# Sound Bisimulations for Higher-Order Distributed Process Calculus*

Adrien Piérard and Eijiro Sumii**

Tohoku University
{adrien,sumii}@kb.ecei.tohoku.ac.jp

**Abstract.** While distributed systems with transfer of processes have become pervasive, methods for reasoning about their behaviour are underdeveloped. In this paper we propose a bisimulation technique for proving behavioural equivalence of such systems modelled in the *higher-order $\pi$-calculus with passivation* (and restriction). Previous research for this calculus is limited to context bisimulations and normal bisimulations which are either impractical or unsound. In contrast, we provide a sound and useful definition of *environmental bisimulations*, with several non-trivial examples. Technically, a central point in our bisimulations is the clause for parallel composition, which must account for passivation of the spawned processes in the middle of their execution.

## 1 Introduction

### 1.1 Background

Higher-order distributed systems are ubiquitous in today's computing environment. To name but a few examples, companies like Dell and Hewlett-Packard sell products using virtual machine live migration [14,3], and Gmail users execute remote JavaScript code on local browsers. In this paper we call *higher-order* the ability to transfer processes, and *distribution* the possibility of location-dependent system behaviour. In spite of the *de facto* importance of such systems, they are hard to analyse because of their inherent complexity.

The $\pi$-calculus [8] and its dialects prevail as models of concurrency, and several variations of these calculi have been designed for distribution. First-order variations include the ambient calculus [1] and D$\pi$ [2], while higher-order include more recent Homer [4] and Kell [15] calculi. In this paper, we focus on the higher-order $\pi$-calculus with *passivation* [7], a simple high-level construct to express distribution. It is an extension of the higher-order $\pi$-calculus [9] (with which the reader is assumed to be familiar) with *located processes* $a[P]$ and two additional transition rules: $a[P] \xrightarrow{\overline{a}\langle P\rangle} 0$ (PASSIV), and $a[P] \xrightarrow{\alpha} a[P']$ if $P \xrightarrow{\alpha} P'$ (TRANSP).

* Appendix with full proofs at http://www.kb.ecei.tohoku.ac.jp/~adrien/pubs/SoundAppendix.pdf

** This research is partially supported by KAKENHI 22300005, the Nakajima Foundation, and the Casio Science Promotion Foundation. The first author is partially supported by the Global COE Program CERIES.

M. Hofmann (Ed.): FOSSACS 2011, LNCS 6604, pp. 123–137, 2011.

The new syntax $a[P]$ reads as "process $P$ located at $a$" where $a$ is a name. Rule TRANSP specifies the transparency of locations, i.e. that a location has no impact on the transitions of the located process. Rule PASSIV indicates that a located process can be *passivated*, that is, be output to a channel of the same name as the location. Using passivation, various characteristics of distributed systems are expressible. For instance, failure of process $P$ located at $a$ can be modelled like $a[P] \mid a(X).\overline{fail} \to 0 \mid \overline{fail}$, and migration of process $Q$ from location $b$ to $c$ like $b[P] \mid b(X).c[X] \to 0 \mid c[P]$.

One way to analyse the behaviour of systems is to compare implementations and specifications. Such comparison calls for satisfying notions of behavioural equivalence, such as *reduction-closed barbed equivalence* (and *congruence*) [5], written $\approx$ (and $\approx_c$ respectively) in this paper.

Unfortunately, these equivalences have succinct definitions that are not very practical as a proof technique, for they both include a condition that quantifies over arbitrary processes, like: if $P \approx Q$ then $\forall R.\ P \mid R \approx Q \mid R$. Therefore, more convenient definitions like *bisimulations*, for which membership implies behavioural equivalence, and which come with a co-inductive proof method, are sought after.

Still, the combination of both higher order and distribution has long been considered difficult. Recent research on higher-order process calculi led to defining sound *context bisimulations* [10] (often at the cost of appealing to Howe's method [6] for proving congruence) but those bisimulations suffer from their heavy use of universal quantification: suppose that $\nu\widetilde{c}.\overline{a}\langle M\rangle.P\ \mathcal{X}\ \nu\widetilde{d}.\overline{a}\langle N\rangle.Q$, where $\mathcal{X}$ is a context bisimulation; then it is roughly required that for any process $R$, we have $\nu\widetilde{c}.(P \mid R\{M/X\})\ \mathcal{X}\ \nu\widetilde{d}.(Q \mid R\{N/X\})$. Not only must we consider the outputs $M$ and $N$, but we must also handle interactions of arbitrary $R$ with the continuation processes $P$ and $Q$. Alas, this almost comes down to showing reduction-closed barbed equivalence! In the higher-order $\pi$-calculus, by means of encoding into a first-order calculus, normal bisimulations [10] coincide with (and are a practical alternative to) context bisimulations. Unfortunately, normal bisimulations have proved to be unsound in the presence of passivation (and restriction) [7]. While this result cast a doubt on whether sound normal bisimulations exist for higher-order distributed calculi, it did not affect the potential of environmental bisimulations [16,17,12,13] as a useful proof technique for behavioural equivalence in those calculi.

### 1.2 Our Contribution

To the best of our knowledge, there are not yet any useful sound bisimulations for higher-order distributed process calculi. In this paper we develop environmental (weak) bisimulations for the higher-order $\pi$-calculus with passivation, which (1) are sound with respect to reduction-closed barbed equivalence, (2) can actually be used to prove behavioural equivalence of non-trivial processes (with restrictions), and (3) can also be used to prove reduction-closed barbed *congruence* of processes (see Corollary 1). To prove reduction-closed barbed equivalence (and congruence), we find a new clause to guarantee preservation of bisimilarity by parallel composition of arbitrary processes. Unlike the corresponding clause in previous research [7,13], it can also handle the later removal (i.e. passivation) of these processes while keeping the bisimulation proofs tractable. Several examples are given, thereby supporting our claim of the first useful

bisimulations for a higher-order distributed process calculus. Moreover, we define an up-to context variant of the environmental bisimulations that significantly lightens the burden of equivalence proofs, as utilised in the examples.

*Overview of the bisimulation:* We now outline the definition of our environmental bisimulations. (Generalities on environmental bisimulations can be found in [12].) We define an environmental bisimulation $\mathcal{X}$ as a set of quadruples $(r, \mathcal{E}, P, Q)$ where $r$ is a set of names (i.e. channels and locations), $\mathcal{E}$ is a binary relation (called the *environment*) on terms, and $P$, $Q$ are processes. The bisimulation is a game where the processes $P$ and $Q$ are compared to each other by an *attacker* (or *observer*) who knows and can use the terms in the environment $\mathcal{E}$ and the names in $r$. For readability, the membership $(r, \mathcal{E}, P, Q) \in \mathcal{X}$ is often written $P\ \mathcal{X}_{\mathcal{E};r}\ Q$, and should be understood as "processes $P$ and $Q$ are bisimilar, under the environment $\mathcal{E}$ and the known names $r$."

The environmental bisimilarity is co-inductively defined by several conditions concerning the tested processes and the knowledge. As usual with weak bisimulations, we require that an internal transition by one of the processes is matched by zero or more internal transitions by the other, and that the remnants are still bisimilar.

As usual with (more recent and less common) environmental bisimulations, we require that whenever a term $M$ is output to a known channel, the other tested process can output another term $N$ to the same channel, and that the residues are bisimilar under the environment extended with the pair $(M, N)$. The extension of the environment stands for the growth of knowledge of the attacker of the bisimulation game who observed the outputs $(M, N)$, although he cannot analyse them. This spells out like: for any $P\ \mathcal{X}_{\mathcal{E};r}\ Q$ and $a \in r$, if $P \xrightarrow{\nu\widetilde{c}.\overline{a}\langle M\rangle} P'$ for fresh $\widetilde{c}$, then $Q \xRightarrow{\nu\widetilde{d}.\overline{a}\langle N\rangle} Q'$ for fresh $\widetilde{d}$ and $P'\ \mathcal{X}_{\mathcal{E}\cup\{(M,N)\};r}\ Q'$.

