Then, these two processes can be distinguished by*

$$R_2 \stackrel{\text{def}}{=} a(x).a(y).[x{=}y]\overline{\omega}\langle\omega\rangle.\,\mathbf{0}$$

*Indeed, the adversary can detect that in the first case the two emitted message are the same whereas in the second case they are different.*

*The corresponding hedge would be*

$$h_2 \stackrel{\text{def}}{=} \{(a,a),(\mathrm{E}_k(m),\mathrm{E}_k(m)),(\mathrm{E}_k(m),\mathrm{E}_k(n))\}$$

*which is not consistent because it violates the second clause.*

*(3) Consider finally*

$$P_3 \stackrel{\text{def}}{=} \overline{a}\langle \mathrm{E}_a(m)\rangle.\,\mathbf{0}$$

$$Q_3 \stackrel{\text{def}}{=} (\nu k)\,\overline{a}\langle \mathrm{E}_k(m)\rangle.\,\mathbf{0}$$

*Then, these two processes can be distinguished by*

$$R_3 \stackrel{\text{def}}{=} a(x).[\mathrm{D}_a(x){:}\mathcal{M}]\overline{\omega}\langle\omega\rangle.\,\mathbf{0}$$

*Indeed, the adversary can try and succeed in decrypting the emitted message in the first case whereas in the second case, the decryption would fail.*

*The corresponding hedge would be*

$$h_3 \stackrel{\text{def}}{=} \{(a, a), (\mathrm{E}_a(m), \mathrm{E}_k(m))\}$$

*which is not consistent because it violates the third clause.*

Finally, we recall that the reader who is interested in richer message languages (that include further cryptographic operators) or in seeing formal definitions about hedges is invited to consult [19]; in particular, the definition of analysis is given with great precision and it is shown how to extend the definition of consistency.

### *4.2.2 Late hedged bisimulation*

First, we lift the notion of consistency to the level of environment-sensitive relations on processes, here for environments being hedges.

**Definition 50 (hedged relation)** *A* hedged relation $\mathcal{R}$ *is a subset of* $\mathbf{H} \times \mathcal{P} \times \mathcal{P}$ *such that* $\forall (h, P, Q) \in \mathcal{R} : \mathrm{fn}(P) \subset \mathrm{n}(\pi_1(h)) \wedge \mathrm{fn}(Q) \subset \mathrm{n}(\pi_2(h))$.

*A hedged relation* $\mathcal{R}$ *is called*

- consistent *if* $\forall (h, P, Q) \in \mathcal{R} : h$ *is consistent;*
- symmetric *if* $\forall (h, P, Q) : (h, P, Q) \in \mathcal{R} \Leftrightarrow (h^{-1}, Q, P) \in \mathcal{R}$

The bisimulation relation is now defined to keep track of hedges under transition. For this, transitions are only enabled after checking that the labels could have been generated w.r.t. the attacker's knowledge, possibly under invention of additional names (listed in $B$), and after transition the hedge component needs to be properly updated, by design decision including the reduction of an updated hedge to its seeds.

**Definition 51 (late hedged bisimulation)** *A symmetric consistent hedged relation* $\mathcal{R}$ *is a* late hedged bisimulation *if for all* $(h, P, Q) \in \mathcal{R}$*, if* $P \xrightarrow{\mu_1} P'$ *with* $\mathrm{bn}(\mu_1) \cap \mathrm{n}(\pi_1(h)) = \emptyset$ *and* $\mathrm{ch}(\mu_1) \in \pi_1(h)$ *(if* $\mu_1 \neq \tau$*), then there exists* $Q'$ *and* $\mu_2$ *such that* $Q \xrightarrow{\mu_2} Q'$ *with* $\mathrm{bn}(\mu_2) \cap \mathrm{n}(\pi_2(h)) = \emptyset$ *and*

- *if* $\mu_1 = \tau$ *then* $\mu_2 = \tau$ *and* $(h, P', Q') \in \mathcal{R}$
- *if* $\mu_1 = a_1(x_1)$ *then* $\mu_2 = a_2(x_2)$ *where* $(a_1, a_2) \in \mathcal{S}(h)$ *and for all* $B \subseteq \mathcal{N} \times \mathcal{N}$ *consistent,* $M_1, M_2 \in \mathcal{M}$ *such that*
  - $\pi_1(B) \setminus \mathrm{n}(M_1) = \emptyset$
  - $\pi_1(B) \cap \mathrm{n}(\pi_1(h)) = \emptyset = \pi_2(B) \cap \mathrm{n}(\pi_2(h))$
  - $(M_1, M_2) \in \mathcal{S}(h \cup B)$

  *we have* $(h \cup B, P'\{^{M_1}/_{x_1}\}, Q'\{^{M_2}/_{x_2}\}) \in \mathcal{R}$

- *if* $\mu_1 = (\nu \tilde{c})\,\overline{a_1}\,M_1$ *then* $\mu_2 = (\nu \tilde{d})\,\overline{a_2}\,M_2$ *where* $(a_1, a_2) \in \mathcal{S}(h)$ *and* $(\mathcal{I}(h \cup \{(M_1, M_2)\}), P', Q') \in \mathcal{R}$

**Definition 52 (late hedged bisimilarity)** *Let* $P, Q \in \mathcal{P}$ *and* $h \in \mathbf{H}$ *such that* $\mathrm{fn}(P) \subseteq \mathrm{n}(\pi_1(h))$ *and* $\mathrm{fn}(Q) \subseteq \mathrm{n}(\pi_2(h))$.
*Then,* $P$ *and* $Q$ *are called* late $h$-hedged bisimilar*, written* $P \sim^{h}_{\mathrm{LH}} Q$*, if there exists a late hedged bisimulation* $\mathcal{R}$ *such that* $(h, P, Q) \in \mathcal{R}$.

We can now further explain why a notion of indistinguishability should be encoded in environments. Consider the process $P(M) = (\nu k)\,\overline{a}\langle \mathrm{E}_k(M)\rangle.\,\mathbf{0}$. Since the fresh key $k$ is never disclosed, it is reasonable to consider $P(M)$ equivalent to $P(M')$, for any pair $(M, M')$ of messages. In late hedged bisimilarity, the hedge contains the pair $(\mathrm{E}_k(M), \mathrm{E}_k(M'))$ to reflect the fact that these two messages cannot be distinguished by the environment (the attacker). Actually, since it is possible to hide a different number of names on each side of the bisimulation game (thanks to the encryption primitive), it is not required that freshly created names (or input variables) are exactly the same, as it can be observed in Definition 51. This fact permits to consider, in the bisimulation, indistinguishable actions instead of requiring to have the same actions on both sides.

### *4.3 An example of late hedged bisimulation*

**Example 53** *We consider for a message* $M$ *such that* $k \notin \mathrm{n}(M)$*, the processes*

$$
\begin{aligned}
A &\stackrel{\mathrm{def}}{=} \overline{a}\langle \mathrm{E}_k(M)\rangle.\,\mathbf{0}\\
B &\stackrel{\mathrm{def}}{=} a(x).\overline{a}\langle \mathrm{D}_k(x)\rangle.\,\mathbf{0}\\
\underline{B} &\stackrel{\mathrm{def}}{=} a(x).[\mathrm{D}_k(x) : \mathcal{M}]\overline{a}\langle M\rangle.\,\mathbf{0}\\
P &\stackrel{\mathrm{def}}{=} (\nu k)\,(A \mid B)\\
\underline{P} &\stackrel{\mathrm{def}}{=} (\nu k)\,(A \mid \underline{B})
\end{aligned}
$$

*Intuitively, the process* $P$ *is composed of two principals* $A$ *and* $B$ *that shares a secret key* $k$*.* $A$ *sends the message* $M$ *encrypted with* $k$ *along the channel* $a$ *while* $B$ *waits on channel* $a$ *for some message to be bound to variable* $x$*. Then* $B$ *tries to send the result of the decryption of this message by* $k$ *along the channel* $a$*.*

*The corresponding* protocol narration *is given Figure 1. It uses the annotations we proposed in [21].*

*Since the key* $k$ *is never disclosed, the only message that* $B$ *can receive from outside that is encrypted with* $k$ *is* $\mathrm{E}_k(M)$*.*

$$\textbf{private } k$$

$$A \textbf{ knows } M$$

$$A \rightsquigarrow B : \mathrm{E}_k(M)$$

$$B \rightsquigarrow A : M$$

Fig. 1. A simple cryptographic protocol

*So this seems natural to consider $P$ equivalent to $\underline{P}$ where $B$ is replaced by $\underline{B}$ which instead checks that the received message is encrypted with $k$ and then sends $M$ along the channel $a$.*

*Note that this kind of equation for proving authenticity results was first introduced by Abadi and Gordon in [7].*

*We thus prove that $P \sim^{h_0}_{\mathrm{LH}} \underline{P}$ where $h_0 = \mathcal{I}(\{(a,a),(M,M)\})$.*

*We first define the following shortcuts:*

$$\begin{aligned} h_0 &\stackrel{\text{def}}{=} \mathcal{I}(\{(a,a),(M,M)\}) \\ h_1(k,l) &\stackrel{\text{def}}{=} \mathcal{I}(h_0 \cup \{(\mathrm{E}_k(M), \mathrm{E}_l(M))\}) \\ h_1 &\stackrel{\text{def}}{=} h_1(k,k) \end{aligned}$$

*and*

$$\begin{aligned} B_1(N,k) &\stackrel{\text{def}}{=} \overline{a}\langle \mathrm{D}_k(N)\rangle.\,\mathbf{0} \\ B_1 &\stackrel{\text{def}}{=} B_1(\mathrm{E}_k(M),k) \\ \underline{B}_1(N,k) &\stackrel{\text{def}}{=} [\,\mathrm{D}_k(N) : \mathcal{M}\,]\overline{a}\langle M\rangle.\,\mathbf{0} \\ \underline{B}_1 &\stackrel{\text{def}}{=} \underline{B}_1(\mathrm{E}_k(M),k) \\ P_1(N,k) &\stackrel{\text{def}}{=} (\nu k)\,(A \mid B_1(N,k)) \\ \underline{P}_1(N,k) &\stackrel{\text{def}}{=} (\nu k)\,(A \mid \underline{B}_1(N,k)) \\ P_2 &\stackrel{\text{def}}{=} (\nu k)\,\overline{a}\langle \mathrm{D}_k(\mathrm{E}_k(M))\rangle.\,\mathbf{0} \\ \underline{P}_2 &\stackrel{\text{def}}{=} (\nu k)\ \ [\,\mathrm{D}_k(\mathrm{E}_k(M)) : \mathcal{M}\,]\ \overline{a}\langle M\rangle.\,\mathbf{0} \end{aligned}$$

*Let*

$$\mathcal{R} = \{(h_0, P, \underline{P})\} \cup \mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3$$

*where*

$$\mathcal{R}_1 = \{(h_0, P_2, \underline{P}_2)\,, (h_0, \mathbf{0}, \mathbf{0})\}$$

$$
\begin{aligned}
\mathcal{R}_2 = \quad & \{(h_1, B, \underline{B}) \mid k \textit{ fresh}\} \\
& \cup \{(h_1, B_1, \underline{B}_1) \mid k \textit{ fresh}\} \\
& \cup \left\{ \begin{array}{l} (h_1 \cup \{(x_1, y_1), \ldots, (x_n, y_n)\}, B_1(N_1, k), \underline{B}_1(N_2, k)) \\ \textit{where } k \textit{ fresh} \\ \textit{and } x_1, \ldots, x_n \textit{ fresh on left} \\ \textit{and } y_1, \ldots, y_n \textit{ fresh on right} \\ \textit{and for all } i,\ x_i \in \mathrm{n}(N_1) \\ \textit{and } (N_1, N_2) \in \mathcal{S}(h_1 \cup \{(x_1, y_1), \ldots, (x_n, y_n)\}) \\ \textit{and } N_1 \neq \mathrm{E}_k(M) \end{array} \right\} \\
& \cup \{(h_1, \mathbf{0}, \mathbf{0})\}
\end{aligned}
$$

*and*

$$
\begin{aligned}
\mathcal{R}_3 = & \left\{ \begin{array}{l} (h_0 \cup \{(x_1, y_1), \ldots, (x_n, y_n)\}, P_1(N_1, k), \underline{P}_1(N_2, l)) \\ \textit{where } k, x_1, \ldots, x_n \textit{ fresh on left} \\ \textit{and } l, y_1, \ldots, y_n \textit{ fresh on right} \\ \textit{and for all } i,\ x_i \in \mathrm{n}(N_1) \\ \textit{and } (N_1, N_2) \in \mathcal{S}(h_0 \cup \{(x_1, y_1), \ldots, (x_n, y_n)\}) \end{array} \right\} \\
\cup & \left\{ \begin{array}{l} \left(h_1(k, l) \cup \left\{(x'_1, y'_1), \ldots, (x'_p, y'_p)\right\}, B_1(N_1, k), \underline{B}_1(N_2, l)\right) \\ \textit{where } k, x_1, \ldots, x_n \textit{ fresh on left} \\ \textit{and } l, y_1, \ldots, y_n \textit{ fresh on right} \\ \textit{and for all } i,\ x_i \in \mathrm{n}(N_1) \\ \textit{and } (N_1, N_2) \in \mathcal{S}(h_0 \cup \{(x_1, y_1), \ldots, (x_n, y_n)\}) \end{array} \right\}
\end{aligned}
$$

*Then, the "symmetric" closure of $\mathcal{R}$ is a late hedged bisimulation.*

## 5 Open hedged bisimulation

We now present an extension of K-open bisimulation to the case of the spi-calculus. Several of the following ideas have already been developed in [22].

