Howe's Method for Contextual Semantics

Sergueï Lenglet, Alan Schmitt

▶ **To cite this version:**

Sergueï Lenglet, Alan Schmitt. Howe's Method for Contextual Semantics. CONCUR 2015 26th International Conference on Concurrency Theory, Sep 2015, Madrid, Spain. 10.4230/LIPIcs.CONCUR.2015.212 . hal-01192699

HAL Id: hal-01192699
https://inria.hal.science/hal-01192699v1

Submitted on 3 Sep 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Howe's Method for Contextual Semantics*

Sergueï Lenglet[1] **and Alan Schmitt**[2]

1 **Université de Lorraine, France**
serguei.lenglet@univ-lorraine.fr
2 **Inria, France**
alan.schmitt@inria.fr

Abstract

We show how to use Howe's method to prove that context bisimilarity is a congruence for process calculi equipped with their usual semantics. We apply the method to two extensions of HOπ, with passivation and with join patterns, illustrating different proof techniques.

1998 ACM Subject Classification F.1.1 Models of Computation

Keywords and phrases Bisimulations, process calculi, Howe's Method

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2015.212

1 Introduction

Process equivalence relates processes whose behavior may not be distinguished, even when inserted in arbitrary contexts. Equivalent processes may thus be used interchangeably in any larger system, with no observable difference. This property is quite strong, and to prove it directly, one has to consider every possible context. Much effort has thus been applied to techniques that simplify the proofs of process equivalence. Such techniques often involve the definition of a relation between processes that is easier to establish. The relation, typically a form of *bisimilarity*, is then shown to characterize process equivalence. This characterization has two parts: bisimilarity is *sound* – bisimilar processes are equivalent – and *complete* – equivalent processes are bisimilar.

As process equivalence is generally intended to be preserved by every context, it is often a congruence. Hence a sound and complete bisimilarity also has to be a congruence. Even when considering sound (but not complete) bisimilarities, it is very convenient that they be congruences. Indeed, to prove that two processes are equivalent, one can then simply show they have the same external structure (context) with bisimilar processes inside. Proving congruence is thus a crucial step when working with process equivalence.

Howe's method [7] is a powerful approach to show that a bisimilarity is a congruence. In a nutshell, it reverses the problem: first define a relation, called "Howe's closure", that includes the bisimilarity of interest and is a congruence by definition. Second, show it is a bisimulation. As bisimilarity contains every bisimulations, Howe's closure is thus included in bisimilarity. Third, conclude that the bisimilarity and its Howe's closure coincide, thus the former is a congruence.

This approach works well in a functional setting. Until now, its application to higher-order process calculi has required significant adjustments, either yielding a sound but not complete bisimilarity [5], or requiring the definition of a new semantics [11]. We present a direct

* This work has been partially supported by the ANR project 2010-BLAN-0305 PiCoq and the PHC Polonium project No. 33271XH.

© Sergueï Lenglet and Alan Schmitt;
licensed under Creative Commons License CC-BY
26th International Conference on Concurrency Theory (CONCUR 2015).
Editors: Luca Aceto and David de Frutos Escrig; pp. 212–225

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

application of Howe's method for the higher-order π calculus (HOπ) with its usual semantics, and state the central *pseudo-simulation* property that enables the application of the method (Section 2). We then detail two approaches to prove this lemma for two extensions of HOπ: one with *passivation* (Section 3), the other with *join-patterns* (Section 4). The complete proofs are available in an accompanying research report [10].

2 Howe's Method in HOπ with Contextual Semantics

2.1 Syntax and Contextual Semantics

We recall the syntax and contextual semantics of (the process-passing fragment of) HOπ [14] in Figure 1, omitting the symmetric rules for Par and HO. We use a, b, c to range over channel names, $\overline{a}$, $\overline{b}$, $\overline{c}$ to range over conames, γ to range over names and conames, and X, Y to range over process variables. We define $\overline{\overline{a}}$ as a. Multisets $\{x_1 \dots x_n\}$ (where x ranges over some entities) are written $\widetilde{x}$. Finally, we write $\uplus$ for multiset union.

An input $a(X)P$ binds X in P, and a restriction $\nu a.P$ binds a in P. We write $\mathrm{fv}(P)$ for the free variables of a process P and $\mathrm{fn}(P)$ for its free names. A *closed process* has no free variable. We identify processes up to α-conversion of names and variables: processes and agents are always chosen such that their bound names and variables are pairwise distinct, and distinct from their free names and variables. We write $P\{Q/X\}$ for the capture-free substitution of X by Q in P. *Structural congruence* $\equiv$ equates processes up to reorganization of their sub-processes and their name restrictions; it is the smallest congruence verifying the rules of Figure 1. Because the ordering of restrictions does not matter, we abbreviate $\nu a_1 \dots \nu a_n.P$ as $\nu\widetilde{a}.P$; since bound names are pairwise distinct, $\widetilde{a}$ is a set.

We define a labeled transition system (LTS), where agents transition to processes, *abstractions* F of the form $(X)Q$, or *concretions* C of the form $\nu\widetilde{b}.\langle R\rangle S$. Like for processes, the ordering of restrictions does not matter for a concretion, therefore we write them using a set of names $\widetilde{b}$; in particular, we write $\langle R\rangle S$ if $\widetilde{b} = \emptyset$. Labels of the LTS are ranged over by α. Transitions are either an *internal action* $P \xrightarrow{\tau} P'$, a *message input* $P \xrightarrow{a} F$, or a *message output* $P \xrightarrow{\overline{a}} C$. The transition $P \xrightarrow{a} (X)Q$ means that P may receive a process R on a to continue as $Q\{R/X\}$. The transition $P \xrightarrow{\overline{a}} \nu\widetilde{b}.\langle R\rangle S$ means that P may send the process R on a and then continue as S, and the scope of the names $\widetilde{b}$ has to be expanded to encompass the recipient of R. A higher-order communication takes place when a concretion interacts with an abstraction (rule HO).

2.2 Behavioral Equivalences

Barbed congruence relates processes based on their observable actions, or *barbs*. The observable actions γ of a process P, written $P \downarrow_\gamma$, are unrestricted names or conames on which a communication may immediately occur ($P \xrightarrow{\gamma} A$, for some A). A context $\mathbb{C}$ is a term with a single hole $\Box$, that may be filled with a process P, written $\mathbb{C}\{P\}$; the free names or free variables of P may be captured by $\mathbb{C}$. An equivalence relation $\mathcal{R}$ is a *congruence* if $P\,\mathcal{R}\,Q$ implies $\mathbb{C}\{P\}\,\mathcal{R}\,\mathbb{C}\{Q\}$ for all contexts $\mathbb{C}$.

▶ **Definition 1.** A symmetric relation $\mathcal{R}$ on closed processes is a strong barbed bisimulation if $P\,\mathcal{R}\,Q$ implies:

- $P \downarrow_\gamma$ implies $Q \downarrow_\gamma$;
- if $P \xrightarrow{\tau} P'$, then there exists Q' such that $Q \xrightarrow{\tau} Q'$ and $P'\,\mathcal{R}\,Q'$.