Unsurprisingly, input must be doable on the same known channel by each process, and the continuations must still be bisimilar under the same environment since nothing is learnt by the context. However, we require that the input terms are generated from the *context closure* of the environment. Intuitively, this closure represents all the processes an attacker can build by combining what he has learnt from previous outputs. Roughly, we define it as:

$$(\mathcal{E}; r)^\star = \{(C[\widetilde{M}], C[\widetilde{N}]) \mid C\ context,\ fn(C) \subseteq r,\ \widetilde{M}\ \mathcal{E}\ \widetilde{N}\}$$

where $\widetilde{M}$ denotes a sequence $M_0, \ldots, M_n$, and $\widetilde{M}\mathcal{E}\widetilde{N}$ means that for all $0 \leq i \leq n$, $M_i\,\mathcal{E}\,N_i$. Therefore, the input clause looks like: for any $P\ \mathcal{X}_{\mathcal{E};r}\ Q,\ a \in r$ and $(M, N) \in (\mathcal{E}; r)^\star$, if $P \xrightarrow{a(M)} P'$, then $Q \xRightarrow{a(N)} Q'$ and $P'\ \mathcal{X}_{\mathcal{E};r}\ Q'$.

The set $r$ of known names can be extended at will by the observer, provided that the new names are fresh: for any $P\ \mathcal{X}_{\mathcal{E};r}\ Q$ and $n$ fresh, we have $P\ \mathcal{X}_{\mathcal{E};r\cup\{n\}}\ Q$.

*Parallel composition:* The last clause is crucial to the soundness and usefulness of environmental bisimulations for languages with passivation, and not as straightforward as the other clauses. The idea at its base is that not only may an observer run arbitrary processes $R$ in parallel to the tested ones (as in reduction-closed barbed equivalence), but he may also run arbitrary processes $M, N$ he assembled from previous observations. It is critical to ensure that bisimilarity (and hopefully equivalence) is preserved by such parallel composition, and that this property can be easily proved. As $(\mathcal{E}; r)^\star$ is this set of

processes that can be assembled from previous observations, we would naively expect the appropriate clause to look like:

$$\text{For any } P\ \mathcal{X}_{\mathcal{E};r}\ Q \text{ and } (M,N) \in (\mathcal{E};r)^\star,\ \text{we have } P \mid M\ \mathcal{X}_{\mathcal{E};r}\ Q \mid N$$

but this subsumes the already impractical clause of reduction-closed barbed equivalence which we want to get round. Previous research [7,13] uses a weaker condition:

$$\text{For any } P\ \mathcal{X}_{\mathcal{E};r}\ Q \text{ and } (M,N) \in \mathcal{E},\ \text{we have } P \mid M\ \mathcal{X}_{\mathcal{E};r}\ Q \mid N$$

arguing that $(\mathcal{E};r)^\star$ can informally do no more observations than $\mathcal{E}$, but this clause is unsound in the presence of passivation. The reason behind the unsoundness is that, in our settings, not only can a context spawn new processes $M$, $N$, but it can also *remove* running processes it created by passivating them later on. For example, consider the following processes $P = \overline{a}\langle R\rangle.!R$ and $Q = \overline{a}\langle 0\rangle.!R$. Under the above weak condition, it would be easy to construct an environmental bisimulation that relates $P$ and $Q$. However, a process $a(X).m[X]$ may distinguish them. Indeed, it may receive processes $R$ and start running it in location $m$, or may receive process $0$ and run a copy of $R$ from $!R$. If $R$ is a process doing several sequential actions (for example if $R = lock.unlock$) and is passivated *in the middle* of its execution, then the remaining processes after passivation would not be equivalent any more.

To account for this new situation, we decide to modify the condition on the provenance of process that can be spawned, drawing them from $\{(a[M], a[N]) \mid a \in r,\ (M,N) \in \mathcal{E}\}$, thus giving the clause:

$$\text{For any } P\ \mathcal{X}_{\mathcal{E};r}\ Q,\ a \in r \text{ and } (M,N) \in \mathcal{E},\ \text{we have } P \mid a[M]\ \mathcal{X}_{\mathcal{E};r}\ Q \mid a[N].$$

The new condition allows for any running process that has been previously created by the observer to be passivated, that is, removed from the current test. This clause is much more tractable than the first one using $(\mathcal{E};r)^\star$ and, unlike the second one using only $\mathcal{E}$, leads to sound environmental bisimulations (albeit with a limitation; see Remark 1).

*Example:* With our environmental bisimulations, non-trivial equivalence of higher-order distributed processes can be shown, such as $P_0 = !a[e \mid \overline{e}]$ and $Q_0 = !a[e] \mid !a[\overline{e}]$, where $e$ abbreviates $e(X).0$ and $\overline{e}$ is $\overline{e}\langle 0\rangle.0$. We explain here informally how we build a bisimulation $\mathcal{X}$ relating those processes.

$$\begin{aligned}\mathcal{X} = \{(r,\mathcal{E},P,Q) \mid\ & r \supseteq \{a,e\},\ \ \mathcal{E} = \{0,e,\overline{e},e \mid \overline{e}\} \times \{0,e,\overline{e}\},\\ & P \equiv P_0 \mid \textstyle\prod_{i=1}^n l_i[M_i],\ \ Q \equiv Q_0 \mid \prod_{i=1}^n l_i[N_i],\ \ n \geq 0,\\ & \widetilde{l} \in r,\ \ (\widetilde{M},\widetilde{N}) \in \mathcal{E}\}\end{aligned}$$

Since we want $P_0\ \mathcal{X}_{\mathcal{E};r}\ Q_0$, the spawning clause of the bisimulation requires that for any $(M_1,N_1) \in \mathcal{E}$ and $l_1 \in r$, we have $P_0 \mid l_1[M_1]\ \mathcal{X}_{\mathcal{E};r}\ Q_0 \mid l_1[N_1]$. Then, by repeatedly applying this clause, we obtain $(P_0 \mid \prod_{i=1}^n l_i[M_i])\ \mathcal{X}_{\mathcal{E};r}\ (Q_0 \mid \prod_{i=1}^n l_i[N_i])$. Since the observer can add fresh names at will, we require $r$ to be a superset of the free names $\{a,e\}$ of $P_0$ and $Q_0$. Also, we have the intuition that the only possible outputs from $P$ and $Q$ are processes $e \mid \overline{e}$, $e$, $\overline{e}$, and $0$. Thus, we set ahead $\mathcal{E}$ as the Cartesian product of $\{0,e,\overline{e},e \mid \overline{e}\}$ with $\{0,e,\overline{e}\}$, that is, the combination of expectable outputs. We emphasize that it is indeed reasonable to relate $\overline{e}, e$ and $e \mid \overline{e}$ to $0, e$ and $\overline{e}$ in $\mathcal{E}$ for the observer cannot analyse the pairs: he can only use them along the tested processes $P$ and $Q$ which, by the design of environmental bisimulations, will make up for the differences.