It is not sufficient to consider as S-environment a simple extension of K-environment by saying that a S-environment is a triple $(O, V, \prec)$ where $O$ would be a set of messages $V$ a (finite) set of names, and $\prec$ a subset of $O \times V$. One reason is that it would not be possible to build up an indistinguishability relation on top of this data. Thus, as with hedges, we split the sets $O$ and $V$ and obtain respectively a set of message pairs $h$ and a set of name pairs $v$. Another reason is that in the spi-calculus, unlike the $\pi$-calculus, we need to record that some names that were at some moment considered as channels must not later on by replaced by complex messages.

The intuition behind a S-environment $\mathsf{se} = (h, v, \prec, (\gamma_l, \gamma_r))$ is then as for K-environments. The hedge $h$ represents the messages emitted by the two players; likewise, $v$ represents the names input by these two players; the relation $\prec$ stores the time precedence between the emitted messages and the input names (thus a message containing $x$ cannot have been emitted before the name $x$ had been input); the pair $(\gamma_l, \gamma_r)$ is an additional component that tells which input names must remain names and not become arbitrary messages.

**Definition 54 (S-environment)**
*The quadruple* $\mathsf{se} = (h, v, \prec, (\gamma_l, \gamma_r))$ *is a S-environment if* $h \subseteq \mathcal{M} \times \mathcal{M}$, $v \subseteq \mathcal{N} \times \mathcal{N}$ *are two finite sets such that* $h \cap v = \emptyset$, $\prec \subseteq h \times v$, $\gamma_l \subseteq \pi_1(v)$ *and* $\gamma_r \subseteq \pi_2(v)$ *such that*

$$\forall (M, N) \in h, (x, y) \in v : (M, N) \prec (x, y) \Rightarrow x \notin \mathrm{n}(M) \wedge y \notin \mathrm{n}(N)$$

*The set of all S-environments is* $\mathcal{S}_{\mathrm{H}}$.

*For* $(x, y) \in v$, *we define* $h^{\prec}_{(x,y)} \stackrel{\text{def}}{=} \{\ (M, N) \in h \mid (M, N) \prec (x, y)\ \}$.

*We define* $\mathsf{se}^{-1} \stackrel{\text{def}}{=} (h^{-1}, v^{-1}, \prec^{-1}, (\gamma_r, \gamma_l))$
*where* $\prec^{-1} = \{((N, M), (y, x)) \mid (M, N) \prec (x, y)\}$.

*We define* $\mathrm{n}_1(\mathsf{se}) \stackrel{\text{def}}{=} \mathrm{n}(\pi_1(h \cup v))$ *and* $\mathrm{n}_2(\mathsf{se}) \stackrel{\text{def}}{=} \mathrm{n}(\pi_2(h \cup v))$.

*We define* $\mathrm{H}(\mathsf{se}) = \mathcal{I}(h \cup v)$ *and* $\mathcal{S}(\mathsf{se}) = \mathcal{S}(\mathrm{H}(\mathsf{se}))$.

**Example 55** *Let* $a, x \in \mathcal{N}$ $(a \neq x)$ *and* $M \in \mathcal{M}$ *a message such that* $x \notin M$.

*Let*

$$h \stackrel{\text{def}}{=} \{(a,a),(M,M)\}$$
$$v \stackrel{\text{def}}{=} \{(x,x)\}$$
$$\prec \stackrel{\text{def}}{=} \{((a,a),(x,x)),((M,M),(x,x))\}$$

*In other words we have* $(a,a)\prec(x,x)$ *and* $(M,M)\prec(x,x)$.

*Then* $\mathsf{se} = (h, v, \prec, (\emptyset,\emptyset))$ *is a S-environment.*

*Moreover, we have* $h^{\prec}_{(x,x)} = h$.

Since we build upon hedges, we cannot use the same substitution on both sides for representing the output actions of the attacker. We thus use a pair of substitutions.

**Definition 56** *Let* $h$ *be a hedge and* $(\sigma,\rho)$ *be a pair of substitutions. We define* $h(\sigma,\rho) \stackrel{\text{def}}{=} \{(M\sigma, N\rho) \mid (M,N) \in h\}$.

The notion of respectfulness for substitutions, here, is a bit delicate. We first show the details and explain the intuition behind afterwards.

**Definition 57 (respectful substitutions)** *Let* $(\sigma,\rho)$ *be a pair of substitutions,* $\mathsf{se} = (h, v, \prec, (\gamma_l, \gamma_r))$ *be a S-environment and* $B \subseteq \mathcal{N} \times \mathcal{N}$ *a consistent hedge. We say that* $(\sigma,\rho)$ respects $\mathsf{se}$ with $B$ *– written* $(\sigma,\rho) \triangleright_B \mathsf{se}$ *– if*

- $\mathrm{supp}(\sigma) \subseteq \pi_1(v)$ *and* $\mathrm{supp}(\rho) \subseteq \pi_2(v)$
- $\forall (x,y) \in v : x \in \mathrm{supp}(\sigma) \Leftrightarrow y \in \mathrm{supp}(\rho)$
- $\pi_1(B) \setminus \mathrm{n}(\mathrm{cosupp}(\sigma)) = \emptyset$
- $\pi_1(B) \cap \mathrm{n}(\pi_1(h \cup (v \setminus v_{(\sigma,\rho)}))) = \emptyset = \pi_2(B) \cap \mathrm{n}(\pi_2(h \cup (v \setminus v_{(\sigma,\rho)})))$
- $\forall (x,y) \in v_{(\sigma,\rho)} : (x\sigma, y\rho) \in \mathcal{S}(\mathcal{I}(h^{\prec}_{(x,y)}(\sigma,\rho) \cup B \cup (v \setminus v_{(\sigma,\rho)})))$ *where* $v_{(\sigma,\rho)} = v \cap (\mathrm{supp}(\sigma) \times \mathrm{supp}(\rho))$
- $\forall x \in \gamma_l : x\sigma \in \mathcal{N}$
- $\forall y \in \gamma_r : y\rho \in \mathcal{N}$

In this definition, we see that substitutions affect only names seen as input variables. Moreover, names may be replaced by messages that can be synthesised by the environment who, for this purpose, can also make use of fresh names mentioned in $B$. Finally, a name that was required to enable a transition—according to the transition subscript $S$—can only be replaced by another name, and not by a complex message.

**Example 58** *Consider the S-environment of Example 55. Let* $y, z \in \mathcal{N}$ $(y \neq z)$ *such that* $\{y,z\} \cap \mathrm{n}(M,a) = \emptyset$.

*Consider* $\sigma = \{x \mapsto \mathrm{E}_y(M)\}$ *and* $\rho = \{x \mapsto \mathrm{E}_z(M)\}$.

*We have* $\mathrm{supp}(\sigma) = \mathrm{supp}(\rho) = x = \pi_1(v) = \pi_2(v)$.

*We have* $v_{(\sigma,\rho)} = \{(x,x)\} = v$.

*Let* $B \stackrel{\mathrm{def}}{=} \{(y,z)\}$.

*We have* $\mathrm{n}(\mathrm{cosupp}(\sigma)) = \{y\} \cup \mathrm{n}(M)$ *and* $\mathrm{n}(\mathrm{cosupp}(\rho)) = \{z\} \cup \mathrm{n}(M)$.

*Thus* $\pi_1(B) \setminus \mathrm{n}(\mathrm{cosupp}(\sigma)) = \emptyset$.

*Moreover* $\pi_1(B) \cap \mathrm{n}(\pi_1(h \cup (v \setminus v_{(\sigma,\rho)}))) = \{y\} \cap \mathrm{n}(\pi_1(h)) = \{y\} \cap \mathrm{n}(M,a) = \emptyset$.

*Similarly* $\pi_2(B) \cap \mathrm{n}(\pi_2(h \cup (v \setminus v_{(\sigma,\rho)}))) = \emptyset$.

*We have* $h^{\prec}_{(x,x)}(\sigma,\rho) = h(\sigma,\rho) = h$ *because* $x \notin \mathrm{n}(h)$.

*So we have* $\mathcal{S}(\mathcal{I}(h^{\prec}_{(x,x)}(\sigma,\rho) \cup B \cup (v \setminus v_{(\sigma,\rho)}))) = \mathcal{S}(\mathcal{I}(h \cup B))$.

*Therefore* $(x\sigma, x\rho) \in \mathcal{S}(\mathcal{I}(h^{\prec}_{(x,x)}(\sigma,\rho) \cup B \cup (v \setminus v_{(\sigma,\rho)})))$ *(by applying* SYN-ENC*).*

*We have finally proved that* $(\sigma,\rho) \triangleright_B \mathsf{se}$.

If a pair of substitutions respects a S-environment, we can define the updating of this environment with respect to the considered pair of substitutions.

**Definition 59 (S-environment updating)** *Let* $(\sigma,\rho)$ *be a pair of substitutions,* $\mathsf{se} = (h, v, \prec, (\gamma_l, \gamma_r))$ *be a S-environment and* $B \subseteq \mathcal{N} \times \mathcal{N}$ *a consistent hedge such that* $(\sigma,\rho) \triangleright_B \mathsf{se}$*. The update* $\mathsf{se}^{(\sigma,\rho)}_B = (h', v', \prec', (\gamma'_l, \gamma'_r))$ *of* $\mathsf{se}$ *by* $(\sigma,\rho)$ *is defined as follows:*

- $h' = h(\sigma,\rho)$
- $v' = (v \setminus (\mathrm{supp}(\sigma) \times \mathrm{supp}(\rho))) \cup B$
- $\prec'$ *is defined by*

$$(M\sigma, N\rho) \prec' (x', y') \Leftrightarrow \bigwedge_{(x,y) \in v \,\wedge\, x' \in \mathrm{n}(x\sigma)} (M,N) \prec (x,y)$$

- $\gamma'_l = \gamma_l\sigma \cap \pi_1(v')$
- $\gamma'_r = \gamma_r\rho \cap \pi_2(v')$

**Example 60** *We continue Example 58. We have* $\mathsf{se}^{(\sigma,\rho)}_B = (h', v', \prec', (\emptyset, \emptyset))$

*with*

$$h' \stackrel{\text{def}}{=} h$$
$$v' \stackrel{\text{def}}{=} \{(y, z)\}$$
$$\prec' \stackrel{\text{def}}{=} \{((a, a), (y, z)), ((M, M), (y, z))\}$$

*In other words, we have* $(a, a) \prec' (y, z)$ *and* $(M, M) \prec' (y, z)$.

A S-environment is consistent if the knowledge contained in it does not lead to contradictions. This means that the attacker cannot distinguish between the two halves (each corresponding to a player in the bisimulation game) of the S-environment. Obviously, we make use of the notion of consistency for the underlying concept of hedges.

**Definition 61 (consistency)** *A S-environment* $\mathsf{se} = (h, v, \prec, (\gamma_l, \gamma_r))$ *is* consistent *if for all* $(\sigma, \rho)$, $B$ *such that* $(\sigma, \rho) \triangleright_B \mathsf{se}$, *we have:*

- $\mathcal{I}(h' \cup v')$ *is consistent*
- $\forall (x, y) \in v' : x \in \gamma'_l \Leftrightarrow y \in \gamma'_r$

*where* $(h', v', \prec', (\gamma'_l, \gamma'_r)) = \mathsf{se}_B^{(\sigma,\rho)}$.

In the above definition, we need to consider every respectful pair of substitutions. All these pairs correspond to what the attacker can derive, and none of them should lead to a contradiction.

Finally, we define three ways to add information to an environment: adding a pair of emitted messages, adding a pair of input names or adding some name constraints. The definition represents a straightforward adaptation of the principle that we used for the $\pi$-calculus.