Syntax: $P ::= \mathbf{0} \mid X \mid P|P \mid a(X)P \mid \overline{a}\langle P\rangle P \mid \nu a.P$

Agents: $A, B ::= P \mid F \mid C$

Abstractions $F, G ::= (X)P$ Concretions $C, D ::= \langle P\rangle Q \mid \nu a.C$

Extension of operators to abstractions and concretions

$$(X)Q|P \triangleq (X)(Q|P) \text{ if } X \notin \mathrm{fv}(P)$$
$$P|(X)Q \triangleq (X)(P|Q) \text{ if } X \notin \mathrm{fv}(P)$$
$$\nu a.(X)P \triangleq (X)\nu a.P$$
$$(\nu\widetilde{b}.\langle Q\rangle R)|P \triangleq \nu\widetilde{b}.\langle Q\rangle(R|P) \text{ if } \widetilde{b} \cap \mathrm{fn}(P) = \emptyset$$
$$P|(\nu\widetilde{b}.\langle Q\rangle R) \triangleq \nu\widetilde{b}.\langle Q\rangle(P|R) \text{ if } \widetilde{b} \cap \mathrm{fn}(P) = \emptyset$$
$$\nu a.(\nu\widetilde{b}.\langle Q\rangle R) \triangleq \nu a,\widetilde{b}.\langle Q\rangle R \text{ if } a \in \mathrm{fn}(\nu\widetilde{b}.Q)$$
$$\nu a.(\nu\widetilde{b}.\langle Q\rangle R) \triangleq \nu\widetilde{b}.\langle Q\rangle \nu a.R \text{ if } a \notin \mathrm{fn}(\nu\widetilde{b}.Q)$$

Pseudo-application and process application

$$(X)P \bullet \nu\widetilde{b}.\langle R\rangle Q \triangleq \nu\widetilde{b}.(P\{R/X\}|Q) \text{ if } \widetilde{b} \cap \mathrm{fn}(P) = \emptyset \qquad (X)P \circ Q \triangleq P\{Q/X\}$$

Structural congruence

$$(P|Q)|R \equiv P|(Q|R) \qquad P|Q \equiv Q|P \qquad P|\mathbf{0} \equiv P \qquad \nu a.\nu b.P \equiv \nu b.\nu a.P$$
$$P|\nu a.Q \equiv \nu a.(P|Q) \text{ if } a \notin \mathrm{fn}(P) \qquad \nu a.P \equiv P \text{ if } a \notin \mathrm{fn}(P)$$

LTS rules: $\alpha ::= \tau \mid a \mid \overline{a}$

$$a(X)P \xrightarrow{a} (X)P \ \textsc{In} \qquad \overline{a}\langle Q\rangle P \xrightarrow{\overline{a}} \langle Q\rangle P \ \textsc{Out} \qquad \frac{P \xrightarrow{\alpha} A}{P|Q \xrightarrow{\alpha} A|Q} \ \textsc{Par}$$

$$\frac{P \xrightarrow{\alpha} A \qquad \alpha \notin \{a, \overline{a}\}}{\nu a.P \xrightarrow{\alpha} \nu a.A} \ \textsc{Restr} \qquad \frac{P \xrightarrow{a} F \qquad Q \xrightarrow{\overline{a}} C}{P|Q \xrightarrow{\tau} F \bullet C} \ \textsc{HO}$$

Figure 1 Contextual LTS for HOπ.

Two processes P, Q are strong barbed congruent, written $P \sim_b Q$, if for all context $\mathbb{C}$, there exists a strong barbed bisimulation $\mathcal{R}$ such that $\mathbb{C}\{P\}\ \mathcal{R}\ \mathbb{C}\{Q\}$.

A relation $\mathcal{R}$ is *sound* with respect to $\sim_b$ if $\mathcal{R} \subseteq \sim_b$; $\mathcal{R}$ is *complete* with respect to $\sim_b$ if $\sim_b \subseteq \mathcal{R}$. In [14], barbed congruence is characterized by a (strong) *context bisimilarity*, defined as follows.

▶ **Definition 2.** A relation $\mathcal{R}$ on closed processes is a context simulation if $P\ \mathcal{R}\ Q$ implies:
- for all $P \xrightarrow{\tau} P'$, there exists Q' such that $Q \xrightarrow{\tau} Q'$ and $P'\ \mathcal{R}\ Q'$;
- for all $P \xrightarrow{a} F$, for all C, there exists F' such that $Q \xrightarrow{a} F'$ and $F \bullet C\ \mathcal{R}\ F' \bullet C$;
- for all $P \xrightarrow{\overline{a}} C$, for all F, there exists C' such that $Q \xrightarrow{\overline{a}} C'$ and $F \bullet C\ \mathcal{R}\ F \bullet C'$.

A relation $\mathcal{R}$ is a context bisimulation if $\mathcal{R}$ and $\mathcal{R}^{-1}$ are context simulations. Context bisimilarity, written $\sim$, is the largest context bisimulation.

The definition is written in the *early* style, because the answer $Q \xrightarrow{a} F'$ depends on the particular C considered in the input case, and $Q \xrightarrow{\overline{a}} C'$ depends on F in the output case. In

the *late* style, this dependency is broken by moving the universal quantification on C or F after the existential one on F' or C'.

We extend the equivalences to open terms by defining the *open extension* of a relation $\mathcal{R}$.

▶ **Definition 3.** For two open processes P and Q, $P\ \mathcal{R}^\circ\ Q$ holds if $P\sigma\ \mathcal{R}\ Q\sigma$ holds for all process substitutions σ that close P and Q.

Conversely, we write $\mathcal{R}_c$ for the relation $\mathcal{R}$ restricted to closed processes.

In the following, we use (bi)simulation up to structural congruence, a (bi)simulation proof technique which allows to use $\equiv$ when relating processes.

▶ **Definition 4.** A relation $\mathcal{R}$ is a context simulation up to $\equiv$ if $P\ \mathcal{R}\ Q$ implies the clauses of Definition 2, where $\mathcal{R}$ is changed into $\equiv\mathcal{R}\equiv$.

Since $\equiv$ is a context bisimulation, the resulting proof technique is sound.

▶ **Lemma 5.** *If $\mathcal{R}$ is a context bisimulation up to $\equiv$, then $\mathcal{R} \subseteq \sim$.*

Context bisimilarity is sound and complete. The congruence proof of [14] does not apply, however, to certain process calculi, such as the ones with passivation [11]. For this reason, other congruence proof techniques, such as Howe's method [7], have been considered.

2.3 Howe's Method

We sketch the principles behind Howe's method and recall why its application to (early) context bisimilarity has been deemed problematic.

Howe's method [7, 6] is a systematic proof technique to show that a bisimilarity $\mathcal{B}$ (and its open extension $\mathcal{B}^\circ$) is a congruence. The method can be divided in three steps: first, prove some basic properties on the *Howe's closure* $\mathcal{B}^\bullet$ of the relation. By construction, $\mathcal{B}^\bullet$ contains $\mathcal{B}^\circ$ and is a congruence. Second, prove a simulation-like property for $\mathcal{B}^\bullet$. Finally, prove that $\mathcal{B}$ and $\mathcal{B}^\bullet$ coincide on closed processes. Since $\mathcal{B}^\bullet$ is a congruence, then so is $\mathcal{B}$.

Given a relation $\mathcal{R}$, Howe's closure is inductively defined as the smallest congruence which contains $\mathcal{R}^\circ$ and is closed under right composition with $\mathcal{R}^\circ$.

▶ **Definition 6.** Howe's closure $\mathcal{R}^\bullet$ of a relation $\mathcal{R}$ is defined inductively by the following rules, where op ranges over the operators of the language.

$$\frac{P\ \mathcal{R}^\circ\ Q}{P\ \mathcal{R}^\bullet\ Q} \qquad \frac{P\ \mathcal{R}^\bullet\ P' \qquad P'\ \mathcal{R}^\circ\ Q}{P\ \mathcal{R}^\bullet\ Q} \qquad \frac{\widetilde{P}\ \mathcal{R}^\bullet\ \widetilde{Q}}{\mathsf{op}(\widetilde{P})\ \mathcal{R}^\bullet\ \mathsf{op}(\widetilde{Q})}$$

Instantiating $\mathcal{R}$ as $\mathcal{B}$, $\mathcal{B}^\bullet$ is a congruence by definition. The composition with $\mathcal{B}^\circ$ enables some transitivity and additional properties. In particular, we can prove that $\mathcal{B}^\bullet$ is *substitutive*: if $P\ \mathcal{B}^\bullet\ Q$ and $R\ \mathcal{B}^\bullet\ S$, then $P\{R/X\}\ \mathcal{B}^\bullet\ Q\{S/X\}$. By definition, we have $\mathcal{B}^\circ \subseteq \mathcal{B}^\bullet$; for the reverse inclusion to hold, we prove that $\mathcal{B}^\bullet$ is a bisimulation, hence it is included in the bisimilarity. To this end, we first prove that $\mathcal{B}^\bullet$ (restricted to closed terms) is a simulation, using a pseudo-simulation lemma (second step of the method, discussed below). We then use the following result on the reflexive and transitive closure $(\mathcal{B}^\bullet)^*$ of $\mathcal{B}^\bullet$.