$$
\begin{array}{lll}
 & \xrightarrow{\gamma \in \{e,\overline{e}\}} & P_0 \mid a[\overline{\gamma}] \mid \prod_{i=1}^{n} l_i[M_i] \quad \text{(i)} \\
 & & = P_0 \mid \prod_{i=1}^{n+1} l_i[M_i] \quad \text{for } l_{n+1} = a,\ M_{n+1} = \overline{\gamma} \\
P_0 \mid \prod_{i=1}^{n} l_i[M_i] & \xrightarrow{\overline{l_n}\langle M_n \rangle} & P_0 \mid \prod_{i=1}^{n-1} l_i[M_i] \quad \text{(ii)} \\
 & \xrightarrow{\alpha \in \{e,\overline{e}\}} & P_0 \mid \prod_{i=1}^{n-1} l_i[M_i] \mid l_n[M'_n] \quad \text{(iii)} \\
 & & \equiv P_0 \mid a[e \mid \overline{e}] \mid \prod_{i=1}^{n-1} l_i[M_i] \mid l_n[M'_n] \\
 & & = P_0 \mid \prod_{i=1}^{n+1} l_i[M'_i] \\
 & & \text{for} \quad M'_i = M_i \ (0 \le i \le n-1),\ M_n \xrightarrow{\alpha} M'_n, \\
 & & \quad l_{n+1} = a,\ M_{n+1} = e \mid \overline{e}.
\end{array}
$$

$$
\begin{array}{lll}
 & \xrightarrow{\gamma} & Q_0 \mid a[0] \mid \prod_{i=1}^{n} l_i[N_i] \quad \text{(i)} \\
 & & = Q_0 \mid \prod_{i=1}^{n+1} l_i[N_i] \quad \text{for } l_{n+1} = a,\ N_{n+1} = 0 \\
Q_0 \mid \prod_{i=1}^{n} l_i[N_i] & \xrightarrow{\overline{l_n}\langle N_n \rangle} & Q_0 \mid \prod_{i=1}^{n-1} l_i[N_i] \quad \text{(ii)} \\
 & \xrightarrow{\alpha} & Q_0 \mid a[0] \mid \prod_{i=1}^{n} l_i[N_i] \quad \text{(iii)} \\
 & & = Q_0 \mid \prod_{i=1}^{n+1} l_i[N_i] \quad \text{for } l_{n+1} = a,\ N_{n+1} = 0
\end{array}
$$

**Fig. 1.** Simulation of observable transitions

Let us now observe the possible transitions from $P$ and their corresponding transitions from $Q$ by glossing over two pairs of trees, where related branches represent the correspondences. (Simulation in the other direction is similar and omitted for brevity.) First, let us consider the input and output actions as shown in Figure 1. (i) When $P_0$ does an input action $e$ or an output action $\overline{e}$, it leaves behind a process $a[\overline{e}]$ or $a[e]$, respectively. $Q_0$ can also do the same action, leaving $a[0]$. Since both $(\overline{e}, 0)$ and $(e, 0)$ are in $\mathcal{E}$, we can add the leftover processes to the respective products $\prod$; (ii) output by passivation is trivial to match (without loss of generality, we only show the case $i = n$), and (iii) observable actions $\alpha$ of an $M_n$, leaving a residue $M'_n$, are matched by one of $Q_0$'s $a[\alpha]$, leaving $a[0]$. To pair with this $a[0]$, we replicate an $a[e \mid \overline{e}]$ from $P_0$, and then, as in (i), they add up to the products $\prod$.

In a similar way, we explain how $\tau$ transitions of $P$ are matched by $Q$, with another pair of transitions trees described in Figure 2.

(1) When an $a[e \mid \overline{e}]$ from $P_0$ turns into $a[0]$, $Q$ does not have to do any action, for we work with weak bisimulations. By replication, $Q$ can produce a copy $a[e]$ (or alternatively $a[\overline{e}]$) from $Q_0$, and since $(0, e)$ is in $\mathcal{E}$, we can add the $a[0]$ and the copy $a[e]$ to the products $\prod$; (2) $P$ can also make a reaction between two copies of $a[e \mid \overline{e}]$ in $P_0$, leaving behind $a[e]$ and $a[\overline{e}]$. As in (1), $Q$ can draw two copies of $a[e]$ from $Q_0$, and each product can be enlarged by two elements; (3) it is also possible for $M_n = e \mid \overline{e}$ to do a $\tau$ transition, becoming $M'_n = 0$. It stands that $(M'_n, N_n) \in \mathcal{E}$ and we are done; (4) very similarly, two processes $M_n$ and $M_{n-1}$ may react, becoming $M'_n$ and $M'_{n-1}$. It stands also that $(M'_{n-1}, N_{n-1})$ and $(M'_n, N_n)$ are in $\mathcal{E}$, so the resulting processes are still related; (5) it is possible for $M_n$ to follow the transition $M_n \xrightarrow{\alpha} M'_n$ and react with a copy from $P_0$ which leaves behind $a[\alpha]$ (since $\overline{\alpha}$ has been consumed to conclude the

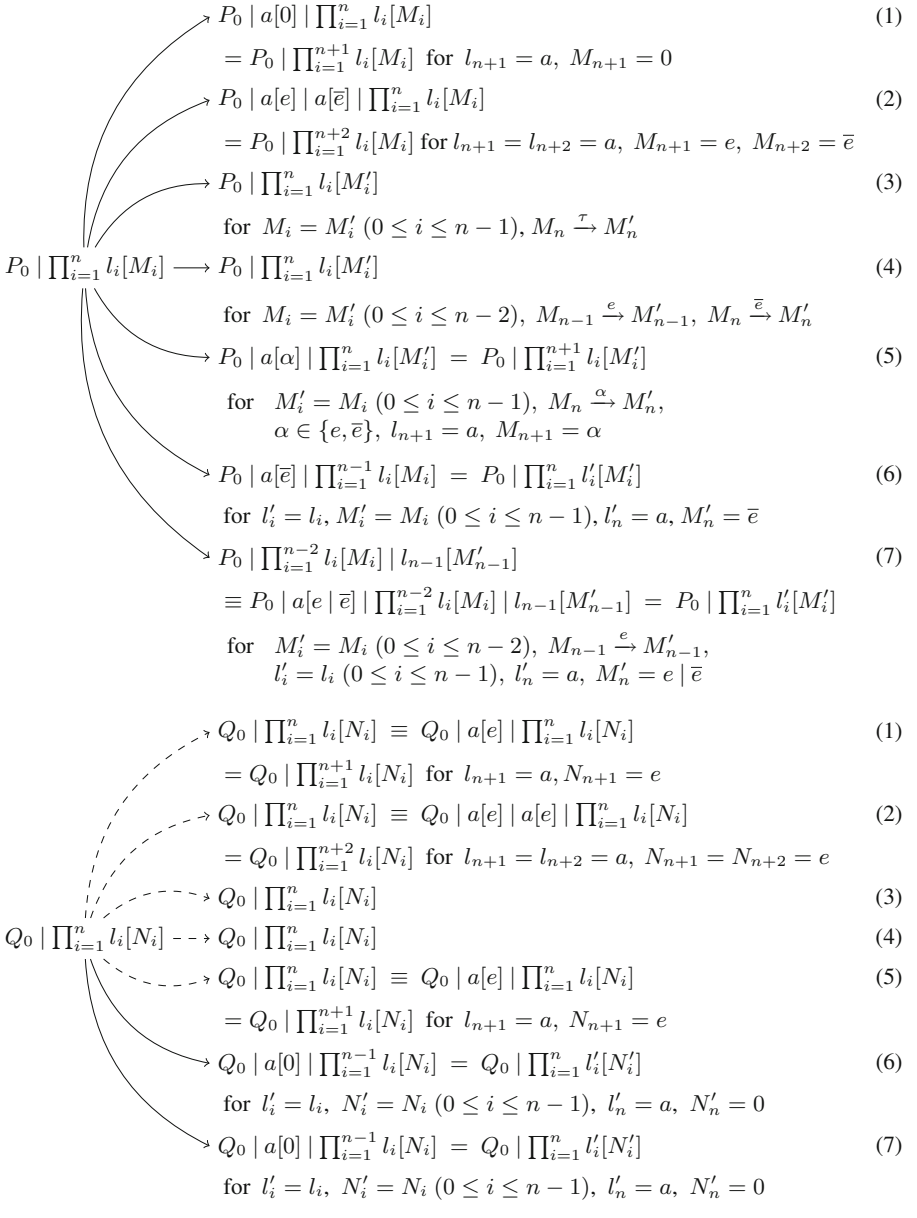


**Fig. 2.** Simulation of internal transitions (dotted lines mean zero transitions)

reaction). Again, it stands that $M'_n$ and $N_n$ are related by $\mathcal{E}$, and that we can draw an $a[e]$ from $Q_0$ to pair it with the residue $M'_n$ in the products $\prod$; (6) also, a copy $a[e \mid \overline{e}]$ from $P_0$ may passivate an $l_i[M_i]$, provided $l_i = e$, and leave a residue $a[\overline{e}]$. $Q$ can do the same passivation using $Q_0$'s $a[e]$, and leave $a[0]$. As it happens that $(\overline{e}, 0)$ is in $\mathcal{E}$, the residues can be added to the products too; (7) finally, the process $l_n[M_n]$, if $l_n = e$, may be passivated by $M_{n-1}$, reducing the size of $P$'s product. $Q$ can passivate $l_n[N_n]$ too, using a copy $a[e]$ from $P_0$, which becomes $a[0]$ after the reaction. $Q$'s product too is shorter, but we need to add the $a[0]$ to it. To do so, we draw a copy $a[e \mid \overline{e}]$ from $P_0$, and since $(e \mid \overline{e}, 0)$ is in $\mathcal{E}$, $a[e \mid \overline{e}]$ and $a[0]$ are merged into their respective product.