**Definition 62 (extension)** *Let* $\mathsf{se} = (h, v, \prec, (\gamma_l, \gamma_r))$ *be a S-environment. If* $(M, N) \in \mathcal{M} \times \mathcal{M}$, *we define*

$$\mathsf{se} \oplus_{\mathrm{O}} (M, N) \stackrel{\text{def}}{=} (h', v, \prec, (\gamma_l, \gamma_r))$$

*where* $h' \stackrel{\text{def}}{=} \begin{cases} h \cup \{(M, N)\} & \text{if } (M, N) \notin v, \\ h & \text{otherwise.} \end{cases}$

*If* $(x, y) \in \mathcal{N} \times \mathcal{N}$ *with* $x \notin \mathrm{n}_1(\mathsf{se})$ *and* $y \notin \mathrm{n}_2(\mathsf{se})$, *we define*

$$\mathsf{se} \oplus_{\mathrm{V}} (x, y) \stackrel{\text{def}}{=} (h, v \cup \{(x, y)\}, \prec \cup\, h \times \{(x, y)\}, (\gamma_l, \gamma_r)).$$

*Finally, if* $S_1$ *and* $S_2$ *are two finite sets of names, we define*

$$\mathsf{se} \oplus_{\mathrm{C}} (S_1, S_2) \stackrel{\text{def}}{=} (h, v, \prec, (\gamma_l \cup (S_1 \cap \pi_1(v)), \gamma_r \cup (S_2 \cap \pi_2(v)))).$$

**Example 63** *We keep the notations of Example 55 and we consider* $\mathsf{se}_0 = (h, \emptyset, \emptyset, (\emptyset, \emptyset))$.

*Then* $\mathsf{se}_0 \oplus_{\mathrm{V}} (x, x) = \mathsf{se}$.

### *5.2 Open hedged bisimulation*

We first define the notion of open hedged relation.

**Definition 64** *An* open hedged relation $\mathcal{R}$ *is a subset of* $\mathcal{S}_{\mathrm{H}} \times \mathcal{P} \times \mathcal{P}$ *such that* $\forall (\mathsf{se}, P, Q) \in \mathcal{R} : \mathrm{fn}(P) \subseteq \mathrm{n}_1(\mathsf{se}) \wedge \mathrm{fn}(Q) \subseteq \mathrm{n}_2(\mathsf{se})$.

*It is called*

- consistent *if* $\forall (\mathsf{se}, P, Q) \in \mathcal{R} : \mathsf{se}$ *is consistent*
- symmetric *if* $\forall (\mathsf{se}, P, Q) : (\mathsf{se}, P, Q) \in \mathcal{R} \Leftrightarrow (\mathsf{se}^{-1}, Q, P) \in \mathcal{R}$

The definition of open hedged bisimulations now arises naturally:

**Definition 65 (open hedged bisimulation)**
*A symmetric consistent open hedged relation* $\mathcal{R}$ *is an* open hedged bisimulation *if for all* $(\mathsf{se}, P, Q) \in \mathcal{R}$*, for all* $(\sigma, \rho)$ *and* $B$ *such that* $(\sigma, \rho) \rhd_B \mathsf{se}$*, if* $P\sigma \xrightarrow{\mu_1}_{S_1} P'$ *with* $\mathrm{bn}(\mu_1) \cap \mathrm{n}_1(\mathsf{se}_B^{(\sigma,\rho)}) = \emptyset$ *and* $\mathrm{ch}(\mu_1) \in \pi_1(\mathcal{S}(\mathsf{se}_B^{(\sigma,\rho)}))$ *(if* $\mu_1 \neq \tau$*), there exists* $Q'$*,* $\mu_2$ *and* $S_2$ *such that* $Q\rho \xrightarrow{\mu_2}_{S_2} Q'$ *with* $\mathrm{bn}(\mu_2) \cap \mathrm{n}_2(\mathsf{se}_B^{(\sigma,\rho)}) = \emptyset$ *and*

- *if* $\mu_1 = \tau$ *then* $\mu_2 = \tau$ *and* $(\mathsf{se}_B^{(\sigma,\rho)} \oplus_{\mathrm{C}} (S_1, S_2), P', Q') \in \mathcal{R}$
- *if* $\mu_1 = a_1(x_1)$ *then* $\mu_2 = a_2(x_2)$ *where* $(a_1, a_2) \in \mathcal{S}(\mathsf{se}_B^{(\sigma,\rho)})$ *and* $(\mathsf{se}_B^{(\sigma,\rho)} \oplus_{\mathrm{V}} (x_1, x_2) \oplus_{\mathrm{C}} (S_1, S_2), P', Q') \in \mathcal{R}$
- *if* $\mu_1 = (\nu \tilde{c})\,\overline{a_1}\,M_1$ *then* $\mu_2 = (\nu \tilde{d})\,\overline{a_2}\,M_2$ *where* $(a_1, a_2) \in \mathcal{S}(\mathsf{se}_B^{(\sigma,\rho)})$ *and* $(\mathsf{se}_B^{(\sigma,\rho)} \oplus_{\mathrm{O}} (M_1, M_2) \oplus_{\mathrm{C}} (S_1, S_2), P', Q') \in \mathcal{R}$

In every case, we keep track of the name constraints ($S_1$ and $S_2$) that come from the transitions. In case of an input, we add the two input names to the input part of the environment. In case of an output, we add the emitted message to the output part of the environment. By requiring that every environment is consistent, we ensure that these emitted messages do not permit the attacker to distinguish between the two processes.

**Definition 66 (open hedged bisimilarity)** *Let* $P, Q \in \mathcal{P}$ *and* $\mathsf{se} \in \mathcal{S}_{\mathrm{H}}$ *such that* $\mathrm{fn}(P) \subseteq \mathrm{n}_1(\mathsf{se})$ *and* $\mathrm{fn}(Q) \subseteq \mathrm{n}_2(\mathsf{se})$*. We say that* $P$ *and* $Q$ *are open* $\mathsf{se}$-

*hedged bisimilar – written* $P \sim^{\mathsf{se}}_{\mathrm{OH}} Q$ *– if there exists an open hedged bisimulation* $\mathcal{R}$ *such that* $(\mathsf{se}, P, Q) \in \mathcal{R}$.

We finally report a soundness result that tells us that open hedged bisimulation is strictly stronger than its late hedged counterpart. Given the analogous situation in the $\pi$-calculus, this result gives us confidence in that we have gotten the definition right.

**Proposition 67** *Let* $P, Q \in \mathcal{P}$ *and* $\mathsf{se} \in \mathcal{S}_{\mathrm{H}}$ *such that* $\mathrm{fn}(P) \subseteq \mathrm{n}_1(\mathsf{se})$ *and* $\mathrm{fn}(Q) \subseteq \mathrm{n}_2(\mathsf{se})$. *Then, we have*

$$P \sim^{\mathsf{se}}_{\mathrm{OH}} Q \implies \left(\forall (\sigma, \rho), B : \ (\sigma, \rho) \triangleright_B \mathsf{se} \implies P\sigma \sim^{\mathrm{H}(\mathsf{se}^{(\sigma,\rho)}_B)}_{\mathrm{LH}} Q\rho\right)$$

**Proof.**

Let $\mathcal{R}$ be an open hedged bisimulation such that $(\mathsf{se}, P, Q) \in \mathcal{R}$.

We show that $\mathcal{R}' = \left\{(\mathrm{H}(\mathsf{se}^{(\sigma,\rho)}_B), P\sigma, Q\rho) \mid (\mathsf{se}, P, Q) \in \mathcal{R} \wedge (\sigma, \rho) \triangleright_B \mathsf{se}\right\}$ is a late hedged bisimulation.

First, since $\mathcal{R}$ is a symmetric consistent open hedged relation, $\mathcal{R}'$ is a symmetric consistent hedged relation.

Let $(h_0, P_0, Q_0) \in \mathcal{R}'$. By definition of $\mathcal{R}'$, there exists $\mathsf{se}, P, Q, \sigma, \rho$ and $B$ such that $h_0 = \mathrm{H}(\mathsf{se}^{(\sigma,\rho)}_B)$, $P_0 = P\sigma$, $Q_0 = Q\rho$, $(\mathsf{se}, P, Q) \in \mathcal{R}$ and $(\sigma, \rho) \triangleright_B \mathsf{se}$.

Assume now that $P_0 \xrightarrow{\mu_1} P'$ with $\mathrm{bn}(\mu_1) \cap \mathrm{n}(\pi_1(h_0)) = \emptyset$ and $\mathrm{ch}(\mu_1) \in \pi_1(h_0)$ (if $\mu_1 \neq \tau$). By definition, there exists $S_1$ such that $P\sigma \xrightarrow{\mu_1}_{S_1} P'$. We have $\mathrm{bn}(\mu_1) \cap \mathrm{n}(\pi_1(h_0)) = \emptyset = \mathrm{bn}(\mu_1) \cap \mathrm{n}_1(\mathsf{se}^{(\sigma,\rho)}_B)$ and if $\mu_1 \neq \tau$ we have $\mathrm{ch}(\mu_1) \in \pi_1(h_0)$, which is equivalent to $\mathrm{ch}(\mu_1) \in \mathcal{S}(\pi_1(h_0))$, so we have $\mathrm{ch}(\mu_1) \in \mathcal{S}(\pi_1(\mathsf{se}^{(\sigma,\rho)}_B))$.

Since $(\mathsf{se}, P, Q) \in \mathcal{R}$, $(\sigma, \rho) \triangleright_B \mathsf{se}$ and $\mathcal{R}$ is an open hedged bisimulation, there exists $Q'$, $\mu_2$ and $S_2$ such that $Q\rho \xrightarrow{\mu_2}_{S_2} Q'$ with $\mathrm{bn}(\mu_2) \cap \mathrm{n}_2(\mathsf{se}^{(\sigma,\rho)}_B) = \emptyset$ and

- if $\mu_1 = \tau$ then $\mu_2 = \tau$ and $(\mathsf{se}^{(\sigma,\rho)}_B \oplus_{\mathrm{C}} (S_1, S_2), P', Q') \in \mathcal{R}$
- if $\mu_1 = a_1(x_1)$ then $\mu_2 = a_2(x_2)$ where $(a_1, a_2) \in \mathcal{S}(\mathsf{se}^{(\sigma,\rho)}_B)$ and $(\mathsf{se}^{(\sigma,\rho)}_B \oplus_{\mathrm{V}} (x_1, x_2) \oplus_{\mathrm{C}} (S_1, S_2), P', Q') \in \mathcal{R}$
- if $\mu_1 = (\nu \tilde{c})\,\overline{a_1}\,M_1$ then $\mu_2 = (\nu \tilde{d})\,\overline{a_2}\,M_2$ where $(a_1, a_2) \in \mathcal{S}(\mathsf{se}^{(\sigma,\rho)}_B)$ and $(\mathsf{se}^{(\sigma,\rho)}_B \oplus_{\mathrm{O}} (M_1, M_2) \oplus_{\mathrm{C}} (S_1, S_2), P', Q') \in \mathcal{R}$

So, there exists $Q'$ and $\mu_2$ such that $Q_0 \xrightarrow{\mu_2} Q'$ and $\mathrm{bn}(\mu_2) \cap \mathrm{n}(\pi_2(h_0)) = \mathrm{bn}(\mu_2) \cap \mathrm{n}_2(\mathsf{se}^{(\sigma,\rho)}_B) = \emptyset$ and