▶ **Lemma 7.** *Let $\mathcal{R}$ be an equivalence. Then $(\mathcal{R}^\bullet)^*$ is symmetric.*

If $\mathcal{B}^\bullet$ is a simulation, then $(\mathcal{B}^\bullet)^*$ (restricted to closed terms) is also a simulation. By Lemma 7, $(\mathcal{B}^\bullet)^*$ is in fact a bisimulation. Consequently, we have $\mathcal{B} \subseteq \mathcal{B}^\bullet \subseteq (\mathcal{B}^\bullet)^* \subseteq \mathcal{B}$ on closed terms, and we conclude that $\mathcal{B}$ is a congruence.

The main challenge is stating and proving a simulation-like property for the Howe's closure $\mathcal{B}^\bullet$ of a bisimilarity $\mathcal{B}$. The labels λ of a LTS $\xrightarrow{\lambda}$ of a higher-order language usually contain or depend on terms (e.g., in the λ-calculus, λ-abstractions are labels), so the technique generally extends $\mathcal{B}^\bullet$ to labels. The simulation-like property then follows the pattern below, similar to a higher-order bisimilarity clause as in Plain CHOCS [18].

If $P\ \mathcal{B}^\bullet\ Q$ and $P \xrightarrow{\lambda} A$, then for all $\lambda\ \mathcal{B}^\bullet\ \lambda'$, there exists B such that $Q \xrightarrow{\lambda'} B$ and $A\ \mathcal{B}^\bullet\ B$.

Stating and proving such a result for a Howe's closure built from an early context bisimilarity $\sim$, where inputs and outputs depend on respectively concretions and abstractions, is problematic. Indeed, we would like to prove that $P \sim^\bullet Q$ implies:

- for all $P \xrightarrow{a} F$, for all $C \sim^\bullet C'$, there exists F' such that $Q \xrightarrow{a} F'$ and $F \bullet C \sim^\bullet F' \bullet C'$;
- for all $P \xrightarrow{\overline{a}} C$, for all $F \sim^\bullet F'$ there exists C' such that $Q \xrightarrow{\overline{a}} C'$ and $F \bullet C \sim^\bullet F' \bullet C'$.

These clauses raise several issues. First, we have to find extensions of Howe's closure to abstractions and concretions which fit an early style. Even assuming such extensions, we cannot use this result to show $\sim^\bullet$ is a simulation. Indeed, suppose we are in the higher-order communication case: the processes are a parallel composition ($P = P_1 \mid P_2$, $Q = Q_1 \mid Q_2$, $P_1 \sim^\bullet Q_1$, and $P_2 \sim^\bullet Q_2$) and the transition is a higher-order communication ($P \xrightarrow{\tau} F \bullet C$, $P_1 \xrightarrow{a} F$, and $P_2 \xrightarrow{\overline{a}} C$). We thus need to find F' and C' such that $Q \xrightarrow{\tau} F' \bullet C'$, and $F \bullet C \sim^\bullet F' \bullet C'$. However, we cannot apply the input clause with $P_1 \sim^\bullet Q_1$: to have a F' such that $Q_1 \xrightarrow{a} F'$, we have to find first a concretion C' such that $C \sim^\bullet C'$. We cannot use the output clause with P_2 and Q_2 either: to have a C' such that $Q_2 \xrightarrow{\overline{a}} C'$, we have to find first an abstraction F' such that $F \sim^\bullet F'$. Taking $C \sim^\bullet C$ to obtain F' such that $F \bullet C \sim^\bullet F' \bullet C$, then $F' \sim^\bullet F'$ to yield C' and $F' \bullet C \sim^\bullet F' \bullet C'$ would not work either: to conclude we would need to show that $\sim^\bullet$ is transitive. Transitivity is the reason usual congruence proof techniques fail with weak bisimulations, and the very motivation to turn to Howe's method [11, Section 3.1]. As we cannot bypass this mutual dependency nor this transitivity requirement, the proof fails in the communication case.

In [5], the authors break the mutual dependency by partially dropping the early style: they write the output clause in the late style. The resulting *input-early* bisimilarity is complete in the strong case, but not in the weak case. In [11], we propose to make the output clause a little less early: instead of first requiring the abstraction to provide a matching output, we only require the process that does the reception – that reduces to the abstraction. This small change is sufficient to break the mutual dependency. Indeed, the concretion C' from Q_2 matching the $P_2 \xrightarrow{\overline{a}} C$ step depends only on P_1, which is known, and not on some unknown abstraction. We can then obtain the abstraction F' from Q_2 that matches the $P_1 \xrightarrow{a} F$ step. This abstraction depends fully on C', in the usual early style.

Unfortunately, we do not directly use abstractions and concretions in [11], we define instead a *complementary* LTS, and its bisimilarity. Such a LTS implements the change above as follows: when P sends a message to Q, this becomes a transition from P *using* Q *as a label*. As a result, in the corresponding bisimilarity, an output action depends on a process that performs the input instead of the input itself. The LTS we obtain is serialized compared to the contextual one: in a communication, we do not have two parallel derivation trees for the output and the input, as with rule HO, but a single one, where we first look for the output, and then look for the input. But creating such a complementary LTS can be difficult, especially to handle scope extrusion properly, as we observed with passivation [11]. In the next section, we show that we can in fact apply Howe's method with the regular LTS.

2.4 Congruence Proof Using Howe's Method

As explained in Section 2.3, the main challenge to apply Howe's method is stating and proving a pseudo-simulation lemma for the Howe's closure $\sim^\bullet$. With contextual semantics, the challenge is to avoid mutual dependencies between the input and output clauses. Following the main idea behind the complementary semantics, we propose to keep the usual LTS but change the definition of the pseudo-simulation property to make the output depend on a process performing an input, and not the input itself. Conversely, the input now depends on a process performing an output, and not the output itself. Formally, if $P_1 \sim^\bullet Q_1$, then

- for all $P_1 \xrightarrow{a} F_1$, for all $P_2 \sim^\bullet Q_2$ such that $P_2 \xrightarrow{\overline{a}} C_1$, there exist F_2, C_2, such that $Q_1 \xrightarrow{a} F_2$, $Q_2 \xrightarrow{\overline{a}} C_2$, and $F_1 \bullet C_1 \sim^\bullet F_2 \bullet C_2$;
- for all $P_1 \xrightarrow{\overline{a}} C_1$, for all $P_2 \sim^\bullet Q_2$ such that $P_2 \xrightarrow{a} F_1$, there exist F_2, C_2, such that $Q_1 \xrightarrow{\overline{a}} C_2$, $Q_2 \xrightarrow{a} F_2$, and $F_1 \bullet C_1 \sim^\bullet F_2 \bullet C_2$.

This definition offers two advantages. First, we do not have to define an extension of $\sim^\bullet$ to abstractions and concretions as we relate only processes. Second, the clauses for the input and the output are identical, exchanging only the roles of P_1 and P_2, and of Q_1 and Q_2. Therefore, we can capture the input and output clause as a single symmetric clause. This gives us the up-to $\equiv$ pseudo-simulation lemma we will prove for $\sim_c^\bullet$ (the restriction of $\sim^\bullet$ to closed processes).