This ends the sketch of the proof that $\mathcal{X}$ is an environmental bisimulation, and therefore that $!a[e \mid \overline{e}]$ and $!a[e] \mid a[\overline{e}]$ are behaviourally equivalent.

### 1.3 Overview of the Paper

The rest of this paper is structured as follows. In Section 2 we describe the higher-order $\pi$-calculus with passivation. In Section 3 we formalize our environmental bisimulations. In Section 4 we give some examples of bisimilar processes. In Section 5, we bring up some future work to conclude our paper.

## 2 Higher-Order $\pi$-Calculus with Passivation

We introduce a slight variation of the higher-order $\pi$-calculus with passivation [7]—HO$\pi$P for short—through its syntax and a labelled transitions system.

### 2.1 Syntax

The syntax of our HO$\pi$P processes $P$, $Q$ is given by the following grammar, very similar to that of Lenglet *et al.* [7] (the higher-order $\pi$-calculus extended with located processes and their passivation):

$$P, Q \ ::= 0 \mid a(X).P \mid \overline{a}\langle M \rangle.P \mid (P \mid P) \mid a[P] \mid \nu a.P \mid !P \mid \mathit{run}(M)$$
$$M, N ::= X \mid {}`P$$

$X$ ranges over the set of variables, and $a$ over the set of names which can be used for both locations and channels. $a[P]$ denotes the process $P$ running in location $a$. To define a general up-to context technique (Definition 2, see also Section 5), we distinguish terms $M, N$ from processes $P, Q$ and adopt explicit syntax for processes as terms $`P$ and their execution $\mathit{run}(M)$.

### 2.2 Labelled Transitions System

We define $n$, $fn$, $bn$ and $fv$ to be the functions that return respectively the set of names, free names, bound names and free variables of a process or an action. We abbreviate a (possibly empty) sequence $x_0, x_1, \ldots, x_n$ as $\widetilde{x}$ for any meta-variable $x$. The transition semantics of HO$\pi$P is given by the following labelled transition system, which is based on that of the higher-order $\pi$-calculus (omitting symmetric rules PAR-R and REACT-R):

$$\frac{}{a(X).P \xrightarrow{a(M)} P\{M/X\}}\ \textsc{Ho-in} \qquad \frac{}{\overline{a}\langle M\rangle.P \xrightarrow{\overline{a}\langle M\rangle} P}\ \textsc{Ho-out}$$

$$\frac{P_1 \xrightarrow{\alpha} P_1' \quad bn(\alpha) \cap fn(P_2) = \emptyset}{P_1 \mid P_2 \xrightarrow{\alpha} P_1' \mid P_2}\ \textsc{Par-l} \qquad \frac{!P \mid P \xrightarrow{\alpha} P'}{!P \xrightarrow{\alpha} P'}\ \textsc{Rep}$$

$$\frac{P_1 \xrightarrow{(\nu\widetilde{b}).\overline{a}\langle M\rangle} P_1' \quad P_2 \xrightarrow{a(M)} P_2' \quad \{\widetilde{b}\} \cap fn(P_2) = \emptyset}{P_1 \mid P_2 \xrightarrow{\tau} \nu\widetilde{b}.(P_1' \mid P_2')}\ \textsc{React-l}$$

$$\frac{P \xrightarrow{\alpha} P' \quad a \notin n(\alpha)}{\nu a.P \xrightarrow{\alpha} \nu a.P'}\ \textsc{Guard} \qquad \frac{P \xrightarrow{(\nu\widetilde{b}).\overline{a}\langle M\rangle} P' \quad c \neq a \quad c \in fn(M) \setminus \{\widetilde{b}\}}{\nu c.P \xrightarrow{\nu(\widetilde{b},c).\overline{a}\langle M\rangle} P'}\ \textsc{Extr}$$

extended with the following three rules:

$$\frac{P \xrightarrow{\alpha} P'}{a[P] \xrightarrow{\alpha} a[P']}\ \textsc{Transp} \qquad \frac{}{a[P] \xrightarrow{\overline{a}\langle `P\rangle} 0}\ \textsc{Passiv} \qquad \frac{}{\mathit{run}(`P) \xrightarrow{\tau} P}\ \textsc{Run}$$

Assuming again knowledge of the standard higher-order $\pi$-calculus [9,11], we only explain below the three added rules that are not part of it. The `Transp` rule expresses the *transparency* of locations, the fact that transitions can happen below a location and be observed outside its boundary. The `Passiv` rule illustrates that, at any time, a process running under a location can be passivated (stopped and turned into a term) and sent along the channel corresponding to the location's name. Quotation of the process output reminds us that higher-order communications transport terms. Finally, the `Run` rule shows how, at the cost of an internal transition, a process term be instantiated. As usual with small-steps semantics, transition does not progress for undefined cases (such as $\mathit{run}(X)$) or when the assumptions are not satisfied.

Henceforth, we shall write $\overline{a}.P$ to mean $\overline{a}\langle `0\rangle.P$ and $a.P$ for $a(X).P$ if $X \notin fv(P)$. We shall also write $\equiv$ for the structural congruence, whose definition is standard (see the appendix, Definition A.1).

## 3 Environmental Bisimulations of HO$\pi$P

Given the higher-order nature of the language, and in order to get round the universal quantification issue of context bisimulations, we would like observations (terms) to be stored and reusable for further testing. To this end, let us define an *environmental relation* $\mathcal{X}$ as a set of elements $(r, \mathcal{E}, P, Q)$ where $r$ is a finite set of names, $\mathcal{E}$ is a binary relation (with finitely many free names) on variable-closed terms (i.e. terms with no free variables), and $P$ and $Q$ are variable-closed processes.

We generally write $x \oplus S$ to express the set union $\{x\} \cup S$. We also use graphically convenient notation $P\ \mathcal{X}_{\mathcal{E};r}\ Q$ to mean $(r, \mathcal{E}, P, Q) \in \mathcal{X}$ and define the *term context closure* $(\mathcal{E}; r)^\star = \mathcal{E} \cup \{(`P, `Q) \mid (P, Q) \in (\mathcal{E}; r)^\circ\}$ with the *process context closure* $(\mathcal{E}; r)^\circ = \{(C[\widetilde{M}], C[\widetilde{N}]) \mid \widetilde{M}\,\mathcal{E}\,\widetilde{N},\ C\ \textit{context},\ bn(C) \cap fn(\mathcal{E}, r) = \emptyset,\ fn(C) \subseteq r\}$, where a context is a process with zero or more *holes* for terms. Note the distinction of terms $`P$, $`Q$ from processes $P$, $Q$. We point out that $(\emptyset; r)^\star$ is the identity on terms

with free names in $r$, that $(\mathcal{E};r)^\star$ includes $\mathcal{E}$ by definition, and that the context closure operations are monotonic on $\mathcal{E}$ (and $r$). Therefore, for any $\mathcal{E}$ and $r$, the set $(\mathcal{E};r)^\star$ includes the identity $(\emptyset;r)^\star$ too. Also, we use the notations $\mathcal{S}.1$ and $\mathcal{S}.2$ to denote the first and second projections of a relation (i.e. set of pairs) $S$. Finally, we define weak transitions $\Rightarrow$ as the reflexive, transitive closure of $\xrightarrow{\tau}$, and $\overset{\alpha}{\Rightarrow}$ as $\Rightarrow \xrightarrow{\alpha} \Rightarrow$ for $\alpha \neq \tau$ (and define $\overset{\tau}{\Rightarrow}$ as $\Rightarrow$).