- if $\mu_1 = \tau$ then $\mu_2 = \tau$.
  Moreover we have $(\mathsf{se}', P', Q') \in \mathcal{R}$ with $\mathsf{se}' = \mathsf{se}_B^{(\sigma,\rho)} \oplus_{\mathrm{C}} (S_1, S_2)$.
  We have clearly $(\mathrm{id}, \mathrm{id}) \triangleright_\emptyset \mathsf{se}'$ where id is the identity function.
  So, we have $(\mathrm{H}(\mathsf{se}_\emptyset'^{(\mathrm{id},\mathrm{id})}), P'\mathrm{id}, Q'\mathrm{id}) = (\mathrm{H}(\mathsf{se}'), P', Q') \in \mathcal{R}'$.
  And, by definition, it is clear also that $\mathrm{H}(\mathsf{se}') = \mathrm{H}(\mathsf{se}_B^{(\sigma,\rho)})$ thus we have $(h_0, P', Q') \in \mathcal{R}'$.
- if $\mu_1 = a_1(x_1)$ then $\mu_2 = a_2(x_2)$.
  Moreover we have $(\mathsf{se}', P', Q') \in \mathcal{R}$ with $\mathsf{se}' = \mathsf{se}_B^{(\sigma,\rho)} \oplus_{\mathrm{V}} (x_1, x_2) \oplus_{\mathrm{C}} (S_1, S_2)$.
  Let $B' \subseteq \mathcal{N} \times \mathcal{N}$ consistent, $M_1, M_2 \in \mathcal{M}$ such that
  $\cdot$ $\pi_1(B') \setminus \mathrm{n}(M_1) = \emptyset$
  $\cdot$ $\pi_1(B') \cap \mathrm{n}(\pi_1(h_0)) = \emptyset = \pi_2(B') \cap \mathrm{n}(\pi_2(h_0))$
  $\cdot$ $(M_1, M_2) \in \mathcal{S}(h_0 \cup B')$
  Let $\sigma' = \{^{M_1}/_{x_1}\}$ and $\rho' = \{^{M_2}/_{x_2}\}$. We have $(\sigma', \rho') \triangleright_{B'} \mathsf{se}'$ (in particular, note that $x_1 \notin S_1$ and $x_2 \notin S_2$ by definition).
  So we have $(\mathrm{H}(\mathsf{se}_{B'}'^{(\sigma',\rho')}), P'\sigma', Q'\rho') \in \mathcal{R}'$.
  And, by definition, we have $\mathrm{H}(\mathsf{se}_{B'}'^{(\sigma',\rho')}) = h_0 \cup B'$, thus we have $(h_0 \cup B', P'\{^{M_1}/_{x_1}\}, Q'\{^{M_2}/_{x_2}\}) \in \mathcal{R}'$.
- if $\mu_1 = (\nu\tilde{c})\,\overline{a_1}\,M_1$ then $\mu_2 = (\nu\tilde{c})\,\overline{a_2}\,M_2$.
  Moreover we have $(\mathsf{se}', P', Q') \in \mathcal{R}$ with $\mathsf{se}' = \mathsf{se}_B^{(\sigma,\rho)} \oplus_{\mathrm{O}} (M_1, M_2) \oplus_{\mathrm{C}} (S_1, S_2)$.
  We have clearly $(\mathrm{id}, \mathrm{id}) \triangleright_\emptyset \mathsf{se}'$ where id is the identity function.
  So, we have $(\mathrm{H}(\mathsf{se}_\emptyset'^{(\mathrm{id},\mathrm{id})}), P'\mathrm{id}, Q'\mathrm{id}) = (\mathrm{H}(\mathsf{se}'), P', Q') \in \mathcal{R}'$.
  And, by definition, we have $\mathrm{H}(\mathsf{se}') = \mathcal{I}(h_0 \cup \{(M_1, M_2)\})$, thus we have $(\mathcal{I}(h_0 \cup \{(M_1, M_2)\}), P', Q') \in \mathcal{R}'$.

Hence $\mathcal{R}'$ is a late hedged bisimulation. ∎

### *5.3 Example*

**Definition 68** *If $h$ is a hedge and $(x, y) \in \mathcal{N} \times \mathcal{N}$, we write $(x, y) : h$ for the relation $\mathcal{R} \stackrel{\mathrm{def}}{=} \{((M, N), (x, y)) \mid (M, N) \in h\}$.*

*Now, if $\{(x_1, y_1), \ldots, (x_n, y_n)\} \subset \mathcal{N} \times \mathcal{N}$, we write $(x_1, y_1), \cdots, (x_n, y_n) : h$ for $(x_1, y_1) : h \cup \cdots \cup (x_n, y_n) : h$.*

**Example 69** *Consider again the processes $P$ and $\underline{P}$ defined at Example 53.*

*We show that $P \sim_{\mathrm{OH}}^{\mathsf{se}_0} \underline{P}$ where $\mathsf{se}_0 = (\{(a, a), (M, M)\}, \emptyset, \emptyset, (\emptyset, \emptyset))$.*

*We (re)define some shortcuts:*

$$\begin{aligned} h_0 &\stackrel{\text{def}}{=} \{(a,a),(M,M)\} \\ h_1(k,l) &\stackrel{\text{def}}{=} h_0 \cup \{(\mathrm{E}_k(M),\mathrm{E}_l(M))\} \\ h_1 &\stackrel{\text{def}}{=} h_1(k,k) \end{aligned}$$

$$\begin{aligned} B_1(N,k) &\stackrel{\text{def}}{=} \overline{a}\langle \mathrm{D}_k(N)\rangle.\,\mathbf{0} \\ B_1 &\stackrel{\text{def}}{=} B_1(x,k) \\ \underline{B}_1(N,k) &\stackrel{\text{def}}{=} [\mathrm{D}_k(N){:}\mathcal{M}]\overline{a}\langle M\rangle.\,\mathbf{0} \\ \underline{B}_1 &\stackrel{\text{def}}{=} \underline{B}_1(x,k) \\ P_1 &\stackrel{\text{def}}{=} (\nu k)\,(A \mid B_1) \\ \underline{P}_1 &\stackrel{\text{def}}{=} (\nu k)\,(A \mid \underline{B}_1) \\ P_2 &\stackrel{\text{def}}{=} (\nu k)\,\overline{a}\langle \mathrm{D}_k(\mathrm{E}_k(M))\rangle.\,\mathbf{0} \\ \underline{P}_2 &\stackrel{\text{def}}{=} (\nu k)\ \ [\mathrm{D}_k(\mathrm{E}_k(M)){:}\mathcal{M}]\ \overline{a}\langle M\rangle.\,\mathbf{0} \end{aligned}$$

*Let* $\mathcal{R} = \{((h_0,\emptyset,\emptyset,(\emptyset,\emptyset)),P,\underline{P})\} \cup \mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3$ *where*

$$\mathcal{R}_1 = \{((h_0,\emptyset,\emptyset,(\emptyset,\emptyset)),P_2,\underline{P}_2)\,,((h_0,\emptyset,\emptyset,(\emptyset,\emptyset)),\mathbf{0},\mathbf{0})\}$$

$$\begin{aligned} \mathcal{R}_2 = \quad & \{((h_1,\emptyset,\emptyset,(\emptyset,\emptyset)),B,\underline{B}) \mid k \textit{ fresh}\} \\ \cup\ & \{((h_1,\{(x,x)\}\,,(x,x):h_1,(\emptyset,\emptyset)),B_1,\underline{B}_1) \mid k \textit{ and } x \textit{ fresh}\} \\ \cup\ & \{((h_1,\emptyset,\emptyset,(\emptyset,\emptyset)),\mathbf{0},\mathbf{0}) \mid k \textit{ fresh}\} \end{aligned}$$

*and*

$$\begin{aligned} \mathcal{R}_3 = \quad & \{((h_0,\{(x,x)\}\,,(x,x):h_0,(\emptyset,\emptyset)),P_1,\underline{P}_1) \mid x \textit{ fresh}\} \\ \cup\ & \left\{ \begin{array}{l} \left(\begin{array}{l} h_1(k,l), \\ \{(y_1,z_1),\ldots,(y_n,z_n)\}\,, \\ (y_1,z_1)\cdots(y_n,z_n):h_0, \\ (\emptyset,\emptyset) \end{array}\right), B_1(N_1,k),\underline{B}_1(N_2,l)) \\ \textit{where } k,y_1,\ldots,y_n \textit{ fresh on left} \\ \textit{and } l,z_1,\ldots,z_n \textit{ fresh on right} \\ \textit{and for all } i,\ y_i \in \mathrm{n}(N_1) \\ \textit{and } (N_1,N_2) \in \mathcal{S}(\mathcal{I}(h_0 \cup \{(y_1,z_1),\ldots,(y_n,z_n)\})) \end{array} \right\} \end{aligned}$$

*Then, the "symmetric" closure of $\mathcal{R}$ is an open hedged bisimulation.*

*The Figures 2 and 3 show the transition graphs of $P$ and $\underline{P}$ and illustrate the open hedged bisimulation induced by $\mathcal{R}$. Below the arrow is the substitution applied to the term. The notations are the same as for $\mathcal{R}$.*

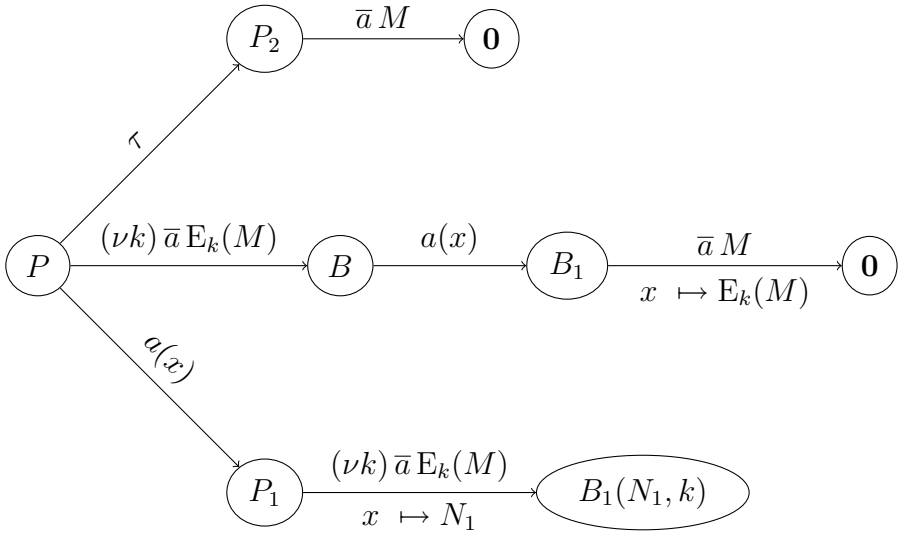


Fig. 2. Transition graph of $P$ (Example 69)

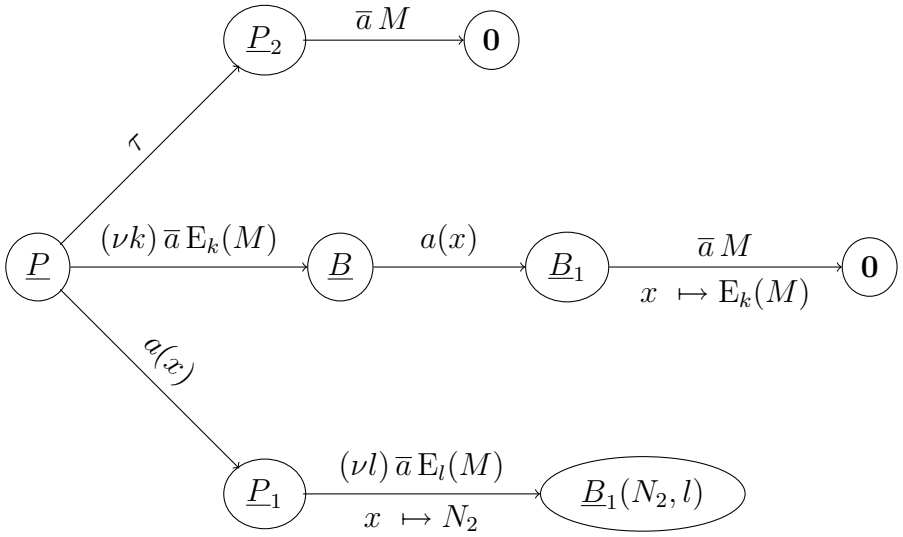


Fig. 3. Transition graph of $\underline{P}$ (Example 69)

### *5.4 Open hedged bisimulation is an extension of* K*-open bisimulation*

In this section, we compare open hedged bisimulation with K-open bisimulation by studying the effect of open hedged bisimulation on $\pi$-calculus processes. The main work to achieve is to relate S-environments and K-environments.

For this, we first define the subclass of *$\pi$-restricted* S-environments, roughly by restricting the hedge components to the domain of names and requiring consistency when considered as standard hedges. We then show that such S-environments are *induced* by K-environments. After this, we relate the set of respectful substitutions of a K-environment and the set of respectful substitution pairs of the induced S-environments. Finally, we state and show that open hedged bisimulation is a conservative extension of K-open bisimulation.

### *5.4.1 Restricting S-environments to $\pi$-calculus data*

The consistency condition of a hedge containing just names is simplified to:

**Lemma 70** *Let $h$ be a hedge such that $h \subset \mathcal{N} \times \mathcal{N}$. Then*

$$h \text{ is consistent} \iff (\forall (a,b),(a',b') \in h : a = a' \iff b = b')$$

**Proof.**

Trivial (by Definition 48). ∎

We now study S-environments where the hedge part contains just names.

**Lemma 71** *Let $\mathsf{se} = (h, v, \prec, (\gamma_l, \gamma_r))$ a S-environment such that $h \subset \mathcal{N} \times \mathcal{N}$ and assume that $h \cup v$ is consistent.*

*Then for all $\sigma, \rho, B$ such that $(\sigma, \rho) \triangleright_B \mathsf{se}$, we have $h(\sigma, \rho) = h$.*

**Proof.**

We first show that $\pi_1(h) \cap \pi_1(v) = \emptyset$.

By contradiction, assume that there is $x \in \pi_1(h) \cap \pi_1(v)$.

This means that there is $y$ and $y'$ such that $(x, y) \in h$ and $(x, y') \in v$. Since $h \cup v$ is consistent, we have $y = y'$.

