

On Decidability of the Bisimilarity on Higher-order Processes with Parameterization*

Xian Xu

East China University of Science and Technology
xuxian@ecust.edu.cn

Wenbo Zhang

Shanghai Ocean University
Shanghai Key Laboratory of Trustworthy Computing
wbzhang@shou.edu.cn

Abstract Higher-order processes with parameterization are capable of abstraction and application (migrated from the lambda-calculus), and thus are computationally more expressive. For the minimal higher-order concurrency, it is well-known that the strong bisimilarity (i.e., the strong bisimulation equality) is decidable in absence of parameterization. By contrast, whether the strong bisimilarity is still decidable for parameterized higher-order processes remains unclear. In this paper, we focus on this issue. There are basically two kinds of parameterization: one on names and the other on processes. We show that the strong bisimilarity is indeed decidable for higher-order processes equipped with both kinds of parameterization. Then we demonstrate how to adapt the decision approach to build an axiom system for the strong bisimilarity. On top of these results, we provide an algorithm for the bisimilarity checking.

Keywords: Decidability, Strong bisimilarity, Parameterization, Higher-order, Processes

2000 MSC: 68Q05, 68Q10, 68Q85

1 Introduction

Bisimulation is a most important concept for comparing the behaviour of computing systems, particularly concurrent systems. An accompanying vital question is to check whether two given systems are equal in terms of bisimulation, hence the bisimilarity checking. Bisimilarity checking is an important topic in concurrency theory and formal verification. Basically there are two directions for this topic. One is to adopt an abstract manner, using process rewrite systems [13]. An advantage of this direction is that some core techniques can be extracted and potentially adapted to various models. The other is to work directly on concrete models [18]. An edge of this direction is that some well-defined operators can be harnessed thoroughly to guide the checking. We focus on the second direction in this work.

The bisimilarity checking, including checking bisimulation equalities, simulations, and preorders, has been attracting tremendous attention in the past few decades [9, 11–13, 23]. In contrast to the fruitful work of bisimilarity checking on first-order models, checking bisimulation equalities for higher-order processes has been more challenging. Much fewer results have been known in higher-order process models. Indeed, a major reason is that higher-order processes communicate in the fashion of process-passing (i.e., program-passing), and have the innate capability of encoding recursion. Besides, the standard bisimulation for higher-order processes, i.e., the context bisimulation, is strikingly different from those for first-order processes. It requires the matching of two output processes to be compared in arbitrary contexts. To this point, simplifying the context bisimilarity has also been a significant topic [19, 20].

*This work is supported by ANR 12IS02001 PACE and NSF of China (61872142, 61772336, 61572318, 61261130589).

To this day, the best known result of bisimilarity checking for higher-order processes is reported in [15], to our knowledge. In that work, Lanese et al. show that the strong bisimilarity checking of HOcore processes is decidable. As a matter of fact, they show that all known strong bisimilarities in HOcore are decidable. HOcore is a minimal higher-order process model that only has the input, (asynchronous) output, and the concurrency operator (i.e., the parallel composition). HOcore is also proven to be Turing complete, and this result is somehow refined toward a more implementable interpretation [3], in the manner of encoding lambda-calculus in HOcore through abstract machines.

That HOcore is Turing complete renders its process-termination problem undecidable. This fact adds to the contrast that the strong bisimilarity is decidable, which in turn implies the decidability of the barbed congruence. Technically, the decidability is achieved by showing all the strong bisimilarities to be coincident with a very special strong bisimilarity, called IO-bisimilarity, which is decidable by its definition in the first place. On the basis of this decidability outcome, a complete axiom system is also established, as well as an algorithm with acceptable complexity. It is then possible to implement the algorithm for bisimilarity checking HOcore processes in software systems [1]. Intuitively, the essential element making the strong bisimilarity decidable is that HOcore does not have the restriction operator, and thus the capability of expressing recursion is weakened. It is also shown that if restriction is recovered, i.e., if at least four static (i.e., top concurrency level) restrictions are included in HOcore, then the strong bisimilarity immediately becomes undecidable. The undecidability is proven through a reduction from the PCP; similar reductions are also used in other settings, e.g., the Ambient calculus [8]. Building upon [10, 15] further studies the possibility of making the termination decidable, in the setting of a fragment of HOcore where nested higher-order outputs are disabled. Specifically, it is shown that in such a setting the termination of processes becomes decidable (though convergence is still undecidable), due to the reason that the Minsky machines are no longer expressible. Technically, such decidability is achieved using the well-structured transition systems employed in [7]. Following [10, 18] shows that termination turns back to be undecidable if such a fragment of HOcore is enriched with a passivation operator [22], because Turing completeness is retained with the help of passivation.