We can now define environmental bisimulations formally:

**Definition 1.** *An environmental relation $\mathcal{X}$ is an environmental bisimulation if $P\ \mathcal{X}_{\mathcal{E};r}\ Q$ implies:*

1. *if $P \xrightarrow{\tau} P'$, then $\exists Q'.\ Q \Rightarrow Q'$ and $P'\ \mathcal{X}_{\mathcal{E};r}\ Q'$,*
2. *if $P \xrightarrow{a(M)} P'$ with $a \in r$, and if $(M,N) \in (\mathcal{E};r)^\star$, then $\exists Q'.\ Q \overset{a(N)}{\Longrightarrow} Q'$ and $P'\ \mathcal{X}_{\mathcal{E};r}\ Q'$,*
3. *if $P \xrightarrow{\nu\widetilde{b}.\overline{a}\langle M\rangle} P'$ with $a \in r$ and $\widetilde{b} \notin fn(r,\mathcal{E}.1)$, then $\exists Q', N.\ Q \overset{\nu\widetilde{c}.\overline{a}\langle N\rangle}{\Longrightarrow} Q'$ with $\widetilde{c} \notin fn(r,\mathcal{E}.2)$ and $P'\ \mathcal{X}_{(M,N)\oplus\mathcal{E};r}\ Q'$,*
4. *for any $(`P_1, `Q_1) \in \mathcal{E}$ and $a \in r$, we have $P \mid a[P_1]\ \mathcal{X}_{\mathcal{E};r}\ Q \mid a[Q_1]$,*
5. *for any $n \notin fn(\mathcal{E},P,Q)$, we have $P\ \mathcal{X}_{\mathcal{E};n\oplus r}\ Q$, and*
6. *the converse of 1, 2 and 3 on $Q$'s transitions.*

Modulo the symmetry resulting from clause 6, clause 1 is usual; clause 2 enforces bisimilarity to be preserved by any input that can be built from the knowledge, hence the use of the context closure; clause 3 enlarges the knowledge of the observer with the leaked out terms. Clause 4 allows the observer to spawn (and immediately run) terms concurrently to the tested processes, while clause 5 shows that he can also create fresh names at will.

A few points related to the handling of free names are worth mentioning: as the set of free names in $\mathcal{E}$ is finite, clause 5 can always be applied; therefore, the attacker can add arbitrary fresh names to the set $r$ of known names so as to use them in terms $M$ and $N$ in clause 2. Fresh $\widetilde{b}$ and $\widetilde{c}$ in clause 3 also exist thanks to the finiteness of free names in $\mathcal{E}$ and $r$.

We define environmental bisimilarity $\sim$ as the union of all environmental bisimulations, and it holds that it is itself an environmental bisimulation (all the conditions above are monotone on $\mathcal{X}$). Therefore, $P \sim_{\mathcal{E};r} Q$ if and only if $P\ \mathcal{X}_{\mathcal{E};r}\ Q$ for some environmental bisimulation $\mathcal{X}$. We do particularly care about the situation where $\mathcal{E} = \emptyset$ and $r = fn(P,Q)$. It corresponds to the equivalence of two processes when the observer knows all of their free names (and thus can do all observations), but has not yet learnt any output pair.

For improving the practicality of our bisimulation proof method, let us devise an up-to context technique [11, p. 86]: for an environmental relation $\mathcal{X}$, we write $P\ \mathcal{X}^\star_{\mathcal{E};r}\ Q$ if $P \equiv \nu\widetilde{c}.(P_0 \mid P_1)$, $Q \equiv \nu\widetilde{d}.(Q_0 \mid Q_1)$, $P_0\ \mathcal{X}_{\mathcal{E}';r'}\ Q_0$, $(P_1,Q_1) \in (\mathcal{E}';r')^\circ$, $\mathcal{E} \subseteq (\mathcal{E}';r')^\star$, $r \subseteq r'$, and $\{\widetilde{c}\} \cap fn(r,\mathcal{E}.1) = \{\widetilde{d}\} \cap fn(r,\mathcal{E}.2) = \emptyset$. As a matter of fact, this is actually an up-to context and up-to environment and up-to restriction and up-to structural congruence technique, but because of the clumsiness of this appellation we will restrain ourselves to "up-to context" to preserve clarity. To roughly explain the

convenience behind this notation and its (long) name: (1) "up-to context" states that we can take any $(P_1, Q_1)$ from the (process) context closure $(\mathcal{E}'; r')^\circ$ of the environment $\mathcal{E}'$ (with free names in $r'$) and execute them in parallel with processes $P_0$ and $Q_0$ related by $\mathcal{X}_{\mathcal{E}';r'}$; similarly, we allow environments $\mathcal{E}$ with terms that are not in $\mathcal{E}'$ itself but are in the (term) context closure $(\mathcal{E}'; r')^\star$; (2) "up-to environment" states that, when proving the bisimulation clauses, we please ourselves with environments $\mathcal{E}'$ that are *larger* than the $\mathcal{E}$ requested by Definition 1; (3) "up-to restriction" states that we also content ourselves with tested processes $P, Q$ with extra restrictions $\nu\widetilde{c}$ and $\nu\widetilde{d}$ (i.e. less observable names); (4) finally, "up-to structural congruence" states that we identify all processes that are structurally congruent to $\nu\widetilde{c}.(P_0 \mid P_1)$ and $\nu\widetilde{d}.(Q_0 \mid Q_1)$.

Using this notation, we define environmental bisimulations up-to context as follows:

**Definition 2.** *An environmental relation $\mathcal{X}$ is an environmental bisimulation up-to context if $P\ \mathcal{X}_{\mathcal{E};r}\ Q$ implies:*

1. *if $P \xrightarrow{\tau} P'$, then $\exists Q'.\ Q \Rightarrow Q'$ and $P'\ \mathcal{X}^\star_{\mathcal{E};r}\ Q'$,*
2. *if $P \xrightarrow{a(M)} P'$ with $a \in r$, and if $(M, N) \in (\mathcal{E}; r)^\star$, then $\exists Q'.\ Q \xRightarrow{a(N)} Q'$ and $P'\ \mathcal{X}^\star_{\mathcal{E};r}\ Q'$,*
3. *if $P \xrightarrow{\nu\widetilde{b}.\overline{a}\langle M\rangle} P'$ with $a \in r$ and $\widetilde{b} \notin fn(r, \mathcal{E}.1)$, then $\exists Q', N.\ Q \xRightarrow{\nu\widetilde{c}.\overline{a}\langle N\rangle} Q'$ with $\widetilde{c} \notin fn(r, \mathcal{E}.2)$ and $P'\ \mathcal{X}^\star_{(M,N)\oplus\mathcal{E};r}\ Q'$,*
4. *for any $(\text{`}P_1, \text{`}Q_1) \in \mathcal{E}$ and $a \in r$, we have $P \mid a[P_1]\ \mathcal{X}^\star_{\mathcal{E};r}\ Q \mid a[Q_1]$,*
5. *for any $n \notin fn(\mathcal{E}, P, Q)$, we have $P\ \mathcal{X}_{\mathcal{E};n\oplus r}\ Q$, and*
6. *the converse of 1, 2 and 3 on $Q$'s transitions.*

The conditions on each clause (except 5, which is unchanged for the sake of technical convenience) are weaker than that of the standard environmental bisimulations, as we require in the positive instances bisimilarity modulo a context, not just bisimilarity itself. It is important to remark that, unlike in [12] but as in [13], we do not need a specific context to avoid stating a tautology in clause 4; indeed, we spawn terms $(\text{`}P_1, \text{`}Q_1) \in \mathcal{E}$ immediately as processes $P_1$ and $Q_1$, while the context closure can only use the terms under an explicit *run* operator.