Thus $(x, y) \in h \cap v$. But $h \cap v = \emptyset$ by hypothesis. Contradiction.

So $\pi_1(h) \cap \pi_1(v) = \emptyset$. Similarly, $\pi_2(h) \cap \pi_2(v) = \emptyset$.

Now, by definition, $h(\sigma, \rho) = \{(M\sigma, N\rho) \mid (M, N) \in h\}$.

Let $(M, N) \in h$.

By hypothesis, $M \in \mathcal{N}$. Moreover, we have shown that $M \notin \pi_1(v)$. So $M \notin \text{supp}(\sigma)$. Thus $M\sigma = M$.

Similarly, $N\rho = N$.

Thus $h(\sigma,\rho) = \{(M,N) \mid (M,N) \in h\} = h$. ∎

We now give a simple characterisation of consistency for S-environment where the hedge part just contains names.

**Lemma 72** *Let* $\mathsf{se} = (h, v, \prec, (\gamma_l, \gamma_r))$ *a S-environment such that* $h \subset \mathcal{N} \times \mathcal{N}$. *Then*

$$\mathsf{se} \text{ is consistent} \iff \begin{cases} h \cup v \text{ is consistent} \\ \forall (x,y) \in v : x \in \gamma_l \iff y \in \gamma_r \end{cases}$$

**Proof.**

$\Rightarrow$) Assume that $\mathsf{se}$ is consistent.
It is clear that $(\mathrm{id},\mathrm{id}) \triangleright_\emptyset \mathsf{se}$ and $\mathsf{se} = \mathsf{se}_\emptyset^{(\mathrm{id},\mathrm{id})}$.
Since $\mathsf{se}$ is consistent, we thus have $\mathcal{I}(h \cup v)$ is consistent.
But $\mathcal{I}(h \cup v) = h \cup v$ because $h \cup v \subset \mathcal{N} \times \mathcal{N}$. So $h \cup v$ is consistent.
And we also have (by Definition 61) $\forall (x,y) \in v : x \in \gamma_l \iff y \in \gamma_r$.

$\Leftarrow$) Assume now that $h \cup v$ is consistent and $\forall (x,y) \in v : x \in \gamma_l \iff y \in \gamma_r$.
We have to show that $\mathsf{se}$ is consistent.
Let $(\sigma,\rho)$ and $B$ such that $(\sigma,\rho) \triangleright_B \mathsf{se}$.
Let $(h', v', \prec', (\gamma_l', \gamma_r')) = \mathsf{se}_B^{(\sigma,\rho)}$.
By definition, we have

- $h' = h(\sigma,\rho)$,
- $v' = (v \setminus \mathrm{supp}(\sigma) \times \mathrm{supp}(\rho)) \cup B$,
- $\gamma_l' = \gamma_l\sigma \cap \pi_1(v')$ and
- $\gamma_r' = \gamma_r\rho \cap \pi_2(v')$.

According to Definition 61, we have to show that $\mathcal{I}(h' \cup v')$ is consistent and that $\forall (x,y) \in v' : x \in \gamma_l' \iff y \in \gamma_r'$.

But by Lemma 71, we have $h' = h$. Thus $\mathcal{I}(h' \cup v') = \mathcal{I}(h \cup v') = h \cup v'$ (because $h \cup v' \subset \mathcal{N} \times \mathcal{N}$).
We can then use Lemma 70 to show consistency of $h \cup v'$.
Let $(a,b),(a',b') \in h \cup v' = h \cup (v \setminus \mathrm{supp}(\sigma) \times \mathrm{supp}(\rho)) \cup B$.
Assume that $a = a'$. We want to show that $b = b'$.
There are four cases:

(1) $(a,b) \in h \cup (v \setminus \mathrm{supp}(\sigma) \times \mathrm{supp}(\rho))$ and $(a',b') \in h \cup (v \setminus \mathrm{supp}(\sigma) \times \mathrm{supp}(\rho))$.
In this case, $(a,b) \in h \cup v$ and $(a',b') \in h \cup v$. Since $h \cup v$ is consistent we have $b = b'$.

(2) $(a,b) \in B$ and $(a',b') \in B$.
In this case, since $B$ is consistent by hypothesis we have $b = b'$.

(3) $(a,b) \in h \cup (v \setminus \mathrm{supp}(\sigma) \times \mathrm{supp}(\rho))$ and $(a',b') \in B$.
This case is impossible because by hypothesis we have that $\pi_1(B) \cap \pi_1(h \cup (v \setminus \mathrm{supp}(\sigma) \times \mathrm{supp}(\rho))) = \emptyset$ and otherwise $a$ would be in this

empty intersection.

(4) $(a,b) \in B$ and $(a',b') \in h \cup (v \setminus \mathrm{supp}(\sigma) \times \mathrm{supp}(\rho))$.

As before, this case is impossible.

So $b = b'$.

Similarly, if $b = b'$ we show that $a = a'$.

Thus $h \cup v'$ is consistent.

Now let $(x', y') \in v'$.

Assume that $x' \in \gamma'_l = \gamma_l\sigma \cap \pi_1(v')$.

Thus there exists $x \in \gamma_l$ such that $x' = x\sigma$. But $\gamma_l \subset \pi_1(v)$.

So there exists $y$ such that $(x,y) \in v$.

Since $x \in \gamma_l$, by hypothesis we have $y \in \gamma_r$.

If $x \notin \mathrm{supp}(\sigma)$, then $y \notin \mathrm{supp}(\rho)$. In this case, we have $(x,y) \in v'$ and $x\sigma = x = x'$. Since $v'$ is consistent, we have $y' = y = y\rho$. Thus $y' \in \gamma'_r$.

Otherwise, if $x \in \mathrm{supp}(\sigma)$, then $y \in \mathrm{supp}(\rho)$. By definition, we then have $(x\sigma, y\rho) \in \mathcal{S}(\mathcal{I}(h^{\prec}_{(x,y)}(\sigma,\rho) \cup B \cup (v \setminus \mathrm{supp}(\sigma) \times \mathrm{supp}(\rho))))$.

Moreover, since $x\sigma \in \mathcal{N}$ and $y\rho \in \mathcal{N}$, this can be simplified to $(x\sigma, y\rho) \in h^{\prec}_{x,y} \cup B \cup (v \setminus \mathrm{supp}(\sigma) \times \mathrm{supp}(\rho)) = h^{\prec}_{x,y} \cup v'$.

So, we have $(x', y\rho) \in h \cup v'$ and $(x', y') \in h \cup v'$. Since $h \cup v'$ is consistent, we have $y' = y\rho$. Thus $y' \in \gamma'_r$.

We have finally shown that $\mathsf{se}$ is consistent.

■

We now define $\pi$-restricted S-environments.

**Definition 73** *Let $\mathsf{se} = (h, v, \prec, (\gamma_l, \gamma_r))$ be a S-environment. We say that $\mathsf{se}$ is* $\pi$-restricted *if $h \subset \mathcal{N} \times \mathcal{N}$, $h \cup v$ is consistent, and there exists $\Gamma \subset v$ such that $\gamma_l = \pi_1(\Gamma)$ and $\gamma_r = \pi_2(\Gamma)$.*

The next lemma states that a $\pi$-restricted S-environment is always consistent.

**Lemma 74** *Let $\mathsf{se}$ be a S-environment. Assume that $\mathsf{se}$ is $\pi$-restricted. Then $\mathsf{se}$ is consistent.*

**Proof.**

Trivial by Lemma 72. ■

The next lemma states that every instantiation of a $\pi$-restricted S-environment is still $\pi$-restricted.

**Lemma 75** *Let $\mathsf{se}$ be a S-environment and $\sigma, \rho, B$ such that $(\sigma,\rho) \triangleright_B \mathsf{se}$. Assume that $\mathsf{se}$ is $\pi$-restricted.*

*Then $\mathsf{se}^{(\sigma,\rho)}_B$ is $\pi$-restricted.*

**Proof.**

Let $(h', v', \prec', (\gamma'_l, \gamma'_r)) = \mathsf{se}_B^{(\sigma,\rho)}$.

By definition, we have

- $h' = h(\sigma, \rho)$,
- $v' = (v \setminus \mathrm{supp}(\sigma) \times \mathrm{supp}(\rho)) \cup B$,
- $\gamma'_l = \gamma_l\sigma \cap \pi_1(v')$ and
- $\gamma'_r = \gamma_r\rho \cap \pi_2(v')$.

We know by Lemma 71 that $h' = h(\sigma, \rho) = h$.

So $h' \subset \mathcal{N} \times \mathcal{N}$.

Since $\mathsf{se}$ is consistent (Lemma 74), we know that $\mathcal{I}(h \cup v')$ is consistent. But $\mathcal{I}(h \cup v') = h \cup v'$ so $h \cup v'$ is consistent.

Furthermore, we have $\forall (x, y) \in v' : x \in \gamma'_l \iff y \in \gamma'_r$. This implies that there exists $\Gamma' \subset v'$ such that $\gamma'_l = \pi_1(\Gamma')$ and $\gamma'_r = \pi_2(\Gamma')$.

■

### *5.4.2 Relating S-environments and* K*-environments*

We first define how to obtain a set of S-environments from a K-environment.

**Definition 76** *Let $\mathsf{pe} = (O, V, \prec)$ be a* K*-environment. Let $\alpha, \beta : O \cup V \to \mathcal{N}$ two injective functions.*

*We define*

$$h \stackrel{\text{def}}{=} \{(\alpha(x), \beta(x)) \mid x \in O\}$$
$$v \stackrel{\text{def}}{=} \{(\alpha(x), \beta(x)) \mid x \in V\}$$
$$\prec_2 \stackrel{\text{def}}{=} \{((\alpha(n), \beta(n)), (\alpha(x), \beta(x))) \mid n \prec x\}$$

*We define the sets of S-environments* induced by $\mathsf{pe}$, $\alpha$ and $\beta$ *as being*

$$\mathsf{pe}\langle\alpha, \beta\rangle = \{(h, v, \prec_2, (\pi_1(\Gamma), \pi_2(\Gamma))) \mid \Gamma \subset v\}$$

**Proof.**

We quickly check that every element of $\mathsf{pe}\langle\alpha, \beta\rangle$ is a S-environment.

Let $\Gamma \subset v$.

Clearly $h \subset \mathcal{M} \times \mathcal{M}$ and $v \subset \mathcal{N} \times \mathcal{N}$ are two finite sets. Moreover, it is clear that $\pi_1(\Gamma) \subset \pi_1(v)$, $\pi_2(\Gamma) \subset \pi_2(v)$ and $\prec_2 \subset h \times v$.

We now show that $h \cap v = \emptyset$.

By contradiction, assume that $(a, b) \in h \cap v$.

By definition, there exists $n \in O$ such that $a = \alpha(n)$ and $b = \beta(n)$.

Still by definition, there exists $x \in V$ such that $a = \alpha(x)$ and $b = \beta(x)$.

Since $\alpha$ is injective and $a = \alpha(n) = \alpha(x)$, we have $n = x$. So $n = x \in O \cap V = \emptyset$. Contradiction. Thus $h \cap v = \emptyset$.

Now, let $(a, b) \in h$ and $(y, z) \in v$. Assume that $(a, b) \prec_2 (y, z)$. We have to show that $y \notin \mathrm{n}(a)$ and $z \notin \mathrm{n}(b)$, i.e. that $y \neq a$ and $z \neq b$.

There exists $n \in O$ and $x \in V$ such that $(a, b) = (\alpha(n), \beta(n))$ and $(y, z) = (\alpha(x), \beta(x))$.

By contradiction, assume that $y = a$ or $z = b$. By symmetry, assume that $y = a$.

Then $\alpha(n) = \alpha(x)$. Since $\alpha$ is injective, $n = x$. So $n = x \in O \cap V = \emptyset$. Contradiction. So $y \neq a$ and $z \neq b$.

Therefore, every element of $\mathsf{pe}\langle\alpha, \beta\rangle$ is a S-environment. ■

The next lemma states that every S-environment induced from a K-environment is $\pi$-restricted.

**Lemma 77** *Let $\mathsf{pe} = (O, V, \prec)$ be a* K*-environment and $\alpha, \beta : O \cup V \to \mathcal{N}$ two injective functions.*

*Then every element of $\mathsf{pe}\langle\alpha, \beta\rangle$ is $\pi$-restricted.*

**Proof.**

With the same notations as in Definition 76, it is clear that $h \subset \mathcal{N} \times \mathcal{N}$ and by definition $\gamma_l = \pi_1(\Gamma)$, $\gamma_r = \pi_2(\Gamma)$ where $\Gamma \subset v$.

We thus just have to show that $h \cup v$ is consistent.

To achieve this goal, we use Lemma 70.

Let $(a, b)$ and $(a', b') \in h \cup v$.