▶ **Lemma 8** (Pseudo-Simulation Lemma). *Let $P_1 \sim_c^\bullet Q_1$ and $P_2 \sim_c^\bullet Q_2$. If $P_1 \xrightarrow{\overline{a}} C_1$ and $P_2 \xrightarrow{a} F_1$, then there exist C_2, F_2 such that $Q_1 \xrightarrow{\overline{a}} C_2$, $Q_2 \xrightarrow{a} F_2$, and $F_1 \bullet C_1 \equiv \sim_c^\bullet \equiv F_2 \bullet C_2$.*

With this formulation of the pseudo-simulation lemma, we easily dispatch the communication case. Suppose $P = P_1 \mid P_2$ and $Q = Q_1 \mid Q_2$ with $P_1 \sim_c^\bullet Q_1$ and $P_2 \sim_c^\bullet Q_2$. If $P \xrightarrow{\tau} F \bullet C$, with $P_1 \xrightarrow{a} F_1$ and $P_2 \xrightarrow{\overline{a}} C_1$, then we immediately have F_2, C_2 such that $Q \xrightarrow{\tau} F_2 \bullet C_2$ and $F_1 \bullet C_1 \equiv \sim_c^\bullet \equiv F_2 \bullet C_2$.

Lemma 8 can be proved in several ways, using either serialized inductions, or a simultaneous induction on $P_1 \sim_c^\bullet Q_1$ and $P_2 \sim_c^\bullet Q_2$. We discuss here the former, with proofs detailed in [10, Appendix A]. We then adapt this approach to a calculus with passivation (Section 3). The simultaneous induction approach is presented in Section 4 for a calculus with join patterns.

Using serialized inductions, we can start with $P_1 \sim_c^\bullet Q_1$ or with $P_2 \sim_c^\bullet Q_2$. Suppose we start with an induction on the sending processes $P_1 \sim_c^\bullet Q_1$. Most cases consist in using the induction hypothesis, followed by congruence properties of $\sim_c^\bullet$. There are two exceptions: (1) the base case $P_1 \sim Q_1$, and (2) the case $P_1 = \overline{a}\langle P_1^1 \rangle P_1^2$, $Q_1 = \overline{a}\langle Q_1^1 \rangle Q_1^2$, with $P_1^1 \sim_c^\bullet Q_1^1$ and $P_1^2 \sim_c^\bullet Q_1^2$. In these cases, we know which concretion C_2 the process Q_1 reduces to (either using $\sim$ in case (1), or by construction of P_1 and Q_1 in case (2)), but we have to find the abstraction F_2 the process Q_2 reduces to. To do so, we prove the following.

▶ **Lemma 9.** *Let $P_1^1 \sim_c^\bullet Q_1^1$ and $P_2 \sim_c^\bullet Q_2$ such that $P_2 \xrightarrow{a} F_1$. There exists F_2 such that $Q_2 \xrightarrow{a} F_2$, and $F_1 \circ P_1^1 \sim_c^\bullet F_2 \circ Q_1^1$.*

The proof of this lemma is by induction on the derivation of $P_2 \sim_c^\bullet Q_2$. Lemma 9 deals with case (2) directly (just add the continuations P_1^2 and Q_1^2 using congruence), but it also handles case (1) ($P_1 \sim Q_1$). Indeed, if R is the message of C_1, applying Lemma 9 with $P_1^1 = Q_1^1 = R$ gives $F_1 \circ R \sim_c^\bullet F_2 \circ R$, which implies $F_1 \bullet C_1 \sim_c^\bullet F_2 \bullet C_1$ by congruence of $\sim_c^\bullet$. Since $P_1 \sim Q_1$, there exists C_2 such that $Q_1 \xrightarrow{\overline{a}} C_2$, and $F_2 \bullet C_1 \sim F_2 \bullet C_2$. We therefore have $F_1 \bullet C_1 \sim_c^\bullet \sim F_2 \bullet C_2$, which implies $F_1 \bullet C_1 \sim_c^\bullet F_2 \bullet C_2$ by right transitivity with $\sim$.

Alternatively, we can prove Lemma 8 by starting with the induction on the receiving processes $P_2 \sim_c^\bullet Q_2$. To handle the two cases (3) $P_2 \sim Q_2$ and (4) $P_2 = a(X)P$, $Q_2 = a(X)Q$, $P \sim^\bullet Q$, we need the following result.

▶ **Lemma 10.** *Let $P \sim^\bullet Q$ such that $fv(P) \cup fv(Q) \subseteq \{X\}$, and $P_1 \sim_c^\bullet Q_1$ such that $P_1 \xrightarrow{\overline{a}} C_1$. There exists C_2 such that $Q_1 \xrightarrow{\overline{a}} C_2$ and $(X)P \bullet C_1 \equiv \sim_c^\bullet \equiv (X)Q \bullet C_2$.*

▶ Remark. Lemmas 8 and 10 are defined up to $\equiv$ while Lemma 9 is not. Structural congruence is needed to move name restriction: suppose we have $P_1 \sim_c^\bullet Q_1$, $\nu b.P_2 \sim_c^\bullet \nu b.Q_2$, with $P_2 \sim_c^\bullet Q_2$, $P_1 \xrightarrow{a} F_1$, and $\nu b.P_2 \xrightarrow{\overline{a}} \nu b.C_2$ (which comes from $P_2 \xrightarrow{\overline{a}} C_2$). Using the induction with P_1, Q_1, P_2, and Q_2, there exist F_2 and C_2 such that $Q_1 \xrightarrow{a} F_2$, $Q_2 \xrightarrow{\overline{a}} C_2$, and $F_1 \bullet C_1 \sim_c^\bullet F_2 \bullet C_2$. We also have $\nu b.Q_2 \xrightarrow{\overline{a}} \nu b.C_2$. Note that, by our convention on bound names, b is neither in F_1 nor in F_2.

We want to prove $F_1 \bullet (\nu b.C_1) \sim_c^\bullet F_2 \bullet (\nu b.C_2)$, but from $F_1 \bullet C_1 \sim_c^\bullet F_2 \bullet C_2$, we can deduce $\nu b.(F_1 \bullet C_1) \sim_c^\bullet \nu b.(F_2 \bullet C_2)$ by congruence of $\sim_c^\bullet$. Depending on whether the scope of b has to be extended or not, it is not the same as $F_1 \bullet (\nu b.C_1) \sim_c^\bullet F_2 \bullet (\nu b.C_2)$; at best, we have $F_1 \bullet (\nu b.C_1) \equiv \nu b.(F_1 \bullet C_1) \sim_c^\bullet \nu b.(F_2 \bullet C_2) \equiv F_2 \bullet (\nu b.C_2)$, hence the need for $\equiv$. We do not have this issue in Lemma 9, since only messages, and not concretions, are involved.

For $\sim_c^\bullet$ to be a simulation, we have to prove the following result on τ-actions (by induction on the derivation of $P \sim_c^\bullet Q$), using Lemma 8 in the communication case.

▶ **Lemma 11.** *If $P \sim_c^\bullet Q$ and $P \xrightarrow{\tau} P'$, then there exists Q' such that $Q \xrightarrow{\tau} Q'$ and $P' \equiv \sim_c^\bullet \equiv Q'$.*

We can then prove $\sim_c^\bullet$ is a simulation up to $\equiv$. Suppose $P \sim_c^\bullet Q$. If $P \xrightarrow{a} F$, then for all $C = \nu\widetilde{b}.\langle R\rangle S$, we apply Lemma 8 with $P_2 = P$, $Q_2 = Q$, and $P_1 = Q_1 = \nu\widetilde{b}.\overline{a}\langle R\rangle S$. This yields an F' such that $Q \xrightarrow{a} F'$ and $F \bullet C \equiv \sim_c^\bullet \equiv F' \bullet C$. Similarly, if $P \xrightarrow{\overline{a}} C$, then for all $F = (X)R$, we apply Lemma 8 with $P_1 = P$, $Q_1 = Q$, and $P_2 = Q_2 = a(X)R$. We can then deduce that $(\equiv \sim_c^\bullet \equiv)^*$ is a bisimulation, and finally conclude $\sim = \equiv \sim_c^\bullet \equiv$, as explained in Section 2.3. Since $\equiv \sim_c^\bullet \equiv$ is a congruence, then $\sim$ is a congruence.