In [6], Bundgaard et al. study the decidable fragments of Homer [5], a higher order process model with the capacity of expressing locations. They show that two subcalculi of Homer have decidable barbed bisimilarity, in both the strong and weak forms. Intuitively, Homer supports certain kind of pattern matching of name sequences that model the locations of resources, and this plays a central role in enhancing the expressiveness. For this reason, Homer can encode first-order processes and is computationally complete, leaving little hope for the decidability of bisimilarities. Therefore, to obtain decidability, some constraints have to be devised. Technically, such constraints are imposed through a finite control property. That is, some finite reachability criterion is excerpted on the semantics of Homer processes. Such a criterion is the key reason for the decidability of barbed bisimilarities. The approach of [6] provides a valuable reference for acquiring decidability sub-models from a more powerful full model.

However, there is still much space one can exploit concerning bisimilarity checking for higher-order processes, as mentioned in [15]. HOcore is a minimal model, with somewhat low modelling capacity. It would be interesting to quest for a more expressive model by adding certain constructs, while still maintaining the decidability result. Parameterization has been known to be an effective approach of promoting the expressiveness of higher-order processes, that is, abstraction-passing is strictly more expressive than mere process-passing [14]. In this work, we focus on the minimal higher-order processes with parameterization, notation Π^{mp} , basically HOcore extended with parameterization. This minimal model contains solely the most elementary parts to formalize higher-order concurrency, with extension of the abstraction and application, two operations originating from the lambda-calculus [2]. We will show that in such a calculus, the strong bisimilarity remains decidable. Similar result is only conjectured

in [15]. To this point, we go beyond that conjecture in two respects. Firstly, although our general approach resembles that of [15], the technical route has some key lemmas with essentially different proof structures, due to the complication brought by the parameterization. Secondly, we consider two kinds of parameterization, i.e., both on names and on processes themselves, rather than only one kind. Thus we are working on a potentially more expressive model. This is evidenced by the following two facts. (1) Parameterization, in particular process parameterization, brings strictly more expressiveness to the higher-order process model [14]. (2) Moreover, name parameterization is more expressive than process parameterization [27]. Intuitively, this is true because we can somehow encode process parameterization with name parameterization, using an idea akin to that of encoding process-passing into name-passing. To the best of our knowledge, there has been little work about the decidability of bisimilarities in such a model. The decidability result of this work not only pushes outward the boundary of higher-order processes with decidable bisimilarity, but also digs more into the realm of bisimilarity checking more challenging behavioural equalities, such as weak or branching bisimilarity.

Contribution Now we summarize the main contribution of this paper.

- We show that in the minimal higher-order process model with parameterization, the strong bisimilarities, including the standard context bisimilarity together with other well-known bisimulation equalities, are all decidable. We borrow and revamp the ideas from [15], i.e., defining a bisimilarity decidable from the very beginning and then showing that the bisimilarities of interest coincide with it. The major novel parts are those tackling the parameterization. Due to the presence of the parameterization, we have a completely new design of the key bisimilarities, particularly those defined directly over open processes (i.e., those processes carrying free variables), as well as the normal bisimulation that needs new forms of triggers for the two kinds of parameterizations. In turn, the congruence proofs must take these changes into consideration. Moreover, some crucial properties for establishing the coincidence of the bisimilarities have entirely new proof methods, in particular, among others, the preservation of substitution that claims the closure of variable substitutions with respect to the strong bisimilarity (since now a variable can take an abstraction). Indeed, the discussion of the mutual inclusion of various bisimilarities calls for more rigorous and fine-grained investigation in the setting of parameterization. More explanation is given in Sections 2, 3.