Assume that $a = a'$.

Since $h \cap v = \emptyset$ (according to Definition 76), we have two cases:

(1) $(a, b)$ and $(a', b') \in h$
In this case, there exists $n, n' \in O$ such that $(a, b) = (\alpha(n), \beta(n))$ and $(a', b') = (\alpha(n'), \beta(n'))$.
By hypothesis, $\alpha(n) = \alpha(n')$.
Since $\alpha$ is injective, we have $n = n'$.
Thus $\beta(n) = \beta(n')$ and $b = b'$.
(2) $(a, b)$ and $(a', b') \in v$
Similarly, $b = b'$.

So if $a = a'$ then $b = b'$.

Now, assume that $b = b'$ and show that $a = a'$.

A similar reasoning as above gives this result.

So $h \cup v$ is consistent and finally, every element of $\mathsf{pe}\langle\alpha, \beta\rangle$ is $\pi$-restricted. ■

The next lemma says that every $\pi$-restricted S-environment is induced by a K-environment.

**Lemma 78** *Let $\mathsf{se} = (h, v, \prec_2, (\pi_1(\Gamma), \pi_2(\Gamma)))$ be a S-environment (where $\Gamma \subset v$) which is $\pi$-restricted. Let $O, V \subset \mathcal{N}$ such that $O \cap V = \emptyset$ and on one hand $h$ and $O$, on the other hand $v$ and $V$, are equipotent.*

*Then there exists $\alpha, \beta : O \cup V \to \mathcal{N}$ two injective functions and $\prec \subset O \times V$ such that $\mathsf{se} \in (O, V, \prec)\langle\alpha, \beta\rangle$.*

**Proof.**

Since $h$ and $O$ are equipotent, so are $\pi_1(h)$ and $O$. So there exists $\alpha_1 : O \to \pi_1(h)$ which is bijective.

Similarly, there exists $\alpha_2 : V \to \pi_1(v)$ which is bijective.

We define now $\alpha : O \cup V \to \mathcal{N}$ as follows:

$$\begin{aligned} \alpha : O \cup V &\to \mathcal{N} \\ x &\mapsto \begin{cases} \alpha_1(x) & \text{if } x \in O \\ \alpha_2(x) & \text{if } x \in V \end{cases} \end{aligned}$$

First note that $\alpha$ is well defined because $O \cap V = \emptyset$.

We now show that $\alpha$ is injective.

Let $x, y \in O \cup V$ such that $\alpha(x) = \alpha(y)$. We want to show that $x = y$.

If both $x, y \in O$ or both $x, y \in V$, this is trivial because $\alpha_1$ and $\alpha_2$ are injective.

Assume then (by symmetry) that $x \in O$ and $y \in V$.

We have $\alpha(x) = \alpha_1(x) \in \pi_1(h)$ and $\alpha(y) = \alpha_2(y) \in \pi_1(v)$.

There exists $a, b$ such that $(\alpha(x), a) \in h$ and $(\alpha(x), b) \in v$.

Since $\mathsf{se}$ is $\pi$-restricted, we have $h \cup v$ is consistent. So $a = b$.

Thus, we have $(\alpha(x), a) \in h \cap v = \emptyset$. This is a contradiction. So $x = y$ and $\alpha$ is injective.

Now define $\beta : O \cup V \to \mathcal{N}$ as follows:

$$\begin{aligned} \beta : O \cup V &\to \mathcal{N} \\ x &\mapsto b \quad \text{if } (\alpha(x), b) \in h \cup v \end{aligned}$$

First, $\beta$ is well defined because if $x \in O \cup V$, then $\alpha(x) \in \pi_1(h \cup v)$. So there exists $b$ such that $(\alpha(x), b) \in h \cup v$. Assume now that $b$ and $b'$ are two candidates (i.e. $(\alpha(x), b)$ and $(\alpha(x), b') \in h \cup v$). Then, by consistency of $h \cup v$, we have $b = b'$.

We show now that $\beta$ is injective.

Let $x, y \in O \cup V$ such that $\beta(x) = \beta(y)$.

By definition, we have $(\alpha(x), \beta(x)) \in h \cup v$ and $(\alpha(y), \beta(y) \in h \cup v$.

Since $h \cup v$ is consistent and $\beta(x) = \beta(y)$, we have $\alpha(x) = \alpha(y)$. Since $\alpha$ is injective, we have $x = y$. So $\beta$ is injective.

We now define

$$\prec \stackrel{\text{def}}{=} \{(n, x) \mid ((\alpha(n), \beta(n)), (\alpha(x), \beta(x))) \in \prec_2\}$$

Then, it is clear that $\mathsf{se} \in (O, V, \prec)\langle \alpha, \beta \rangle$. ■

### *5.4.3 Relating respectful substitutions in S-environments and* K*-environments*

In the next definition, we build a pair of substitutions for an induced S-environment from a substitution respecting the source K-environment.

**Definition 79** *Let* $\mathsf{pe} = (O, V, \prec)$ *be a* K*-environment,* $\alpha, \beta : O \cup V \to \mathcal{N}$ *two injective functions and* $\mathsf{se} = (h, v, \prec_2, (\pi_1(\Gamma), \pi_2(\Gamma))) \in \mathsf{pe}\langle\alpha, \beta\rangle$.

*Let* $\sigma$ *such that* $\sigma \blacktriangleright\!\blacktriangleright \mathsf{pe}$.

*Define* $F \stackrel{\text{def}}{=} \text{cosupp}(\sigma)\backslash(O \cup (V \backslash \text{supp}(\sigma)))$ *and* $v' \stackrel{\text{def}}{=} v\backslash\{(\alpha(x), \beta(x)) \mid x \in \text{supp}(\sigma)\}$.

*Let* $\alpha', \beta' : F \to \mathcal{N}$ *two injective functions such that* $\alpha'(F) \cap \text{n}(\pi_1(h \cup v')) = \beta'(F) \cap \text{n}(\pi_2(h \cup v')) = \emptyset$.

*Define*

$$\hat{\alpha} : O \cup (V \setminus \text{supp}(\sigma)) \cup F \to \mathcal{N}$$
$$x \mapsto \begin{cases} \alpha(x) & \text{if } x \in O \cup (V \setminus \text{supp}(\sigma)) \\ \alpha'(x) & \text{otherwise (if } x \in F\text{)} \end{cases}$$

*and*

$$\hat{\beta} : O \cup (V \setminus \text{supp}(\sigma)) \cup F \to \mathcal{N}$$
$$x \mapsto \begin{cases} \beta(x) & \text{if } x \in O \cup (V \setminus \text{supp}(\sigma)) \\ \beta'(x) & \text{otherwise (if } x \in F\text{)} \end{cases}$$

*Moreover assume that*

$$\forall x \in \text{supp}(\sigma) : \alpha(x) \neq \alpha'(x\sigma) \wedge \beta(x) \neq \beta'(x\sigma)$$

*and define* $B \stackrel{\text{def}}{=} \{(\alpha'(x), \beta'(x)) \mid x \in F\}$, $\rho_1, \rho_2 : \mathcal{N} \to \mathcal{N}$ *the two substitutions that coincides with the identity function except that if* $x \in \text{supp}(\sigma)$, $\rho_1(\alpha(x)) = \hat{\alpha}(x\sigma)$, $\rho_2(\beta(x)) = \hat{\beta}(x\sigma)$.

*We denote by* $\mathsf{S}(\mathsf{pe}, \langle\alpha, \beta\rangle, \sigma)$ *the set of all pairs* $((\rho_1, \rho_2), B)$ *for all* $\alpha', \beta'$ *that satisfy the previous conditions.*

Note that the set $\mathsf{S}(\mathsf{pe}, \langle\alpha, \beta\rangle, \sigma)$ is never empty because $\mathcal{N}$ is infinite.

The following theorem is a key result for showing that open hedged bisimulation is an extension of K-open bisimulation.

**Theorem 80** *With the notations of Definition 79:*

- $\hat{\alpha}$ *and* $\hat{\beta}$ *are injective;*
- $(\rho_1, \rho_2) \triangleright_B \mathsf{se}$*;*
- *and* $\mathsf{se}_B^{(\rho_1, \rho_2)} \in \mathsf{pe}^\sigma\langle(\hat{\alpha}, \hat{\beta})\rangle$.

**Proof.**

- We first quickly show that $\mathrm{n}(\pi_1(h \cup v')) = \alpha(O \cup (V \setminus \mathrm{supp}(\sigma)))$. Indeed,

$$\begin{aligned}\mathrm{n}(\pi_1(h \cup v')) &= \pi_1(h \cup v') \\ &= \pi_1(\{(\alpha(x), \beta(x)) \mid x \in O \cup (V \setminus \mathrm{supp}(\sigma))\}) \\ &= \{\alpha(x) \mid x \in O \cup (V \setminus \mathrm{supp}(\sigma))\} \\ &= \alpha(O \cup (V \setminus \mathrm{supp}(\sigma)))\end{aligned}$$

- We show that $\hat{\alpha}$ is injective.
  $\hat{\alpha}$ is well-defined because $(O \cup (V \setminus \mathrm{supp}(\sigma))) \cap F = \emptyset$ by definition.
  Now, let $x, y$ such that $\hat{\alpha}(x) = \hat{\alpha(y)}$. We want to show that $x = y$.
  If $x, y \in O \cup (V \setminus \mathrm{supp}(\sigma))$ or $x, y \in F$, then by injectivity of $\alpha$ and $\alpha'$, we have clearly $x = y$.
  Assume then (by symmetry) that $x \in O \cup (V \setminus \mathrm{supp}(\sigma))$ and $y \in F$.
  We have $\hat{\alpha}(y) = \alpha'(y) \in \alpha'(F)$. So, by hypothesis, $\alpha'(y) \notin \mathrm{n}(\pi_1(h \cup v'))$.
  But by definition, $\hat{\alpha}(x) = \alpha(x) \in \mathrm{n}(\pi_1(h \cup v'))$. So $\hat{\alpha}(x) = \hat{\alpha}(y)$ is impossible.
  Thus $\hat{\alpha}$ is injective.
- We now show that $(\rho_1, \rho_2) \triangleright_B \mathsf{se}$.
  · $B$ is consistent because $\alpha'$ and $\beta'$ are injective.
  · We show that $\mathrm{supp}(\rho_1) = \alpha(\mathrm{supp}(\sigma))$.
    By definition, it is clear that $\mathrm{supp}(\rho_1) \subset \alpha(\mathrm{supp}(\sigma))$.
    Let $y = \alpha(x) \in \alpha(\mathrm{supp}(\sigma))$ (with $x \in \mathrm{supp}(\sigma)$).
    Since $x \in \mathrm{supp}(\sigma)$, we have $x\sigma \neq x$.
    By definition, $\rho_1(y) = \rho_1(\alpha(x)) = \hat{\alpha}(x\sigma)$.
    By definition $\hat{\alpha}(x\sigma) \in \{\alpha(x\sigma), \alpha'(x\sigma)\}$.
    If $\hat{\alpha}(x\sigma) = \alpha(x\sigma)$, then since $\alpha$ is injective and $x\sigma \neq x$, we have $\hat{\alpha}(x\sigma) \neq \alpha(x)$, i.e. $\rho_1(y) \neq y$ and $y \in \mathrm{supp}(\rho_1)$.
    If $\hat{\alpha}(x\sigma) = \alpha'(x\sigma)$, then by hypothesis, $\alpha'(x\sigma) \neq \alpha(x)$ so $\rho_1(y) \neq y$ and $y \in \mathrm{supp}(\rho_1)$.
    In all cases, $y \in \mathrm{supp}(\rho_1)$, so $\alpha(\mathrm{supp}(\sigma)) = \mathrm{supp}(\rho_1)$.
    Similarly, $\mathrm{supp}(\rho_2) = \beta(\mathrm{supp}(\sigma))$.
    So, clearly, $\mathrm{supp}(\rho_1) \subset \pi_1(v)$ and $\mathrm{supp}(\rho_2) \subset \pi_2(v)$.
    Moreover, it is obvious that

$$\forall (x, y) \in v : x \in \mathrm{supp}(\rho_1) \iff y \in \mathrm{supp}(\rho_2)$$

  · By contradiction, let $y' \in \pi_1(B) \setminus \mathrm{n}(\mathrm{cosupp}(\rho_1))$.
    By definition, $y' = \alpha'(y)$ for some $y \in F$.
    Since $y \in F = \mathrm{cosupp}(\sigma) \setminus (O \cup (V \setminus \mathrm{supp}(\sigma)))$, there exists $x \in \mathrm{supp}(\sigma)$ such that $x\sigma = y$.
    So $y' = \alpha'(y) = \hat{\alpha}(y) = \hat{\alpha}(x\sigma) = \rho_1(\alpha(x))$.
    So $y' \in \mathrm{cosupp}(\rho_1)$ since $\alpha(x) \in \alpha(\mathrm{supp}(\sigma)) = \mathrm{supp}(\rho_1)$. This is a contradiction.
    So $\pi_1(B) \setminus \mathrm{n}(\mathrm{cosupp}(\rho_1)) = \emptyset$.