3 Application to a Calculus with Passivation

3.1 The HOπP Calculus

HOπP [11] extends HOπ with passivation, an operation that may stop a running process and capture its state. The granularity of passivation is the *locality* $a[P]$, a new construct added to the syntax of HOπ. The semantics of $a[P]$ is as follows: P can freely reduce and communicate with any other process; it may also be captured at any time by a process $a(X)R$, substituting its contents P for X in R. Formally, we extend the locality construct to all agents, and we add the rules Loc and Passiv to the LTS of Figure 1.

$$a[(X)P] \triangleq (X)a[P] \qquad a[\nu\widetilde{b}.\langle P\rangle Q] \triangleq \nu\widetilde{b}.\langle P\rangle a[Q] \text{ if } a \notin \widetilde{b}$$

$$a[P] \xrightarrow{\overline{a}} \langle P\rangle 0 \ \textsc{Passiv} \qquad \frac{P \xrightarrow{\alpha} A}{a[P] \xrightarrow{\alpha} a[A]} \ \textsc{Loc}$$

The rule Loc and the definition of $a[C]$ imply that the scope of restricted names may cross locality boundaries, but structural congruence is left unchanged. In particular, $\nu b.a[P]$ is not congruent to $a[\nu b.P]$. Indeed, the combination of lazy scope extrusion and passivation may generate two distinct behaviors from these terms. See [11, Section 2.3] for more details.

3.2 Context Bisimilarity

The definition of context bisimulation is more complex in HOπP than in HOπ because of the discriminating power added by passivation. We briefly explain the differences; more details and examples can be found in [11, Section 2.4]. First, we can distinguish between processes with different free names using passivation and lazy scope extrusion [2]. Indeed, suppose a is free in P but not in Q, and consider the context $b[\nu a.\overline{c}\langle\Box\rangle R]$. Then a communication on c extends the scope of a outside b for P but not for Q, which gives us processes of the form $\nu a.(b[R] \mid P')$ and $b[\nu a.R] \mid Q'$ for some P' and Q'. If we then capture the locality b and duplicate its content, we obtain $\nu a.(R \mid R \mid P')$ in one case, and $(\nu a.R) \mid (\nu a.R) \mid Q'$ in the other: for the first process, a is shared, but not for the second one, and by choosing R accordingly, we obtain different behavior. Therefore, two processes P and Q are equivalent only if $\mathrm{fn}(P) = \mathrm{fn}(Q)$.

Next, when a message is sent outside a locality, the continuation stays in the locality (by definition of $a[C]$). The continuation can then be put into a completely different context using passivation. As a result, the message and its continuation may end up in different contexts, but still share a common information (the extruded names). To be able to express this situation specific to calculi with passivation, we introduce *bisimulation contexts* $\mathbb{E}$, i.e., evaluation contexts used for observational purposes.

$$\mathbb{E} ::= \Box \mid \nu a.\mathbb{E} \mid \mathbb{E} \mid P \mid P \mid \mathbb{E} \mid a[\mathbb{E}]$$

Instead of comparing $F \bullet C$ with $F \bullet C'$ in the output case, we now compare $F \bullet \mathbb{E}\{C\}$ with $F \bullet \mathbb{E}\{C'\}$. The extra context $\mathbb{E}$ represents the potential passivation of the continuations of C and C'. The definition of context bisimulation for HOπP is then as follows.

▶ **Definition 12.** A relation $\mathcal{R}$ on closed processes is a context simulation if $P \mathcal{R} Q$ implies $\mathrm{fn}(P) = \mathrm{fn}(Q)$ and:

- for all $P \xrightarrow{\tau} P'$, there exists Q' such that $Q \xrightarrow{\tau} Q'$ and $P' \mathcal{R} Q'$;
- for all $P \xrightarrow{a} F$, for all C, there exists F' such that $Q \xrightarrow{a} F'$ and $F \bullet C \mathcal{R} F' \bullet C$;
- for all $P \xrightarrow{\overline{a}} C$, for all F, $\mathbb{E}$, there exists C' such that $Q \xrightarrow{\overline{a}} C'$ and $F \bullet \mathbb{E}\{C\} \mathcal{R} F \bullet \mathbb{E}\{C'\}$.

A relation $\mathcal{R}$ is a context bisimulation if $\mathcal{R}$ and $\mathcal{R}^{-1}$ are context simulations. Context bisimilarity, written $\sim$, is the largest context bisimulation.

The usual approach to prove soundness of $\sim$ consists in proving that its transitive and congruence closure is a context bisimulation. This proof technique does not carry to the weak case. In [11], we prove soundness of a weak complementary bisimilarity, which coincides with a weak variant of $\sim$, by defining a weak complementary LTS for HOπP, with elaborate labels and subtle side-conditions in the LTS rules to handle lazy scope extrusion. The resulting LTS has almost twice as many rules as the contextual one.

We show here how to directly apply Howe's method with the contextual semantics, as in HOπ. We give these results for the strong bisimilarity $\sim$ to ease the presentation; the proofs for the weak case are in [10, Appendix B]. As usual when adapting Howe's method to calculi with passivation [5, 11], we have to extend Howe's closure to bisimulation contexts. We define $\mathbb{E}_1 \sim^\bullet \mathbb{E}_2$ as the smallest congruence satisfying the following rules.

$$\frac{\mathbb{E}_1 \sim^\bullet \mathbb{E}_2 \qquad P_1 \sim^\bullet P_2}{\mathbb{E}_1 \mid P_1 \sim^\bullet \mathbb{E}_2 \mid P_2} \qquad\qquad \frac{P_1 \sim^\bullet P_2 \qquad \mathbb{E}_1 \sim^\bullet \mathbb{E}_2}{P_1 \mid \mathbb{E}_1 \sim^\bullet P_2 \mid \mathbb{E}_2}$$

We can then write a pseudo-simulation lemma similar to Lemma 8, as follows.

▶ **Lemma 13** (Pseudo-Simulation Lemma). *Let* $P_1 \sim_c^\bullet Q_1$ *and* $P_2 \sim_c^\bullet Q_2$. *If* $P_1 \xrightarrow{\overline{a}} C_1$ *and* $P_2 \xrightarrow{a} F_1$, *then for all* $\mathbb{E}_1 \sim_c^\bullet \mathbb{E}_2$, *there exist* C_2, F_2 *such that* $Q_1 \xrightarrow{\overline{a}} C_2$, $P_2 \xrightarrow{a} F_2$, *and* $F_1 \bullet \mathbb{E}_1\{C_1\} \sim_c^\bullet F_2 \bullet \mathbb{E}_2\{C_2\}$.

Unlike the case with HOπ, we do not have a choice in the induction strategy for the proof of Lemma 13: we cannot prove it by doing first the induction on the derivation for the receiving processes $P_2 \sim_c^\bullet Q_2$. Indeed, suppose $F_1 \bullet \mathbb{E}_1\{C_1\} \sim_c^\bullet F_2 \bullet \mathbb{E}_2\{C_2\}$ holds for all $\mathbb{E}_1 \sim_c^\bullet \mathbb{E}_2$, and we want to prove $b[F_1] \bullet \mathbb{E}_1\{C_1\} \sim_c^\bullet b[F_2] \bullet \mathbb{E}_2\{C_2\}$. With congruence of $\sim_c^\bullet$, we can only deduce $b[F_1 \bullet \mathbb{E}_1\{C_1\}] \sim_c^\bullet b[F_2 \bullet \mathbb{E}_2\{C_2\}]$, and we cannot move the boundaries of b with $\equiv$. Therefore, when reasoning by induction on the receiving processes $P_2 \sim_c^\bullet Q_2$, we cannot apply the resulting abstractions F_1, F_2 to concretions. However, we can apply them to messages, as in the following lemma, identical to Lemma 9.