- With the decidability in place, we design an axiom system and a checking algorithm, in roughly the same vein as those in [15], with the following difference. (1) For the axiom system, the core part amounts to reducing the deduction of the strong bisimilarity to the extended structural congruence. Previously, such extension includes a distribution law. Now with parameterization in the game, we have to further extend the structural congruence with the laws for the application operation. (2) For the bisimilarity-checking algorithm, the core is to transform a term (possibly with parameterization) into certain normal form with the help of a tree representation of the process, and then the bisimilarity checking can be readily done almost syntactically on the normal form. In presence of parameterization, we extend the tree to accommodate abstractions and applications, as well as the normalization procedure. In such an extended procedure, we execute applications as many times any possible, and operate the tree in a bottom-up fashion so as to improve on performance. The algorithm has linear space complexity and polynomial time complexity slightly better than available ones. More details are given in Sections 4, 5.

An extended version of this paper with more details is available [26].

Organization The remainder of this paper is organized as follows. Section 2 gives the definitions of the process model and the strong bisimilarities. Section 3 presents the decidability of the strong bisimilarities, with detailed proofs. Section 4 does the axiomatization and proves its correctness. In Section 5, we demonstrate an algorithm for the bisimilarity checking, and analyses its complexity. Section 6 concludes this paper and points to some future work.

2 Preliminary

In this section, we first define Π^{mp} , the minimal higher-order process model extended with parameterization. Then we introduce the strong bisimilarities to be discussed.

Syntax Calculus Π^{mp} has the following syntax.

$$P, Q := 0 \mid X \mid m(X).P \mid \bar{m}(Q) \mid P \mid Q \mid \langle X \rangle P \mid P \langle Q \rangle \mid \langle x \rangle P \mid P \langle n \rangle$$

Π^{mp} expressions (or terms, processes) are represented by capital letters. For the sake of convenience, we divide names (ranged over by m, n, u, v, \dots) into two groups: one for name constants (ranged over by a, b, c, d, e, \dots) and the other for name variables (ranged over by x, y, z, \dots). The elements of the calculus have their standard meaning. One notices that the output is non-blocking, i.e., asynchronous. Sometimes we write $\bar{m}[Q]$ for output. Input $m(X).P$ and process abstraction $\langle X \rangle P$ bind the process variable X , and name abstraction $\langle x \rangle P$ binds the name variable x . Otherwise, a process or name variable is free. Bound variables can be replaced subject to α -conversion, and the resulting term is deemed as the same. A term is closed if it does not have free process variables. Otherwise it is open. Operations $\text{fpv}(\cdot)$, $\text{bpv}(\cdot)$, $\text{pv}(\cdot)$, $\text{fnv}(\cdot)$, $\text{bnv}(\cdot)$, $\text{nv}(\cdot)$, $\text{nc}(\cdot)$, $\text{n}(\cdot)$ respectively return the free process variables, bound process variables, process variables, free name variables, bound name variables, name variables, name constants, and names of a set of terms. A variable or name is fresh if it does not appear in the terms under examination. We use $\tilde{\cdot}$ for a tuple, for example, a tuple of terms \tilde{P} and a tuple of names \tilde{m} . Process substitution $P\{Q/X\}$ (respectively name substitution $P\{m/x\}$) denotes the replacement of process variable X (respectively name variable x) with the process Q (respectively name m). Substitutions can be extended to tuples in the expected way, i.e., pairwise replacement.

Parameterization refers to abstraction and application, and sometimes parameterization and abstraction are used interchangeably. Intuitively the process abstraction $\langle X \rangle P$ (respectively name abstraction $\langle x \rangle P$) *abstracts* in P the process variable X (respectively name variable x), which is supposed to be instantiated by a concrete process Q (respectively name d) in the application $(\langle X \rangle P)\langle Q \rangle$ (respectively $(\langle x \rangle P)\langle d \rangle$); then in turn the application gives rise to an applied form $P\{P/X\}$ (respectively $P\{d/x\}$). The constructs of abstract and application stem from the counterpart in the lambda-calculus, and somehow extend the domain of the lambda-calculus to a concurrent setting. To ensure correct use of abstraction and application, a type system was designed by Sangiorgi in his seminal thesis [19]. The typing rules in the type system effect to exclude badly formed expressions, such as $(\langle x \rangle P)\langle A \rangle$ and $\langle A \rangle P$ in which A is a (non-variable) term, $P \mid \langle X \rangle Q$ (dangling abstraction), and so on. That type system is important but not essential for our work here, so we do not present it and always assume that terms are well-formed subject to typing; interested readers can refer to [19, 21] and reference thereof for more details. Term $P_1 \mid P_2 \mid \dots \mid P_k$ is abbreviated as $\prod_{i=1}^k P_i$. We also have some CCS-like operations defined as follows: $a.P \stackrel{\text{def}}{=} a(X).P$ where $X \notin \text{pv}(P)$; $a \stackrel{\text{def}}{=} a.0$; $\bar{a} \stackrel{\text{def}}{=} \bar{a}.0$. A context $C[\cdot]$ is an expression with some sub-expression replaced by the hole $[\cdot]$, and $C[A]$ means substituting the hole with A .