· By contradiction, let $y' \in \pi_1(B) \cap \mathrm{n}(\pi_1(h \cup (v \setminus v_{(\rho_1,\rho_2)})))$.
By definition, $y' \in \alpha'(F)$ so there is $y \in F$ such that $y' = \alpha'(y)$.
By hypothesis, $y' \notin \mathrm{n}(\pi_1(h \cup v')) = \alpha(O \cup (V \setminus \mathrm{supp}(\sigma)))$.
But since $\mathrm{supp}(\rho_1) = \alpha(\mathrm{supp}(\sigma))$, we have clearly $\mathrm{n}(h \cup (v \setminus v_{(\rho_1,\rho_2)})) = \alpha(O \cup V \setminus \mathrm{supp}(\sigma))$.
So, we get a contradiction and $\pi_1(B) \cap \mathrm{n}(\pi_1(h \cup (v \setminus v_{(\rho_1,\rho_2)}))) = \emptyset$.
Similarly, $\pi_2(B) \cap \mathrm{n}(\pi_2(h \cup (v \setminus v_{(\rho_1,\rho_2)}))) = \emptyset$.

· Let $(x', y') \in v_{(\rho_1,\rho_2)}$. We have $x' = \alpha(x)$ and $y' = \beta(x)$ for some $x \in \mathrm{supp}(\sigma)$.
$x'\rho_1 = \hat{\alpha}(x\sigma)$ and $y'\rho_2 = \hat{\beta}(x\sigma)$.
Since $\sigma \blacktriangleright\!\blacktriangleright \mathsf{pe}$, we have $x\sigma \in O \implies x\sigma \prec x$.
Since $x \in \mathrm{supp}(\sigma)$, we have $x\sigma \in \mathrm{cosupp}(\sigma) \subset O \cup (V \setminus \mathrm{supp}(\sigma)) \cup F$.
If $x\sigma \in O$, we have $x\sigma \prec x$. And by definition, $\hat{\alpha}(x\sigma) = \alpha(x\sigma)$ and $\hat{\beta}(x\sigma) = \beta(x\sigma)$.
By definition, we have $(\alpha(x\sigma), \beta(x\sigma)) \prec_2 (\alpha(x), \beta(x))$. So $(x'\rho_1, y'\rho_2) \in h^{\prec_2}_{(x',y')}(\rho_1, \rho_2)$.
Otherwise, if $x\sigma \in (V \setminus \mathrm{supp}(\sigma)) \cup F$, then clearly $(x'\rho_1, y'\rho_2) \in (v \setminus v_{(\rho_1,\rho_2)}) \cup B$.

· The last two conditions are trivially satisfied because $\rho_1(\mathcal{N}) \subset \mathcal{N}$ and $\rho_2(\mathcal{N}) \subset \mathcal{N}$.
So we have finally shown that $(\rho_1, \rho_2) \triangleright_B \mathsf{se}$.

• By definition, we have $\mathsf{se}^{(\rho_1,\rho_2)} = (h, (v \setminus \mathrm{supp}(\rho_1) \times \mathrm{supp}(\rho_2)) \cup B, \prec'_2, (\gamma'_l, \gamma'_r))$.
Since $\mathsf{se}$ is consistent, we know that there is $\Gamma' \subset (v \setminus \mathrm{supp}(\rho_1) \times \mathrm{supp}(\rho_2)) \cup B$ such that $\gamma'_l = \pi_1(\Gamma)$ and $\gamma'_r = \pi_2(\Gamma')$.
By definition, we also have that $\mathsf{pe}^{\sigma} = (O, (V \setminus \mathrm{supp}(\sigma)) \cup (\mathrm{cosupp}(\sigma) \setminus (O \cup (V \setminus \mathrm{supp}(\sigma))), \prec') = (O, (V \setminus \mathrm{supp}(\sigma)) \cup F, \prec')$
We know that $\mathrm{supp}(\rho_1) = \alpha(\mathrm{supp}(\sigma))$ and $\mathrm{supp}(\rho_2) = \beta(\mathrm{supp}(\sigma))$.
We thus have $v \setminus \mathrm{supp}(\rho_1) \times \mathrm{supp}(\rho_2) = \{(\alpha(x), \beta(x)) \mid x \in V \setminus \mathrm{supp}(\sigma)\}$ and since $\alpha$ and $\hat{\alpha}$ (resp. $\beta$ and $\hat{\beta}$) coincides on $O \cup (V \setminus \mathrm{supp}(\sigma))$, we clearly have that

$$
\begin{aligned}
h &= \left\{(\hat{\alpha}(x), \hat{\beta}(x)) \mid x \in O\right\} \\
(v \setminus \mathrm{supp}(\rho_1) \times \mathrm{supp}(\rho_2)) \cup B &= \left\{(\hat{\alpha}(x), \hat{\beta}(x)) \mid x \in V \setminus \mathrm{supp}(\sigma)\right\} \cup B \\
&= \left\{(\hat{\alpha}(x), \hat{\beta}(x)) \mid x \in (V \setminus \mathrm{supp}(\sigma)) \cup F\right\}
\end{aligned}
$$

By definition, we have

$$\begin{aligned}
(\hat{\alpha}(n),\hat{\beta}(n)) \prec_2' (\hat{\alpha}(x),\hat{\beta}(x)) &\iff \bigwedge_{(x,y)\in v \wedge \hat{\alpha}(x)\in \mathrm{n}(x\rho_1)} (\alpha(n),\beta(n)) \prec_2 (x,y) \\
&\iff \bigwedge_{x\in V \wedge \hat{\alpha}(x)\in \mathrm{n}(\alpha(x)\rho_1)} (\alpha(n),\beta(n)) \prec_2 (\alpha(x),\beta(x)) \\
&\iff \bigwedge_{x\in V \wedge \hat{\alpha}(x)\in \mathrm{n}(\rho_1(\alpha(x)))} n \prec x \\
&\iff \bigwedge_{x\in V \wedge x\in \mathrm{n}(x\sigma)} n \prec x \qquad \text{by case distinction} \\
&\iff n \prec' x
\end{aligned}$$

So we conclude that $\mathsf{se}_B^{(\rho_1,\rho_2)} \in \mathsf{pe}^\sigma\langle(\hat{\alpha},\hat{\beta})\rangle$.

■

The next lemma is somehow the converse result of the previous theorem (Theorem 80).

**Lemma 81** *Let $\mathsf{se} = \mathsf{pe}\langle\alpha,\beta\rangle$ where $\mathsf{pe} = (O,V,\prec)$ is a* K*-environment. Let $\rho_1,\rho_2 : \mathcal{N} \to \mathcal{N}$ and $B$ such that $(\rho_1,\rho_2) \triangleright_B \mathsf{se}$.*

*Then there exists $\sigma$ such that $\sigma \blacktriangleright\!\blacktriangleright \mathsf{pe}$ and $((\rho_1,\rho_2),B) \in \mathsf{S}(\mathsf{pe},\langle\alpha,\beta\rangle,\sigma)$.*

**Proof.**

Let $(h,v,\prec_2,(\pi_1(\Gamma),\pi_2(\Gamma))) = \mathsf{se}$.

Let $F \subset \mathcal{N}$ such that $F$ is equipotent to $B$ and $F \cap (O \cup V) = \emptyset$ ($F$ exists because $\mathcal{N}$ is infinite).

So, there exists a bijection $f : F \to B$. We define $\alpha' = \pi_1(f)$ (the first projection of $f$) and $\beta' = \pi_2(f)$ (the second projection of $f$).

By definition, we have $B = \{(\alpha'(x),\beta'(x)) \mid x \in F\}$.

We define

$$\begin{aligned}
\alpha'' : O \cup V \cup F &\to \mathcal{N} \\
x &\mapsto \begin{cases} \alpha(x) & \text{if } x \in O \cup V \\ \alpha'(x) & \text{otherwise (if } x \in F) \end{cases}
\end{aligned}$$

Clearly, $\alpha''$ is well defined and injective. So it realises a bijection from $O \cup V \cup F$ to $\alpha''(O \cup V \cup F)$.

It is clear that $\rho_1(O \cup V) \subset \alpha''(O \cup V \cup F)$ because $\rho_1 : \mathcal{N} \to \mathcal{N}$ and we have for every $(x', y') \in v_{(\rho_1,\rho_2)} : (x'\rho_1, y'\rho_2)\mathcal{S}(\mathcal{I}(h^{\prec 2}_{(x',y')} \cup (v \setminus v_{(\rho_1,\rho_2)}) \cup B))$ which is equivalent to say that $(x'\rho_1, y'\rho_2) \in h^{\prec 2}_{(x',y')} \cup (v \setminus v_{(\rho_1,\rho_2)}) \cup B$.

We now define

$$\sigma : \mathcal{N} \to \mathcal{N}$$
$$x \mapsto \begin{cases} \alpha''^{-1}(\alpha(x)\rho_1) & \text{if } x \in V \text{ and } \alpha(x) \in \mathrm{supp}(\rho_1) \\ x & \text{otherwise} \end{cases}$$

We show that $\sigma \blacktriangleright\!\blacktriangleright \mathsf{pe}$.

We clearly have that $\mathrm{supp}(\sigma) \subset V$.

Let $x \in V$ and assume that $x\sigma \in O$.

Necessarily, $x \in \mathrm{supp}(\sigma)$ so $x\sigma = \alpha''^{-1}(\alpha(x)\rho_1)$ with $\alpha(x) \in \mathrm{supp}(\rho_1)$.

Since $x\sigma \in O$, necessarily, $\alpha(x)\rho_1 \in \pi_1(h)$. We thus have $(\alpha(x)\rho_1, \beta(x)\rho_2) \in h^{\prec 2}_{(\alpha(x),\beta(x))}$.

There exists $n \in O$ such that $\alpha(x)\rho_1 = \alpha(n)$ and $\beta(x)\rho_2 = \beta(n)$ and $n \prec x$. Obviously $n = x\sigma$.

So $\sigma \blacktriangleright\!\blacktriangleright \mathsf{pe}$.

Moreover, note that the definition of $\sigma$ is equivalent to

$$\sigma : \mathcal{N} \to \mathcal{N}$$
$$x \mapsto \begin{cases} \beta''^{-1}(\beta(x)\rho_2) & \text{if } x \in V \text{ and } \beta(x) \in \mathrm{supp}(\rho_2) \\ x & \text{otherwise} \end{cases}$$

because, in particular, $\forall (x, y) \in v : x \in \mathrm{supp}(\rho_1) \iff y \in \mathrm{supp}(\rho_2)$.

It is now easy to check that $F, \alpha', \beta'$ and $\sigma$ satisfy the condition of Theorem 80.

Thus $((\rho_1, \rho_2), B) \in \mathsf{S}(\mathsf{pe}, \langle \alpha, \beta \rangle, \sigma)$. ∎

### *5.4.4 Conservative extension result*

The following theorem states that the projection of open hedged bisimulation down to the $\pi$-calculus gives K-open bisimulation.

**Theorem 82** *Let $P, Q \in \mathcal{P}$ two $\pi$-calculus processes and $\mathsf{se}$ a S-environment which is $\pi$-restricted. Assume that $P \sim^{\mathsf{se}}_{\mathrm{OH}} Q$.*

*Then for every $\alpha, \beta, \mathsf{pe}$ such that $\mathsf{se} \in \mathsf{pe}\langle\alpha, \beta\rangle$, we have $P\alpha^{-1} \sim^{\mathsf{pe}}_{\mathrm{K}} Q\beta^{-1}$.*

**Proof.**

The proof uses the previous results.

Note in particular that since the free names of $P$ and $Q$ are included in $\mathsf{se}$, $\alpha^{-1}$ and $\beta^{-1}$ are well-defined on these sets.

If $\mathcal{R}$ is an open hedged bisimulation with $(\mathsf{se}, P, Q) \in \mathcal{R}$, we show that

$$\mathcal{R}' = \left\{ (\mathsf{pe}, P\alpha^{-1}, Q\beta^{-1}) \mid \begin{array}{l} (\mathsf{se}, P, Q) \in \mathcal{R} \quad \wedge \quad \mathsf{se} \text{ is } \pi\text{-restricted} \\ \mathsf{se} \in \mathsf{pe}\langle\alpha, \beta\rangle \end{array} \right\}$$

is a K-open bisimulation.