▶ **Lemma 14.** *Let* $P_1^1 \sim_c^\bullet Q_1^1$ *and* $P_2 \sim_c^\bullet Q_2$ *such that* $P_2 \xrightarrow{a} F_1$. *There exists* F_2 *such that* $Q_2 \xrightarrow{a} F_2$, *and* $F_1 \circ P_1^1 \sim_c^\bullet F_2 \circ Q_1^1$.

Indeed, if $F_1 \circ P_1^1 \sim_c^\bullet F_2 \circ Q_1^1$, then $b[F_1 \circ P_1^1] \sim_c^\bullet b[F_2 \circ Q_1^1]$ by congruence of $\sim_c^\bullet$. We then prove Lemma 13 by induction on the derivation for the sending processes $P_1 \sim_c^\bullet Q_1$. We do not have problems with localities when doing the induction on the derivation of $P_1 \sim_c^\bullet Q_1$, thanks to the bisimulation contexts: if $F_1 \bullet \mathbb{E}_1\{C_1\} \sim_c^\bullet F_2 \bullet \mathbb{E}_2\{C_2\}$ holds for all $\mathbb{E}_1 \sim_c^\bullet \mathbb{E}_2$, then it also holds for $\mathbb{E}_1\{b[\Box]\} \sim_c^\bullet \mathbb{E}_2\{b[\Box]\}$, and we have $F_1 \bullet \mathbb{E}_1\{b[C_1]\} \sim_c^\bullet F_2 \bullet \mathbb{E}_2\{b[C_2]\}$, as wished. Note that it also implies $F_1 \bullet \mathbb{E}_1\{\nu b.C_1\} \sim_c^\bullet F_2 \bullet \mathbb{E}_2\{\nu b.C_2\}$ by taking $\mathbb{E}_1\{\nu b.\Box\} \sim_c^\bullet \mathbb{E}_2\{\nu b.\Box\}$, therefore restriction poses no problem, and Lemma 13 is formulated without structural congruence, unlike Lemma 8. In addition to Lemma 13, we also prove a lemma similar to Lemma 11 for τ-actions, and then deduce that $\sim_c^\bullet$ is a simulation. We conclude as for HOπ.

Completeness. The strong and weak variants of the context bisimilarity $\sim$ coincide with respectively the strong and weak complementary bisimilarities of [11], which are themselves complete (see [11, Section 5.2]). Consequently, the strong and weak context bisimilarities are also complete.

4 Application to a Calculus with Join Patterns

4.1 Syntax and Semantics

Join patterns allow several messages to be received at once by the same process. The syntax of HOπJ is given in Figure 2. We replace the receiving process $a(X)P$ of HOπ by a process $\pi \rhd P$, where π is a join pattern $a_1(X_1) | \ldots | a_n(X_n)$. A higher-order communication takes place when messages are available simultaneously on the names $a_1 \ldots a_n$. We write $\prod_{i\in\{1..n\}} x_i$ or $\prod \widetilde{x}$ (where x ranges over some entity) for the parallel composition $x_1 | \ldots | x_n$ if $n > 1$, or for simply x_1 if $n = 1$. We also abbreviate $\pi = a_1(X_1) \mid \ldots \mid a_n(X_n)$ as $\prod \widetilde{a(X)}$. The syntax of abstractions is changed accordingly ($F \triangleq (\pi)P$), and concretions now accumulate the messages of several emitting processes in parallel. A concretion is of the form $\nu\widetilde{b}.\langle a_1, P_1\rangle \ldots \langle a_n, P_n\rangle Q$, meaning that each process P_i is sent on the name a_i, and the scope of the names $\widetilde{b}$ has to be extended to encompass the recipient of the messages. We abbreviate $\nu\widetilde{b}.\langle a_1, P_1\rangle \ldots \langle a_n, P_n\rangle Q$ as $\nu\widetilde{b}.\langle \widetilde{a, P}\rangle Q$.

The semantics of HOπJ is given by the LTS rules of Figure 2, where the symmetric of rules Par, HO, and Part-HO are omitted. An input $P \xrightarrow{\widetilde{a}} F$ is labelled with the multiset $\widetilde{a}$

Syntax: $P ::= \mathbf{0} \mid X \mid P \mid P \mid \nu a.P \mid \overline{a}\langle P\rangle P \mid \pi \triangleright P \qquad \pi ::= \pi \mid \pi \mid a(X)$

Agents: $F ::= (\pi)P \qquad C ::= D \mid \nu a.D \qquad D ::= \langle a, P\rangle Q \mid \langle a, P\rangle D$

Parallel composition of concretions

$$\nu\widetilde{b}.\langle\widetilde{a, R}\rangle P \mid \nu\widetilde{b'}.\langle\widetilde{a', R'}\rangle Q \triangleq \nu\widetilde{b} \cup \widetilde{b'}.\langle\widetilde{a, R} \uplus \widetilde{a', R'}\rangle(P \mid Q)$$
$$\text{if } \widetilde{b} \cap \mathrm{fn}(Q) = \widetilde{b'} \cap \mathrm{fn}(P) = \widetilde{b} \cap \widetilde{b'} = \emptyset$$

Structural congruence for join patterns

$$\pi_1 \mid \pi_2 \equiv \pi_2 \mid \pi_1 \qquad \pi_1 \mid (\pi_2 \mid \pi_3) \equiv (\pi_1 \mid \pi_2) \mid \pi_3$$

Pseudo-application

$$(\prod \widetilde{a(X)})P \bullet \nu\widetilde{b}.\langle\widetilde{a, R}\rangle Q \vdash \nu\widetilde{b}.(P\{\widetilde{R}/\widetilde{X}\} \mid Q) \text{ if } \widetilde{b} \cap \mathrm{fn}(P) = \emptyset$$
$$(\prod \widetilde{a(X)} \mid \pi)P \bullet \nu\widetilde{b}.\langle\widetilde{a, R}\rangle Q \vdash (\pi)\nu\widetilde{b}.(P\{\widetilde{R}/\widetilde{X}\} \mid Q) \text{ if } \widetilde{b} \cap \mathrm{fn}(P) = \emptyset$$

LTS rules: $\alpha_{\mathrm{j}} ::= \tau \mid \widetilde{a} \mid \widetilde{\overline{a}}$

$$\pi \triangleright P \xrightarrow{\widetilde{a}} (\pi)P \ \textsc{In} \qquad \overline{a}\langle Q\rangle P \xrightarrow{\overline{a}} \langle a, Q\rangle P \ \textsc{Out} \qquad \frac{P \xrightarrow{\alpha_{\mathrm{j}}} A}{P \mid Q \xrightarrow{\alpha_{\mathrm{j}}} A \mid Q} \ \textsc{Par}$$

$$\frac{P \xrightarrow{\widetilde{\overline{a}}} C_1 \qquad Q \xrightarrow{\widetilde{\overline{b}}} C_2}{P \mid Q \xrightarrow{\widetilde{\overline{a}} \uplus \widetilde{\overline{b}}} C_1 \mid C_2} \ \textsc{Par-Out} \qquad \frac{P \xrightarrow{\widetilde{a}} F \qquad Q \xrightarrow{\widetilde{\overline{a}}} C \qquad F \bullet C \vdash P'}{P \mid Q \xrightarrow{\tau} P'} \ \textsc{HO}$$

$$\frac{P \xrightarrow{\alpha_{\mathrm{j}}} A \qquad a \notin \alpha_{\mathrm{j}}}{\nu a.P \xrightarrow{\alpha_{\mathrm{j}}} \nu a.A} \ \textsc{Restr} \qquad \frac{P \xrightarrow{\widetilde{a} \uplus \widetilde{b}} F \qquad Q \xrightarrow{\widetilde{\overline{b}}} C \qquad \widetilde{a} \neq \emptyset \qquad F \bullet C \vdash F'}{P \mid Q \xrightarrow{\widetilde{a}} F'} \ \textsc{Part-HO}$$

Figure 2 Syntax and operational semantics of HOπJ.

of names on which messages are expected, and an output $P \xrightarrow{\widetilde{\overline{a}}} C$ is labelled by the multiset $\widetilde{\overline{a}}$ of conames on which messages are sent. Operators are extended to all agents as in HOπ, with the addition of parallel composition of concretions, to deal with the case where two processes P and Q in parallel reduce to C_1 and C_2. The parallel composition of C_1 and C_2 is defined as a concretion C which merges the messages and extruded names of C_1 and C_2, and composes in parallel their continuations (Figure 2, rule Par-Out).