Semantics We denote by \equiv the standard structural congruence extended by the rules for application, i.e., the smallest congruence meeting the following laws among which the last two formulate the application. $(P \mid Q) \mid R \equiv P \mid (Q \mid R)$, $P \mid Q \equiv Q \mid P$, $P \mid 0 \equiv P$, $(\langle X \rangle P)\langle Q \rangle \equiv P\{Q/X\}$, $(\langle x \rangle P)\langle m \rangle \equiv P\{m/x\}$

Calculus Π^{mp} has the following operational semantics on open terms, with symmetric rules skipped. In the third rule, we assume $\text{bpv}(\lambda) \cap \text{fpv}(Q) = \emptyset$.

$$\frac{}{m(X).P \xrightarrow{m(X)} P} \quad \frac{}{\bar{m}(Q) \xrightarrow{\bar{m}(Q)} 0} \quad \frac{P \xrightarrow{\lambda} P'}{P \mid Q \xrightarrow{\lambda} P' \mid Q} \quad \frac{P \xrightarrow{\bar{m}(A)} P' \quad Q \xrightarrow{m(X)} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'\{A/X\}} \quad \frac{Q \equiv P \quad P \xrightarrow{\lambda} P' \quad P' \equiv Q'}{Q \xrightarrow{\lambda} Q'}$$

The semantics grant a term three kinds of actions: input $P \xrightarrow{a(X)} P'$ means that P can receive a term on channel a to replace the variable X acting as a place-holder in P (here we have a late instantiation style); output $P \xrightarrow{\bar{a}(Q)} P'$ means that P can send a term Q (which could be an abstraction) on channel a in an asynchronous fashion; interaction $P \xrightarrow{\tau} P'$ means that P makes a communication of some term between concurrent components. Actions are ranged over by α, λ . Operations $\text{fpv}(\cdot)$, $\text{bpv}(\cdot)$, $\text{pv}(\cdot)$, $\text{fnv}(\cdot)$, $\text{bnv}(\cdot)$, $\text{nv}(\cdot)$, $\text{nc}(\cdot)$, $\text{n}(\cdot)$ and also substitutions can be extended to actions in the expected way accordingly. We sometimes write $P \xrightarrow{\lambda} \cdot$ to represent the transition $P \xrightarrow{\lambda} P'$ for some P' if P' is not important. Modelling application as part of the structural congruence follows the line of reduction in lambda-calculus, though there are other options (see [19, 21]). Thus up-to \equiv , a term can be somehow turned into an equivalent one by applying applications as many times as possible, ending up with a term containing only those application of the form $X\langle A \rangle$. As in [19], we ensure that applications (substitutions) are bound to end (i.e., normalized), so as to avoid Ω -like terms such as $O\langle O \rangle$ in which $O \stackrel{\text{def}}{=} \langle X \rangle(X\langle X \rangle)$. Said another way, in the sense of order, we focus on abstractions with finite order, not ω order. See [19, 21] for more discussion about this. We further notice that if infinite application were to be admitted (though this is a bit strange), then essentially one would retrieve replication, e.g., $!P \stackrel{\text{def}}{=} O'\langle O' \rangle$ in which $O' \stackrel{\text{def}}{=} \langle X \rangle(P|X\langle X \rangle)$. This would probably lead to a drastically different situation, which we do not tackle in this work. **Before moving on, we give an example to illustrate the modelling capability of Π^{mp} . We define two processes P and Q executing a simple protocol, making good use of the parameterization.**