We prove the soundness (under some condition; see Remark 1) of environmental bisimulations as follows. Full proofs are found in the appendix, Section B but are nonetheless sketched below.

**Lemma 1 (Input lemma).** *If $(P_1, Q_1) \in (\mathcal{E}; r)^\circ$ and $P_1 \xrightarrow{a(M)} P_1'$ then $\forall N.\exists Q_1'.\ Q_1 \xrightarrow{a(N)} Q_1'$ and $(P_1', Q_1') \in ((M, N)\oplus\mathcal{E}; r)^\circ$.*

**Lemma 2 (Output lemma).** *If $(P_1, Q_1) \in (\mathcal{E}; r)^\circ$, $\{\widetilde{b}\} \cap fn(\mathcal{E}, r) = \emptyset$ and $P_1 \xrightarrow{\nu\widetilde{b}.\overline{a}\langle M\rangle} P_1'$ then $\exists Q_1', N.\ Q_1 \xrightarrow{\nu\widetilde{b}.\overline{a}\langle N\rangle} Q_1'$, $(P_1', Q_1') \in (\mathcal{E}; \widetilde{b}\oplus r)^\circ$ and $(M, N) \in (\mathcal{E}; \widetilde{b}\oplus r)^\star$.*

**Definition 3 (Run-erasure).** *We write $P \le Q$ if $P$ can be obtained by (possibly repeatedly) replacing zero or more subprocesses run($\text{`}R$) of $Q$ with $R$, and write $P\ \mathcal{Y}^-_{\mathcal{E};r}\ Q$ for $P \le \mathcal{Y}^\star_{\le\mathcal{E}\ge;r} \ge Q$.*

**Definition 4 (Simple environment).** *A process is called* simple *if none of its subprocesses has the form $\nu a.P$ or $a(X).P$ with $X \in fv(P)$. An environment is called simple if all the processes in it are simple. An environmental relation is called simple if all of its environments are simple (note that the tested processes may still be non-simple).*

**Lemma 3 (Reaction lemma).** *For any simple environmental bisimulation up-to context $\mathcal{Y}$, if $P\ \mathcal{Y}^{-}_{\mathcal{E};r}\ Q$ and $P \xrightarrow{\tau} P'$, then there is a $Q'$ such that $Q \stackrel{\tau}{\Rightarrow} Q'$ and $P'\ \mathcal{Y}^{-}_{\mathcal{E};r}\ Q'$.*

*Proof sketch.* Lemma 1 (resp. 2) is proven by straightforward induction on the transition derivation of $P_1 \xrightarrow{a(M)} P_1'$ (resp. $P_1 \xrightarrow{\nu\tilde{b}.\overline{a}\langle M\rangle} P_1'$). Lemma 3 is proven last, as it uses the other two lemmas (for the internal communication case).

**Lemma 4 (Soundness of up-to context).** *Simple bisimilarity up-to context is included in bisimilarity.*

*Proof sketch.* By checking that $\{(r, \mathcal{E}, P, Q) \mid P\ \mathcal{Y}^{-}_{\mathcal{E};r}\ Q\}$ is included in $\sim$, where $\mathcal{Y}$ is the simple environmental bisimilarity up-to context. In particular, we use Lemma 1 for clause 2, Lemma 2 for clause 3, and Lemma 3 for clause 1 of the environmental bisimulation.

Our definitions of reduction-closed barbed equivalence $\approx$ and congruence $\approx_c$ are standard and omitted for brevity; see the appendix, Definition B.2 and B.3

**Theorem 1 (Barbed equivalence from environmental bisimulation)**
*If $P\ \mathcal{Y}^{-}_{\emptyset;fn(P,Q)}\ Q$ for a simple environmental bisimulation up-to context $\mathcal{Y}$, then $P \approx Q$.*

*Proof sketch.* By verifying that each clause of the definition of $\approx$ is implied by membership of $\mathcal{Y}^{-}$, using Lemma 4 for the parallel composition clause.

**Corollary 1 (Barbed congruence from environmental bisimulation)**
*If $\overline{a}\langle\!\langle P\rangle\!\rangle\ \mathcal{Y}^{-}_{\emptyset;a\oplus fn(P,Q)}\ \overline{a}\langle\!\langle Q\rangle\!\rangle$ for a simple environmental bisimulation up-to context $\mathcal{Y}$, then $P \approx_c Q$.*

We recall that, in context bisimulations, showing the equivalence of $\overline{a}\langle\!\langle P\rangle\!\rangle$ and $\overline{a}\langle\!\langle Q\rangle\!\rangle$ almost amounts to testing the equivalence of $P$ and $Q$ in every context. However, with environmental bisimulations, only the location context in clause 4 of the bisimulation has to be considered.

*Remark 1.* The extra condition "simple" is needed because of a technical difficulty in the proof of Lemma 3: when an input process $a(X).P$ is spawned under location $b$ in parallel with an output context $\nu c.\overline{a}\langle M\rangle.Q$ (with $c \in fn(M)$), they can make the transition $b[a(X).P \mid \nu c.\overline{a}\langle M\rangle.Q] \xrightarrow{\tau} b[\nu c.(P\{M/X\} \mid Q)]$, where the restriction operator $\nu c$ appears *inside* the location $b$ (and therefore can be passivated together with the processes); however, our spawning clause only gives us $b[a(X).P] \mid \nu c.\overline{a}\langle M\rangle.Q \xrightarrow{\tau} \nu c.(b[P\{M/X\}] \mid Q)$ and does not cover the above case. Further investigation is required to overcome this difficulty (although we have not yet found a concrete counterexample of soundness, we conjecture some modification to the bisimulation clauses would be necessary). Note that, even if the environments are simple, the tested processes do not always have to be simple, as in Example 4 and 5. Moreover, thanks to up-to context, even the output terms (including passivated processes) can be non-simple.

## 4 Examples

Here, we give some examples of HO$\pi$P processes whose behavioural equivalence is proven with the help of our environmental bisimulations. In each example, we prove the equivalence by exhibiting a relation $\mathcal{X}$ containing the two processes we consider, and by showing that it is indeed a bisimulation up-to context (and environment, restriction and structural congruence). We write $P \mid \ldots \mid P$ for a finite, possibly null, product of the process $P$.

*Example 1.* $e \mid !a[e] \mid !a[0] \approx !a[e] \mid !a[0]$. (This example comes from [7].)

*Proof.* Take $\mathcal{X} = \{(r,\ \emptyset,\ e \mid P,\ P) \mid r \supseteq \{a, e\}\} \cup \{(r, \emptyset, P, P) \mid r \supseteq \{a, e\}\}$ where $P = !a[e] \mid !a[0]$. It is immediate to verify that whenever $P \xrightarrow{\alpha} P'$, we have $P' \equiv P$, and therefore that transition $e \mid P \xrightarrow{\alpha} e \mid P' \equiv e \mid P$ can be matched by $P \xrightarrow{\alpha} P' \equiv P$ and conversely. Also, for $e \mid P \xrightarrow{e} P$, we have that $P \xrightarrow{e} !a[e] \mid a[0] \mid !a[0] \equiv P$ and we are done since $(r, \emptyset, P, P) \in \mathcal{X}$. Moreover, the set $r$ must contain the free names of $P$, and to satisfy clause 5 about adding fresh names, bigger $r$'s must be allowed too. The passivations of $a[e]$ and $a[0]$ can be matched by syntactically equal actions with the pairs of output terms $(`e, `e)$ and $(`0, `0)$ included in the identity, which in turn is included in the context closure $(\emptyset; r)^\star$. Finally clause 4 of the bisimulation is vacuously satisfied because the environment is empty. We therefore have $e \mid !a[e] \mid !a[0] \approx !a[e] \mid !a[0]$ from the soundness of environmental bisimulation up-to context.

*Example 2.* $!\overline{a} \mid !e \ \approx !a[e]$.