A process P, receiving on names $\widetilde{a}$ (i.e., such that $P \xrightarrow{\widetilde{a}} (\pi)P'$), may communicate with a process Q emitting on names $\widetilde{b}$ (i.e., such that $Q \xrightarrow{\widetilde{\overline{b}}} C$) if $\widetilde{b} \subseteq \widetilde{a}$. We have two possible outcomes: either $\widetilde{b} = \widetilde{a}$ and the resulting agent is a process (rule HO), or $\widetilde{b} \subsetneq \widetilde{a}$ – some inputs of the join patterns are not filled with Q – and we obtain an abstraction (rule Part-HO). For instance, we have $\overline{a}\langle R\rangle\mathbf{0} \mid (a(X) \mid b(Y)) \triangleright P \xrightarrow{b} (b(Y))P\{R/X\}$. The definition of $\bullet$ in Figure 2 takes into account these two cases. Besides, the pseudo-application of an abstraction to a concretion may generate several results, depending on how the matching between the

outputs and the input is done. For instance, $\overline{a}\langle R_1\rangle\mathbf{0}\,|\,\overline{a}\langle R_2\rangle\mathbf{0}\,|\,(a(X)\,|\,a(Y))\triangleright P$ can reduce to either $P\{R_1/X\}\{R_2/Y\}$, or $P\{R_2/X\}\{R_1/Y\}$ (assuming R_1 and R_2 closed). Consequently, we write $\bullet$ as a predicate $F \bullet C \vdash P$ (respectively $F \bullet C \vdash F'$), meaning that P (respectively F') can be obtained as a result of the pseudo-application of F to C.

4.2 Context Bisimilarity

The definition of context bisimilarity for HOπJ is the same as for HOπ, adapted to the fact that $\bullet$ may generate several results for a given F and C.

▶ **Definition 15.** A relation $\mathcal{R}$ on closed processes is a context simulation if $P\ \mathcal{R}\ Q$ implies:

- for all $P \xrightarrow{\tau} P'$, there exists Q' such that $Q \xrightarrow{\tau} Q'$ and $P'\ \mathcal{R}\ Q'$;
- for all $P \xrightarrow{\widetilde{a}} F$, for all C, for all P' such that $F \bullet C \vdash P'$, there exist F', Q' such that $Q \xrightarrow{\widetilde{a}} F'$, $F' \bullet C \vdash Q'$, and $P'\ \mathcal{R}\ Q'$;
- for all $P \xrightarrow{\widetilde{\widetilde{a}}} C$, for all F, for all P' such that $F \bullet C \vdash P'$, there exist C', Q' such that $Q \xrightarrow{\widetilde{\widetilde{a}}} C'$, $F \bullet C' \vdash Q'$, and $P'\ \mathcal{R}\ Q'$.

A relation $\mathcal{R}$ is a context bisimulation if $\mathcal{R}$ and $\mathcal{R}^{-1}$ are context simulations. Context bisimilarity, written $\sim$, is the largest context bisimulation.

A similar context bisimulation has been defined for Kell [17], a higher-order calculus with passivation and join patterns. It is sound and complete in the strong case; the soundness proof of [17] does not rely on Howe's method, but instead shows that the reflexive, transitive, and congruence closure of the bisimilarity is itself a bisimulation. This direct method unfortunately does not scale to the weak case, as explained in [11]. Here, we prove that $\sim$ is a congruence using Howe's method. As in the previous section, even though we present the results in the strong case for simplicity, the complete proofs in [10, Appendix C] are for the weak case. To our knowledge, it is the first proof of soundness of a weak bisimilarity for a higher-order calculus with join patterns.

Bisimulation up to $\equiv$ is defined as in HOπ, by replacing $\mathcal{R}$ by $\equiv\mathcal{R}\equiv$ in the clauses. To prove that $\sim$ is sound with Howe's method, we use the following pseudo-simulation lemma.

▶ **Lemma 16** (Pseudo-Simulation Lemma). *Let $P \sim_c^\bullet Q$ and $\widetilde{R} \sim_c^\bullet \widetilde{R'}$ such that $P \xrightarrow{\widetilde{a}} F$, $R_i \xrightarrow{\widetilde{\widetilde{a}}_i} C_i$ for all i, $\widetilde{a} = \biguplus_i \widetilde{\widetilde{a}}_i$, and let P' such that $F \bullet \prod_i C_i \vdash P'$. Then there exist F', $\widetilde{C'}$, and Q' such that we have $Q \xrightarrow{\widetilde{a}} F'$, $R_i' \xrightarrow{\widetilde{\widetilde{a}}_i} C_i'$ for all i, $F' \bullet \prod_i C_i' \vdash Q'$, and $P' \equiv\sim_c^\bullet\equiv Q'$.*

We extend relations to multisets of same size in a pointwise way: $\widetilde{R} \sim_c^\bullet \widetilde{R'}$ means the two multisets are of the same size, and $R_i \sim_c^\bullet R_i'$ holds for every i. Note that Lemma 16 is a direct extension of Lemma 8 to multisets of sending processes; indeed, if we replace $\widetilde{R}$ and $\widetilde{R'}$ with single processes, we obtain the same formulation as Lemma 8 (with the exception that $\bullet$ is a predicate).

The proofs by serialization of Lemma 8, where we proceed by induction on the derivations for the the sender and then on the receiver (or conversely), do not apply to a calculus with join patterns, where a receiver communicates with several emitters – we cannot focus on a sender in particular, we have to consider them together. As a result, we consider another proof method, where we reason by induction on the derivations of $P \sim_c^\bullet Q$ and all the $\widetilde{R} \sim_c^\bullet \widetilde{R'}$ simultaneously. We distinguish two kinds of cases, depending on whether we need the induction hypothesis (detailed proofs are in [10, Appendix C]). Using the same definitions as in Lemmas 8 and 9, the cases where we do not need induction are those where each

$R_i \sim_c^\bullet R'_i$ verifies either (1) or (2) (bisimilar, or congruent outputs), and $P \sim_c^\bullet Q$ verifies either (3) or (4) (bisimilar, or congruent inputs). In these cases, we can conclude using substitutivity of $\sim_c^\bullet$ and the definition of $\sim$. The remaining cases are dealt with by using the induction hypothesis, and then congruence of $\sim_c^\bullet$ and $\equiv$. Again, we rely on structural congruence to change the scope of names when needed (we have the same issue as described in Remark 2.4).

Using Lemma 16, we can prove that $\sim_c^\bullet$ is a simulation up to $\equiv$, and then conclude that $\equiv\sim_c^\bullet\equiv = \sim$ as in HOπ.

Completeness. In [10, Appendix D], we prove that a weak variant of $\sim$ is complete, using the usual technique of [16]. We can prove completeness in the strong case with a similar proof.