$$\begin{aligned}
 P &\stackrel{\text{def}}{=} \bar{a}A|b(X).(X\langle B \rangle|O), & Q &\stackrel{\text{def}}{=} a(X).(X\langle c \rangle|c(Y).R), & A &\stackrel{\text{def}}{=} \langle x \rangle(\bar{b}[\langle Z \rangle \bar{x}Z]) \\
 P|Q &\xrightarrow{\tau} \equiv b(X).(X\langle B \rangle|O)|A\langle c \rangle|c(Y).R && \equiv b(X).(X\langle B \rangle|O)|\bar{b}[\langle Z \rangle \bar{c}Z]|c(Y).R \\
 &\xrightarrow{\tau} \equiv (\langle Z \rangle \bar{c}Z)\langle B \rangle|O|c(Y).R && \equiv \bar{c}B|O|c(Y).R \\
 &\xrightarrow{\tau} \equiv R\{B/Y\}|O
 \end{aligned}$$

The protocol goes as follows: (1) P sends Q an abstraction A over channel a (which is agreed upon beforehand); (2) Q instantiates the name abstraction carried by A with a name c chosen by Q alone (not necessarily negotiated with P before starting the protocol); (3) Part of the code of A , i.e., $\langle Z \rangle \bar{c}Z$ is sent back to P over channel b chosen by P alone previously; (4) Process B , e.g., some computational resource or data, is sent to Q over channel c , so as to be used in R . In the entire protocol, P and Q only agree on the channel name a , and initially do not disclose on which channel the resource is to be transmitted.

Below we give the notion of “guarded” and some relevant properties.

Definition 1. A variable X is guarded in P if X merely occurs in the following two situations. (1) X occurs in P 's subexpressions of the form $m(Y).P'$ (in which Y could be the same as X), or $Y\langle P' \rangle$ (in which Y is not X). (2) X occurs free in P 's subexpressions of the form $\bar{m}P'$. A term P is guarded if any free variable of it is guarded.

In what follows, we have the abbreviations: $\text{Tr}_m \stackrel{\text{def}}{=} \bar{m}$, $\text{Tr}_m^D \stackrel{\text{def}}{=} \langle Z \rangle \bar{m}Z$, $\text{Tr}_m^{D,d} \stackrel{\text{def}}{=} \langle z \rangle \bar{m}[\langle Z \rangle (Z\langle z \rangle)]$. The proofs of the coming two lemmas are by transition induction.

Lemma 2. We have the following transition properties.

- (1) If $P \xrightarrow{\lambda} P'$, then $P\{R/X\} \xrightarrow{\lambda\{R/X\}} P'\{R/X\}$ for every R with $\text{fpv}(R) \cap (\text{pv}(P) \cup \text{pv}(\lambda) \cup \{X\}) = \emptyset$.
- (2) If $P\{R/X\} \xrightarrow{\lambda'} P_1$ with X guarded in P and $\text{fpv}(R) \cap (\text{pv}(P) \cup \{X\}) = \emptyset$, then $P \xrightarrow{\lambda} P'$, $P_1 \equiv P'\{R/X\}$, and λ' is $\lambda\{R/X\}$ with $\text{fpv}(R) \cap \text{pv}(\lambda) = \emptyset$.
- (3) If $P\{\text{Tr}_m/X\} \xrightarrow{\lambda'} P_1$ with m fresh and not in λ' , then $P \xrightarrow{\lambda} P'$, $P_1 \equiv P'\{\text{Tr}_m/X\}$, and λ' is $\lambda\{\text{Tr}_m/X\}$.

- (4) If $P\{\text{Tr}_m^D/X\} \xrightarrow{\lambda'} P_1$ with m fresh and not in λ' , then $P \xrightarrow{\lambda} P'$, $P_1 \equiv P'\{\text{Tr}_m^D/X\}$, and λ' is $\lambda\{\text{Tr}_m^D/X\}$.
- (5) If $P\{\text{Tr}_m^{D,d}/X\} \xrightarrow{\lambda'} P_1$ with m fresh and not in λ' , then $P \xrightarrow{\lambda} P'$, $P_1 \equiv P'\{\text{Tr}_m^{D,d}/X\}$, and λ' is $\lambda\{\text{Tr}_m^{D,d}/X\}$.
- (6) If $P \xrightarrow{\lambda} P'$, then $P\{g/m\} \xrightarrow{\lambda\{g/m\}} P'\{g/m\}$.
- (7) If $P\{g/m\} \xrightarrow{\lambda'} P_1$ and λ' is not τ , then $P \xrightarrow{\lambda} P'$ in which λ' is $\lambda\{g/m\}$, and $P_1 \equiv P'\{g/m\}$.
- (8) If $P\{g/m\} \xrightarrow{\tau} P_1$, then there are several possibilities: (a) $P \xrightarrow{\tau} P'$ and $P_1 \equiv P'\{g/m\}$. (b) $P \xrightarrow{\bar{m}A} \cdot$ and $P \xrightarrow{g(Y)} \cdot$. That is, $P \equiv \bar{m}A | g(Y).P_2 | P_3$, and $P_1 \equiv (P_2\{A/Y\} | P_3)\{g/m\}$. (c) $P \xrightarrow{\bar{g}A} \cdot$ and $P \xrightarrow{m(Y)} \cdot$. That is, $P \equiv \bar{g}A | m(Y).P_2 | P_3$, and $P_1 \equiv (P_2\{A/Y\} | P_3)\{g/m\}$.