*Proof sketch.* Take $\mathcal{X} = \{(r,\ \mathcal{E},\ P,\ Q) \mid r \supseteq \{a, e, l_1, \ldots, l_n\} \mid \mathcal{E} = \{(`0, `e)\},\ n \geq 0,\ P = !\overline{a} \mid !e \mid \prod_{i=1}^{n} l_i[0],\ Q = !a[e] \mid \prod_{i=1}^{n} l_i[e] \mid a[0] \mid \ldots \mid a[0]\}$. See the appendix, Example C.1 for the rest of the proof.

*Example 3.* $!a[e] \mid !b[\overline{e}] \approx !a[b[e \mid \overline{e}]]$. This example shows the equivalence proof of more complicated processes with nested locations.

*Proof sketch.* Take:

$$\begin{aligned}
\mathcal{X} = \{(r,\ \mathcal{E},\ P,\ Q) \mid r \ &\supseteq \{a, e, b, l_1, \ldots, l_n\},\\
&P_0 = !a[e] \mid !b[\overline{e}],\ Q_0 = !a[b[e \mid \overline{e}]],\\
&P \ = P_0 \mid \textstyle\prod_{i=1}^{n} l_i[P_i] \mid b[0] \mid \ldots \mid b[0],\\
&Q \ = Q_0 \mid \textstyle\prod_{i=1}^{n} l_i[Q_i],\\
&(`\widetilde{P}, `\widetilde{Q}) \in \mathcal{E},\ n \geq 0\},\\
\mathcal{E} \ = \{(`x, `y) \mid x \in \{0, e, \overline{e}\},\ y \equiv\in \{0, e, \overline{e}, (e \mid \overline{e}), b[0], b[e], b[\overline{e}], b[e \mid \overline{e}]\}\}&.
\end{aligned}$$

See the appendix, Example C.2 for the rest of the proof.

*Example 4.* $c(X).\mathit{run}(X) \approx \nu f.(f[c(X).\mathit{run}(X)] \mid !f(Y).f[\mathit{run}(Y)])$. The latter process models a system where a process $c(X).\mathit{run}(X)$ runs in location $f$, and executes any process $P$ it has received. In parallel is a process $f(Y).f[\mathit{run}(Y)]$ which can passivate $f[P]$ and respawn the process $P$ under the same location $f$. Informally, this models a system which can restart a computer and resume its computation after a failure.

*Proof.* Take $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$ where:

$$\mathcal{X}_1 = \{(r,\ \emptyset,\ c(X).\mathit{run}(X),\ \nu f.(f[c(X).\mathit{run}(X)] \mid !f(Y).f[\mathit{run}(Y)])) \mid r \supseteq \{c\}\},$$
$$\mathcal{X}_2 = \{(r,\ \emptyset,\ P,\ Q) \mid r \supseteq c \oplus \mathit{fn}(R), \quad S = \mathit{run}(`\mathit{run}(\ldots `\mathit{run}(`R)\ldots)),$$
$$P \in \{\mathit{run}(`R), R\},\ \ Q = \nu\widetilde{f}.(f[S] \mid !f(Y).[\mathit{run}(Y)])\}.$$

As usual, we require that $r$ contains at least the free name $c$ of the tested processes. All outputs belong to $(\emptyset; r)^\star$ since they come from a process $R$ drawn from $(\emptyset; r)^\star$, and therefore, we content ourselves with an empty environment $\emptyset$. Also, by the emptiness of the environment, clause 4 of environmental bisimulations is vacuously satisfied.

Verification of transitions of elements of $\mathcal{X}_1$, i.e. inputs of some $`R$ (with $(`R, `R) \in (\emptyset; r)^\star$) from $c$, is immediate and leads to checking elements of $\mathcal{X}_2$. For elements of $\mathcal{X}_2$, we observe that $P = \mathit{run}(`R)$ can do one $\tau$ transition to become $R$, while $Q$ can do an internal transition passivating the process $\mathit{run}(`R)$ running in $f$ and place it inside $f[\mathit{run}(\ `)]$, again and again. $Q$ can also do $\tau$ transitions that consume all the $\mathit{run}(\ `)$'s until it becomes $R$. Whenever $P$ (resp. $Q$) makes an observable transition, $Q$ (resp. $P$) can consume the $\mathit{run}(\ `)$'s and weakly do the same action as they exhibit the same process. We observe that all transitions preserve membership in $\mathcal{X}_2$ (thus in $\mathcal{X}$), and therefore we have that $\mathcal{X}$ is an environmental bisimulation up-to context, which proves the behavioural equivalence of the original processes $c(X).\mathit{run}(X)$ and $c(X).\nu f.(f[c(X).\mathit{run}(X)] \mid !f(Y).f[\mathit{run}(Y)])$.

*Example 5.* $c(X).\mathit{run}(X) \approx c(X).\nu a.(\overline{a}\langle X\rangle \mid !\nu f.(f[a(X).\mathit{run}(X)] \mid f(Y).\overline{a}\langle Y\rangle))$. This example is a variation of Example 4 modelling a system where computation is resumed on another computer after a failure.

*Proof.* Take $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2 \cup \mathcal{X}_3$ where:

$$\mathcal{X}_1 = \{(r,\ \emptyset,\ c(X).\mathit{run}(X),\ c(X).\nu a.(\overline{a}\langle X\rangle \mid F)) \mid r \supseteq \{c\}\},$$
$$\mathcal{X}_2 = \{(r,\ \emptyset,\ P_1,\ \nu a.(F \mid R_1 \mid R_2 \mid \overline{a}\langle `P_2\rangle)) \mid$$
$$r \supseteq \{c\} \oplus \mathit{fn}(P),\ \ P_1, P_2 \in \{\mathit{run}(`P), P\},\ \ R_1 = \overline{a}\langle N_1\rangle \mid \ldots \mid \overline{a}\langle N_n\rangle,$$
$$R_2 = \nu l_1.(l_1[Q_1] \mid l_1(Y).\overline{a}\langle Y\rangle) \mid \ldots \mid \nu l_m.(l_m[Q_m] \mid l_m(Y).\overline{a}\langle Y\rangle),$$
$$N_1, \ldots, N_n, `Q_1, \ldots, `Q_m = `\mathit{run}(`\mathit{run}(\ldots `\mathit{run}(`a(X).\mathit{run}(X))\ldots)),\ n \geq 0\},$$
$$\mathcal{X}_3 = \{(r,\ \emptyset,\ P_1,\ \nu a.(F \mid R_1 \mid R_2 \mid \nu l.(l[P_2] \mid l(Y).\overline{a}\langle Y\rangle))) \mid$$
$$r \supseteq \{c\} \oplus \mathit{fn}(P),\ \ P_1, P_2 \in \{\mathit{run}(`P), P\},\ \ R_1 = \overline{a}\langle N_1\rangle \mid \ldots \mid \overline{a}\langle N_n\rangle,$$
$$R_2 = \nu l_1.(l_1[Q_1] \mid l_1(Y).\overline{a}\langle Y\rangle) \mid \ldots \mid \nu l_m.(l_m[Q_m] \mid l_m(Y).\overline{a}\langle Y\rangle),$$
$$N_1, \ldots, N_n, `Q_1, \ldots, `Q_m = `\mathit{run}(`\mathit{run}(\ldots `\mathit{run}(`a(X).\mathit{run}(X))\ldots)),\ n \geq 0\},$$
$$F = !\nu f.(f[a(X).\mathit{run}(X)] \mid f(Y).\overline{a}\langle Y\rangle).$$

The set of names $r$ and the environment share the same fate as those of Example 4 for identical reasons. For ease, we write *lhs* and *rhs* to conveniently denote each of the tested processes.