Theorem 80 (and the way the substitutions are built) is the key argument to mimic the transitions. ■

Concerning the converse of Theorem 82, we will only offer a conjecture. Its validity depends on another conjecture. The idea behind the proof of the latter is that when observing a $\pi$-calculus process within a spi-calculus context, it is sufficient to check for substitutions that do not involve compound messages but just names, because $\pi$-calculus processes do not possess any means to look inside compound messages anyway. A similar idea was developed by Hüttel in [23], where the notion of $d$-framed bisimilarity was introduced to prove decidability; the parameter $d$ indicates the maximal depth of the messages involved in a framed bisimulation. Hüttel also showed that for any triple $(\mathrm{fr}, P, Q)$ there is a critical depth $d$ above which framed bisimilarity and $d$-framed bisimilarity coincide. For $\pi$-calculus terms, the critical depth is 0.

We strongly believe that Hüttel's result can be adapted to the context of open hedged bisimilarity. Since a formal proof would be quite lengthy, for sake of brevity, we state the corresponding result just as a conjecture and we just give the simplified definition for the case $d = 0$ that we call 0-open hedged bisimulation. The difference with open hedged bisimulation is that the substitutions considered have their co-support included in $\mathcal{N}$.

**Definition 83 (0-open hedged bisimulation)**
*A symmetric consistent open hedged relation $\mathcal{R}$ is a* 0-open hedged bisimulation *if for all $(\mathsf{se}, P, Q) \in \mathcal{R}$, for all $\sigma, \rho : \mathcal{N} \to \mathcal{N}$ and $B$ such that $(\sigma, \rho) \triangleright_B \mathsf{se}$, if $P\sigma \xrightarrow{\mu_1}_{S_1} P'$ with $\mathrm{bn}(\mu_1) \cap \mathrm{n}_1(\mathsf{se}_B^{(\sigma,\rho)}) = \emptyset$ and $\mathrm{ch}(\mu_1) \in \pi_1(\mathcal{S}(\mathsf{se}_B^{(\sigma,\rho)}))$ (if $\mu_1 \neq$*

*$\tau$), there exists $Q'$, $\mu_2$ and $S_2$ such that $Q\rho \xrightarrow{\mu_2}_{S_2} Q'$ with $\mathrm{bn}(\mu_2) \cap \mathrm{n}_2(\mathsf{se}_B^{(\sigma,\rho)}) = \emptyset$ and*

- *if $\mu_1 = \tau$ then $\mu_2 = \tau$ and $(\mathsf{se}_B^{(\sigma,\rho)} \oplus_{\mathrm{C}} (S_1, S_2), P', Q') \in \mathcal{R}$*
- *if $\mu_1 = a_1(x_1)$ then $\mu_2 = a_2(x_2)$ where $(a_1, a_2) \in \mathcal{S}(\mathsf{se}_B^{(\sigma,\rho)})$ and $(\mathsf{se}_B^{(\sigma,\rho)} \oplus_{\mathrm{V}} (x_1, x_2) \oplus_{\mathrm{C}} (S_1, S_2), P', Q') \in \mathcal{R}$*
- *if $\mu_1 = (\nu \tilde{c})\,\overline{a_1}\, M_1$ then $\mu_2 = (\nu \tilde{d})\,\overline{a_2}\, M_2$ where $(a_1, a_2) \in \mathcal{S}(\mathsf{se}_B^{(\sigma,\rho)})$ and $(\mathsf{se}_B^{(\sigma,\rho)} \oplus_{\mathrm{O}} (M_1, M_2) \oplus_{\mathrm{C}} (S_1, S_2), P', Q') \in \mathcal{R}$*

The key conjecture we rely on for proving Conjecture 85 is the following.

**Conjecture 84** *Let $\mathsf{se} = (h, v, \prec, (\gamma_l, \gamma_r))$ be a S-environment, $P, Q \in \mathcal{P}$ two $\pi$-calculusprocesses such that $h \subset \mathcal{N} \times \mathcal{N}$.*

*Then*

$$P \sim_{\mathrm{OH}}^{\mathsf{se}} Q \iff \exists \mathcal{R} : \mathcal{R} \text{ is a 0-open hedged bisimulation} \wedge (\mathsf{se}, P, Q) \in \mathcal{R}$$

Finally, we state the conjecture that open hedged bisimulation is a complete extension of K-open bisimulation.

**Conjecture 85** *Let $P, Q \in \mathcal{P}$ two $\pi$-calculus processes and $\mathsf{pe} = (O, V, \prec)$ a* K*-environment. Assume that $P \sim_{\mathrm{K}}^{\mathsf{pe}} Q$.*

*Then for every $\alpha, \beta : O \cup V \to \mathcal{N}$ injective, for every $\mathsf{se} \in \mathsf{pe}\langle \alpha, \beta \rangle$, we have $P\alpha \sim_{\mathrm{OH}}^{\mathsf{se}} Q\beta$.*

**Proof.**

Here, we use Conjecture 84 and the previous results.

If $\mathcal{R}$ is an K-open bisimulation such that $(\mathsf{pe}, P, Q) \in \mathcal{R}$, then we show that

$$\mathcal{R}' = \left\{ (\mathsf{se}, P\alpha, Q\beta) \;\middle|\; \begin{array}{c} (\mathsf{pe}, P, Q) \in \mathcal{R} \;\wedge\; (O, V, \prec) = \mathsf{pe} \\ \alpha, \beta : O \cup V \to \mathcal{N} \text{ injective} \\ \mathsf{se} \in \mathsf{pe}\langle \alpha, \beta \rangle \end{array} \right\}$$

is a 0-open hedged bisimulation.

Lemma 81 (and the way the substitutions are built) is essential to mimic the transitions. ∎

## 6 Conclusion and future work

We have achieved our goal to find an open-style definition of bisimulation in the spi-calculus by studying a carefully crafted knowledge-aware variant of open bisimulation in the $\pi$-calculus. Without knowledge-awareness, the desired lifting open open bisimulation would not have been possible. As the list of individual contributions in the Introduction shows, we have proved formal properties of this definition that witness its usefulness and thus provide a formal justification.

Quite unexpectedly for us, the investigation of the K-open variant of bisimulation itself provided us with a deeper understanding of openness. Apart from this more philosophical interpretation, the improvement on congruence properties was a welcome and equally unexpected side product. Once observed, it may appear less surprising: the refinement builds upon the characterization of contexts that just exploit the additional information of K-environments.

On the $\pi$-calculus side, it is now interesting to study the precise link between the K-open bisimilarity defined in this paper and the open bisimilarity variant defined by Tiu, Miller, Ziegler and Palamidessi in [24–26]. There, two different quantifiers are used to introduce names: $\forall$ for input variables and $\nabla$ for fresh names. We are confident that the two bisimilarities are tightly related.

On the spi-calculus side, we want to understand the precise link between open hedged bisimulation and symbolic bisimulation, as proposed in [15]. We conjecture that open hedged bisimulation is (close to be) the "concrete" version of this symbolic bisimulation. Such a result would compensate for the weakness of open hedged bisimulation of itself not being directly implementable.

Another interesting question is, in how far the refined congruence properties of K-open bisimulation carry over from the $\pi$-calculus to the spi-calculus. However, there the situation is quite more complicated. As noted by Boreale and Gorla in [27], a major difficulty for congruence properties in the spi-calculus is the case of parallel composition, where a naive formulation is simply wrong. We could likely reuse a number of ideas of [27] for studying the congruence properties of open hedged bisimulation. However, it is yet unclear to us whether the distinction between input variables and freshly created names will equally help us to formulate more refined congruence properties.

## References


[1] D. Sangiorgi, A theory of bisimulation for the $\pi$-calculus, Acta Informatica 33 (1996) 69–97.

[2] R. Milner, J. Parrow, D. Walker, A calculus of mobile processes, part I/II, Information and Computation 100 (1992) 1–77.

[3] J. Parrow, An introduction to the $\pi$-calculus, in: J. Bergstra, A. Ponse, S. Smolka (Eds.), Handbook of Process Algebra, Elsevier B.V., 2001, pp. 479–543.

[4] D. Sangiorgi, D. Walker, The $\pi$-calculus: a Theory of Mobile Processes, Cambridge University Press, 2001.

[5] B. Victor, A verification tool for the polyadic $\pi$-calculus, Licentiate thesis, Department of Computer Systems, Uppsala University, Sweden, available as report DoCS 94/50 (May 1994).
URL `http://www.docs.uu.se/~victor/tr/docs-tr-94-50.html`

[6] S. Briais, ABC Bisimulation Checker, EPFL (2003).
URL `http://lamp.epfl.ch/~sbriais/abc/abc.html`

[7] M. Abadi, A. D. Gordon, A calculus for cryptographic protocols: The Spi calculus, Information and Computation 148 (1) (1999) 1–70.

[8] H. Barendregt, The Lambda Calculus: Its Syntax and Semantics, Vol. 103 of Studies in Logic and the Foundations of Mathematics, North-Holland, 1984.

[9] K. Honda, N. Yoshida, On reduction-based process semantics, Theoretical Computer Science 152 (2) (1995) 437–486.

[10] M. Hennessy, J. Rathke, Typed behavioural equivalences for processes in the presence of subtyping., Mathematical Structures in Computer Science 14 (5) (2004) 651–684.

[11] M. Hennessy, H. Lin, Symbolic bisimulations, Theoretical Computer Science 138 (2) (1995) 353–389.

[12] H. Lin, Symbolic bisimulation and proof systems for the $\pi$-calculus, Tech. Rep. 7/94, University of Sussex, Brighton (1994).

[13] M. Boreale, R. De Nicola, A symbolic semantics for the $\pi$-calculus, Information and Computation 126 (1) (1996) 34–52.

[14] J. Borgström, U. Nestmann, On bisimulations for the spi calculus, Mathematical Structures in Computer Science 15 (2005) 487–552.

[15] J. Borgström, S. Briais, U. Nestmann, Symbolic bisimulation in the spi calculus, in: P. Gardner, N. Yoshida (Eds.), Proceedings of CONCUR 2004: Concurrency Theory, Vol. 3170 of LNCS, Springer, 2004, pp. 161–176.

[16] S. Briais, U. Nestmann, Open bisimulation, revisited, in: J. Baeten, I. Phillips (Eds.), Proceedings of EXPRESS 2005: Expressiveness in Concurrency, Vol. 154 of ENTCS, Elsevier B.V., 2005, pp. 93–105.

[17] D. Sangiorgi, D. Walker, On barbed equivalences in $\pi$-calculus, in: Proceedings of CONCUR '01: Concurrency Theory, Vol. 2154 of LNCS, Springer, 2001, pp. 292–304.

[18] Y. Fu, On quasi-open bisimulation, Theoretical Computer Science 338 (2005) 96–126.

[19] S. Briais, Formal proofs about hedges using the Coq proof assistant (2004).
URL `http://lamp.epfl.ch/~sbriais/spi/hedges/hedge.html`

[20] M. Abadi, A. D. Gordon, A bisimulation method for cryptographic protocols, Nordic Journal of Computing 5 (4) (1998) 267–303.
URL
`http://www.cs.helsinki.fi/njc/References/abadig1998:267%.html`

[21] S. Briais, U. Nestmann, A formal semantics for protocol narrations, in: R. de Nicola, D. Sangiorgi (Eds.), Trustworthy Global Computing, Vol. 3705 of LNCS, Springer, 2005, pp. 163–181.

[22] S. Briais, Towards open bisimulation in the spi calculus, Mémoire de D.E.A., Université Paris VII - Denis Diderot (2002).
URL `http://lamp.epfl.ch/~sbriais/ENS/ps/rapport-DEA.ps.gz`

[23] H. Hüttel, Deciding framed bisimilarity, in: A. Kučera, R. Mayr (Eds.), Proceedings of INFINITY 2002, Vol. 68 of ENTCS, Elsevier B.V., 2002, p. 20.

[24] A. Tiu, D. Miller, A proof search specification of the $\pi$-calculus, in: 3rd Workshop on the Foundations of Global Ubiquitous Computing, Vol. 138 of ENTCS, 2004, pp. 79–101.

[25] A. Tiu, Model checking for $\pi$-calculus using proof search, in: M. Abadi, L. de Alfaro (Eds.), Proceedings of CONCUR 2005: Concurrency Theory, Vol. 3653 of LNCS, Springer, 2005, pp. 36–50.

[26] A. Ziegler, D. Miller, C. Palamidessi, A congruence format for name-passing calculi, in: Proceedings of SOS 2005: Structural Operational Semantics, Vol. 156 of ENTCS, Elsevier B.V., Lisbon, Portugal, 2005, pp. 169–189.

[27] M. Boreale, D. Gorla, On compositional reasoning in the spi-calculus, in: M. Nielsen, U. H. Engberg (Eds.), Proceedings of FoSSaCS 2002, Vol. 2303 of LNCS, Springer, 2002, pp. 67–81.