▶ Remark. Proving Lemma 8 in HOπ is possible by reasoning simultaneously on $P_1 \sim_c^\bullet Q_1$ and $P_2 \sim_c^\bullet Q_2$, as described above. However, this method does not work for HOπP (Lemma 13) as pseudo-application and locality contexts do not commute (even up to structural congruence). One way to make the simultaneous induction works in calculi with passivation would be to add bisimulation contexts in the input clause, as follows:

- for all $P \xrightarrow{a} F$, for all C, there exists F' such that $Q \xrightarrow{a} F'$ and for all $\mathbb{E}$, we have $\mathbb{E}\{F\} \bullet C \mathrel{\mathcal{R}} \mathbb{E}\{F'\} \bullet C$.

With such a definition, we can prove soundness of the resulting bisimilarity in a calculus with passivation and join patterns (such as Kell) with the simultaneous induction. However, this extra use of bisimulation context adds complexity to the bisimulation. We conjecture they are not necessary in the input case.

5 Related Work

Howe's method in process calculi. Howe's method has been originally used to prove congruence in a lazy functional programming language [7]. Baldamus and Frauenstein [1] are the first to adapt the method to process calculi for variants of Plain CHOCS [18], and prove in particular the soundness of a weak late delay context bisimilarity. Hildebrandt and Godskesen [5] then adapt Howe's method for their calculus Homer, to prove the congruence of a (delay) input-early context bisimilarity (see Section 2.3). In [11], we use Howe's method to prove congruence of strong and weak complementary bisimilarities in HOπ and HOπP. The Howe's proof of [11] is somewhat similar to the serialized proof of Sections 2 and 3, except for the symmetric formulation of the pseudo-simulation lemma. However, there is no equivalent to the simultaneous induction proof of Section 4 in [11].

Bisimilarities in calculi with passivation. In addition to the context (or complementary) bisimilarities already discussed for Kell [17], Homer [5], and HOπP [11], *environmental bisimilarities* [15] have also been defined by Piérard and Sumii for calculi with passivation [12, 13]. Such relations compare P and Q using an environment $\mathcal{E}$, which represents the knowledge that an observer has about these processes, like the messages they have sent. The observer then uses $\mathcal{E}$ to challenge P and Q. For instance, the observer is able to compare inputs from P and Q with any messages built from the processes inside $\mathcal{E}$. In [12], the authors propose a sound weak environmental bisimilarity for HOπP. Their approach is not complete, seemingly because of the interplay between "by need" scope extrusion and passivation. In [13], they consider a variant of HOπP with name creation instead of name restriction, for which they define a sound and complete weak environmental bisimilarity. With name

creation, a name generated in a given locality becomes automatically known from the whole system. Name creation is therefore less expressive than name restriction with lazy scope extrusion, where we can control more finely the scope of generated names. In particular, it is not possible to implement internal choice or recursion using name creation, as shown in [8]. Finally, Koutavas and Hennessy recently developed a correct and complete symbolic bisimulation for a higher-order process calculus with passivation [8]. Their approach avoids the quantification over contexts at the cost of a more complex calculus, with local ports to recover the expressivity lost by using name creation.

Bisimilarities in calculi with join patterns. In [4], Fournet and Laneve define bisimilarities for the Join-Calculus, a first-order process calculus with join patterns. They define a weak bisimilarity which is sound w.r.t. the weak barbed congruence defined in [3], and also complete if name matching is added to the calculus. To our knowledge, only Kell [17] combines higher-order communication with join patterns. In [9], we define a weak complementary bisimilarity for Kell, which tests inputs by passing them messages one by one. This strategy requires processes to choose which input to perform without having all the necessary information (i.e., all the messages they are going to receive), and the resulting bisimilarity is therefore too discriminating (i.e., not complete).

6 Conclusion

In this paper, we showed how to directly use Howe's method to prove congruence properties of a context bisimilarity, without relying on an auxiliary relation such as complementary bisimilarity. We proposed a symmetric formulation of the pseudo-simulation lemma, which we can prove either with a serialized or with a simultaneous induction on the derivations for the emitting and receiving processes. The latter seems necessary in calculi with join patterns, while the former seems more appropriate for calculi with passivation. The resulting soundness proofs are much simpler than in complementary semantics [11], and they scale better to calculi with join patterns. Indeed, we compare receiving patterns by passing them several messages at once, and not only one by one as in the complementary case [9]. Finally, the bisimilarities of this paper are also complete in the weak case, unlike the input-early bisimilarity of [5], or the bisimilarity of [9] for join patterns. The use of Howe's method remains an open problem for calculi with both passivation and join patterns, such as Kell, if we do not want to make the definition of the bisimilarity more complex by using bisimulation contexts in the input case (see the remark at the end of Section 4).

References

1 Michael Baldamus and Thomas Frauenstein. Congruence proofs for weak bisimulation equivalences on higher–order process calculi. Technical report, Berlin University of Technology, 1995.

2 Giuseppe Castagna, Jan Vitek, and Francesco Zappa Nardelli. The Seal Calculus. *Information and Computation*, 201(1):1–54, 2005.

3 Cédric Fournet and Georges Gonthier. The reflexive cham and the join-calculus. In *POPL'96*, pages 372–385. ACM Press, 1996.

4 Cédric Fournet and Cosimo Laneve. Bisimulations in the join-calculus. *Theorectical Computer Science*, 266(1-2):569–603, 2001.

5 Jens C. Godskesen and Thomas Hildebrandt. Extending howe's method to early bisimulations for typed mobile embedded resources with local names. In *FSTTCS'05*, volume 3821 of *LNCS*, pages 140–151. Springer, 2005.

6 Andrew D. Gordon. Bisimilarity as a theory of functional programming. *Electronic Notes in Theoretical Computer Science*, 1:232–252, 1995.

7 Douglas J. Howe. Proving congruence of bisimulation in functional programming languages. *Information and Computation*, 124(2):103–112, 1996.

8 Vasileios Koutavas and Matthew Hennessy. Symbolic bisimulation for a higher-order distributed language with passivation. In *CONCUR'13*, pages 167–181. Springer-Verlag, 2013.

9 Sergueï Lenglet. *Bisimulations dans les calculs avec passivation*. PhD thesis, Université de Grenoble, 2010.

10 Sergueï Lenglet and Alan Schmitt. Howe's method for contextual semantics. Technical Report RR-8750, Inria, 2015.

11 Sergueï Lenglet, Alan Schmitt, and Jean-Bernard Stefani. Characterizing contextual equivalence in calculi with passivation. *Information and Computation*, 209(11):1390–1433, 2011.

12 Adrien Piérard and Eijiro Sumii. Sound bisimulations for higher-order distributed process calculus. In *FOSSACS'11*, volume 6604 of *LNCS*, pages 123–137. Springer, 2011.

13 Adrien Piérard and Eijiro Sumii. A higher-order distributed calculus with name creation. In *LICS'12*, pages 531–540. IEEE, 2012.

14 Davide Sangiorgi. Bisimulation for higher-order process calculi. *Information and Computation*, 131(2):141–178, 1996.

15 Davide Sangiorgi, Naoki Kobayashi, and Eijiro Sumii. Environmental bisimulations for higher-order languages. *ACM Transactions on Programming Languages and Systems*, 33(1), 2011.

16 Davide Sangiorgi and David Walker. *The Pi-Calculus: A Theory of Mobile Processes*. Cambridge University Press, 2001.

17 Alan Schmitt and Jean-Bernard Stefani. The Kell Calculus: A Family of Higher-Order Distributed Process Calculi. In *Global Computing 2004 workshop*, volume 3267 of *LNCS*, 2004.

18 Bent Thomsen. Plain chocs: A second generation calculus for higher order processes. *Acta Informatica*, 30(1):1–59, 1993.