Verification of the bisimulation clauses of $\mathcal{X}_1$ is immediate and leads to a member $(r, \emptyset, \mathit{run}(`P), \nu a.(\overline{a}\langle `P\rangle \mid F))$ of $\mathcal{X}_2$ for some $`P$ with $(`P, `P) \in (\emptyset; r)^\star$. For $\mathcal{X}_2$, *lhs* can do an internal action (consuming its outer $\mathit{run}(\ `)$) that *rhs* does not have to follow since we work with weak bisimulations, and the results is still in $\mathcal{X}_2$; conversely, internal actions of *rhs* do not have to be matched. Some of those transitions that *rhs* can do are

reactions between replications from $F$. All those transitions creates elements of either $R_1$ or $R_2$ that can do nothing but internal actions and can be ignored further in the proof thanks to the weakness of our bisimulations.

Whenever *lhs* does an observable action $\alpha$, that is, when $P_1 = P \xrightarrow{\alpha} P'$, *rhs* must do a reaction between $\overline{a}\langle \text{`}P_2 \rangle$ and $F$, giving $\nu l.(l[P_2] \,|\, l(Y).\overline{a}\langle Y \rangle) \stackrel{\alpha}{\Rightarrow} \nu l.(l[P'] \,|\, l(Y).\overline{a}\langle Y \rangle)$ which satisfies $\mathcal{X}_3$'s definition. Moreover, all transitions of $P_1$ or $P_2$ in $\mathcal{X}_3$ can be matched by the other, hence preserving the membership in $\mathcal{X}_3$. Finally, a subprocess $\nu l.(l[P_2] \mid l(Y).\overline{a}\langle Y \rangle)$ of *rhs* of $\mathcal{X}_3$ can do a $\tau$ transition to $\overline{a}\langle \text{`}P_2 \rangle$ and the residues belong back to $\mathcal{X}_2$.

This concludes the proof of behavioural equivalence of the original processes $c(X).\,\mathit{run}(X)$ and $c(X).\nu a.(\overline{a}\langle X \rangle.!\nu f.(f[a(X).\mathit{run}(X)] \mid f(Y).f[\mathit{run}(Y)]))$.

## 5 Discussion and Future Work

In the original higher-order $\pi$-calculus with passivation described by Lenglet *et al.* [7], terms are identified with processes: its syntax is just $P ::= 0 \mid X \mid a(X).P \mid \overline{a}\langle P \rangle.P \mid (P \,|\, P) \mid a[P] \mid \nu a.P \mid !P$. We conjecture that it is also possible to develop sound environmental bisimulations (and up-to context, etc.) for this version of HO$\pi$P, as we [12] did for the standard higher-order $\pi$-calculus. However we chose not to cover directly the original higher-order $\pi$-calculus with passivation, for two reasons: (1) the proof method of [12] which relies on guarded processes and a factorisation trick using the spawning clause of the bisimulation is inadequate in the presence of locations; (2) there is a very strong constraint in clause 4 of up-to context in [12, Definition E.1 (Appendix)] (the context has no hole for terms from $\mathcal{E}$). By distinguishing processes from terms, not only is our up-to context method much more general, but our proofs are also direct and technically simple. Although one might argue that the presence of the *run* operator is a burden, by using Definition 3, one could devise an "up-to *run*" technique and abstract $\mathit{run}(\ldots \text{`}\mathit{run}(\text{`}P))$ as $P$, making equivalence proofs easier to write and understand.

As described in Remark 1, removing the limitation on the environments is left for future work. We also plan to apply environmental bisimulations to (a substantial subset of) the Kell calculus so that we can provide a practical alternative to context bisimulations in a more expressive higher-order distributed process calculus. In the Kell calculus, locations are not transparent: one discriminates messages on the grounds of their origins (i.e. from a location above, below, or from the same level). For example, consider the (simplified) Kell processes $P = \overline{a}\langle M \rangle.!b[\overline{a}]$ and $Q = \overline{a}\langle N \rangle.!b[\overline{a}]$ where $M = \overline{a}$ and $N = 0$. They seem bisimilar assuming environmental bisimulations naively like those in this paper: intuitively, both $P$ and $Q$ can output (respectively $M$ and $N$) to channel $a$, and their continuations are identical; passivation of spawned $l[M]$ and $l[N]$ for known location $l$ would be immediately matched; finally, the output to channel $a$ under $l$, turning $P$'s spawned $l[M]$ into $l[0]$, could be matched by an output to $a$ under $b$ by $Q$'s replicated $b[\overline{a}]$. However, $M$ and $N$ behave differently when observed from the same level (or below), for example as in $l[M \mid a(Y).\overline{ok}]$ and $l[N \mid a(Y).\overline{ok}]$ even under the presence of $!b[\overline{a}]$. More concretely, the context $[\cdot]_1 \,|\, a(X).c[X \,|\, a(Y).\overline{ok}]$ distinguishes $P$ and $Q$, showing the unsoundness of such naive definition. This suggests that, to define sound environmental bisimulations in Kell-like calculi with non-transparent locations,

we should require a stronger condition such as bisimilarity of $M$ and $N$ in the output clause. Developments on this idea are in progress.

## References


1. Cardelli, L., Gordon, A.D.: Mobile ambients. In: Nivat, M. (ed.) FOSSACS 1998. LNCS, vol. 1378, pp. 140–155. Springer, Heidelberg (1998)
2. Hennessy, M., Riely, J.: Resource access control in systems of mobile agents. Information and Computation 173, 82–120 (2002)
3. Hewlett-Packard: Live migration across data centers and disaster tolerant virtualization architecture with HP storageworks cluster extension and Microsoft Hyper-V, `http://h20195.www2.hp.com/V2/GetPDF.aspx/4AA2-6905ENW.pdf`
4. Hildebrandt, T., Godskesen, J.C., Bundgaard, M.: Bisimulation congruences for Homer: a calculus of higher-order mobile embedded resources. Technical Report TR-2004-52, IT University of Copenhagen (2004)
5. Honda, K., Yoshida, N.: On reduction-based process semantics. Theoretical Computer Science 151(2), 437–486 (1995)
6. Howe, D.J.: Proving congruence of bisimulation in functional programming languages (1996)
7. Lenglet, S., Schmitt, A., Stefani, J.-B.: Normal bisimulations in calculi with passivation. In: de Alfaro, L. (ed.) FOSSACS 2009. LNCS, vol. 5504, pp. 257–271. Springer, Heidelberg (2009)
8. Milner, R.: Communicating and Mobile Systems: the Pi-Calculus. Cambridge University Press, Cambridge (1999)
9. Sangiorgi, D.: Expressing Mobility in Process Algebras: First-Order and Higher-Order Paradigms. PhD thesis, University of Edinburgh (1992)
10. Sangiorgi, D.: Bisimulation for higher-order process calculi. Information and Computation 131, 141–178 (1996)
11. Sangiorgi, D.: The $\pi$-calculus: a Theory of Mobile Processes. Cambridge University Press, Cambridge (2001)
12. Sangiorgi, D., Kobayashi, N., Sumii, E.: Environmental bisimulations for higher-order languages. In: Proceedings of the Twenty-Second Annual IEEE Symposium on Logic in Computer Science, pp. 293–302 (2007)
13. Sato, N., Sumii, E.: The higher-order, call-by-value applied pi-calculus. In: Hu, Z. (ed.) APLAS 2009. LNCS, vol. 5904, pp. 311–326. Springer, Heidelberg (2009)
14. Schmidt, D., Dhawan, P.: Live migration with Xen virtualization software, `http://www.dell.com/downloads/global/power/ps2q06-20050322-Schmidt-OE.pdf`
15. Schmitt, A., Stefani, J.-B.: The kell calculus: A family of higher-order distributed process calculi. In: Priami, C., Quaglia, P. (eds.) GC 2004. LNCS, vol. 3267, pp. 146–178. Springer, Heidelberg (2005)
16. Sumii, E., Pierce, B.C.: A bisimulation for dynamic sealing. Theoretical Computer Science 375(1-3), 169–192 (2007); Extended abstract appeared in Proceedings of 31st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, pp. 161–172 (2004)
17. Sumii, E., Pierce, B.C.: A bisimulation for type abstraction and recursion. Journal of the ACM 54, 1–43 (2007); Extended abstract appeared in Proceedings of 32nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, pp. 63–74 (2005)