Lemma 3. Assume that P is a term and X is a process variable. There are P' in which X is guarded and natural number $k \geq 0$ such that one of the following cases is true. (1) $P \equiv P' | \Pi_{i=1}^k X$, and $P\{R/X\} \equiv P'\{R/X\} | \Pi_{i=1}^k R$ for every R . (2) $P \equiv P' | \Pi_{i=1}^k X\langle A_i \rangle$, and $P\{R/X\} \equiv P'\{R/X\} | \Pi_{i=1}^k R\langle A_i\{R/X\} \rangle$ for every R . (3) $P \equiv P' | \Pi_{i=1}^k X\langle m_i \rangle$, and $P\{R/X\} \equiv P'\{R/X\} | \Pi_{i=1}^k R\langle m_i \rangle$ for every R .

The strong bisimilarities In the following, we first present a provably decidable strong bisimilarity, named strong HO-IO bisimilarity. Then we go head to define the various strong bisimilarities, including the strong context bisimilarity and other strong bisimilarities of concern. These strong bisimilarities turn out to be equal.

Strong HO-IO bisimilarity We define a bisimulation called strong HO-IO bisimulation, with the corresponding equality called strong HO-IO bisimilarity. As will be seen, the most desirable properties we want from this bisimilarity is that it is decidable. The definition needs to take into account the abstractions, because the terms transmitted to be compared may be abstractions. Jumping ahead, the other strong bisimulations to be defined also have this requirement for abstractions.

Definition 4 (Strong HO-IO bisimilarity). A symmetric binary relation \mathcal{R} over Π^{mp} terms is a strong HO-IO bisimulation, if whenever $P \mathcal{R} Q$ the following properties hold.

- (1) If P is a non-abstraction, then so is Q . (2) If P is a process-abstraction $\langle Y \rangle A$, then Q is a process-abstraction $\langle Y \rangle B$, and $A \mathcal{R} B$. (3) If P is a name-abstraction $\langle y \rangle A$, then Q is a name-abstraction $\langle y \rangle B$, and $A \mathcal{R} B$. (4) If $P \xrightarrow{\bar{a}A} P'$, then $Q \xrightarrow{\bar{a}B} Q'$ with $A \mathcal{R} B$ and $P' \mathcal{R} Q'$. (5) If $P \xrightarrow{a(X)} P'$, then $Q \xrightarrow{a(X)} Q'$ and $P' \mathcal{R} Q'$. (6) If $P \equiv X | P'$, then $Q \equiv X | Q'$ and $P' \mathcal{R} Q'$. (7) If $P \equiv X\langle A \rangle | P'$, then $Q \equiv X\langle B \rangle | Q'$, and $A \mathcal{R} B$ and $P' \mathcal{R} Q'$. (8) If $P \equiv X\langle d \rangle | P'$, then $Q \equiv X\langle d \rangle | Q'$ and $P' \mathcal{R} Q'$. The strong HO-IO bisimilarity, notation \sim_{hoio}° , is the largest strong HO-IO bisimulation.

Strong HO bisimilarity The concept of strong HO bisimilarity is due to Thomsen [24].

Definition 5 (Strong HO bisimilarity). A symmetric binary relation \mathcal{R} over closed Π^{mp} terms is a strong HO bisimulation, if whenever $P \mathcal{R} Q$ the following properties hold.

- (1) If P is a non-abstraction, then so is Q . (2) If P is a process-abstraction $\langle Y \rangle P'$, then Q is a process-abstraction $\langle Y \rangle Q'$, and $P'\{A/Y\} \mathcal{R} Q'\{A/Y\}$ for every closed A . (3) If P is a name-abstraction $\langle y \rangle A$, then Q is a name-abstraction $\langle y \rangle B$, and $A \mathcal{R} B$. (4) If $P \xrightarrow{\bar{a}A} P'$, then $Q \xrightarrow{\bar{a}B} Q'$ with $A \mathcal{R} B$ and $P' \mathcal{R} Q'$. (5) If $P \xrightarrow{a(X)} P'$, then $Q \xrightarrow{a(X)} Q'$ and for every closed A , it holds that $P'\{A/X\} \mathcal{R} Q'\{A/X\}$. (6) If $P \xrightarrow{\tau} P'$, then $Q \xrightarrow{\tau} Q'$ and $P' \mathcal{R} Q'$. The strong HO bisimilarity, notation \sim_{ho} , is the largest strong HO bisimulation.

Strong context bisimilarity We denote by $E(X)$ a process E possibly with X appearing free in it, i.e., $\text{fpv}(E) \subseteq \{X\}$. Accordingly, $E(A)$ denotes $E(X)\{A/X\}$. As a nearly standard version of the bisimilarity for higher-order processes, the context bisimulation was proposed by Sangiorgi [19].

Definition 6 (Strong context bisimilarity). A symmetric binary relation \mathcal{R} over closed Π^{mp} terms is a strong context bisimulation, if whenever $P \mathcal{R} Q$ the following properties hold. **(1)** If P is a non-abstraction, then so is Q . **(2)** If P is a process-abstraction $\langle Y \rangle P'$, then Q is a process-abstraction $\langle Y \rangle Q'$, and $P' \{A/Y\} \mathcal{R} Q' \{A/Y\}$ for every closed A . **(3)** If P is a name-abstraction $\langle y \rangle A$, then Q is a name-abstraction $\langle y \rangle B$, and $A \mathcal{R} B$. **(4)** If $P \xrightarrow{a(X)} P'$, then $Q \xrightarrow{a(X)} Q'$ and for every closed A , it holds that $P' \{A/X\} \mathcal{R} Q' \{A/X\}$. **(5)** If $P \xrightarrow{\bar{a}A} P'$ in which A is a non-abstraction, process-abstraction, or name-abstraction, then $Q \xrightarrow{\bar{a}B} Q'$ for some B that is respectively a non-abstraction, process-abstraction, or name-abstraction, and for every $E(X)$, it holds that $E(A) | P' \mathcal{R} E(B) | Q'$. **(6)** If $P \xrightarrow{\tau} P'$, then $Q \xrightarrow{\tau} Q'$ and $P' \mathcal{R} Q'$. The strong context bisimilarity, notation \sim_{ctx} , is the largest strong context bisimulation.

We note that \sim_{ctx} can be extended to open process terms, similar for \sim_{ho} . That is, for open terms P and Q with $\text{fpv}(P, Q) = \tilde{X}$, $P \sim_{\text{ctx}} Q$ if and only if $P\{\tilde{R}/\tilde{X}\} \sim_{\text{ctx}} Q\{\tilde{R}/\tilde{X}\}$ for any closed \tilde{R} .

Strong normal bisimilarity Higher-order process expressions here may be parameterized over processes themselves or names. Accordingly, abstractions can be transmitted in communications, and thus process variables have three types: non-abstraction, process-abstraction, and name-abstraction. We refer the reader to [19] for the detailed formalization of types. To cater for our need, knowing which of the three types a process variable belongs to is sufficient for our work. For convenience, we may simply say that a process variable is a non-abstraction, process-abstraction, or name-abstraction.

Before presenting the definition of the strong normal bisimilarity, we give the definition of triggers: $\text{Tr}_m \stackrel{\text{def}}{=} \bar{m}$, $\text{Tr}_m^D \stackrel{\text{def}}{=} \langle Z \rangle \bar{m}Z$, $\text{Tr}_m^{D,d} \stackrel{\text{def}}{=} \langle z \rangle \bar{m}[\langle Z \rangle (Z \langle z \rangle)]$. These triggers correspond to the three types of process variables represented above, and will be used to handle abstractions bound to instantiate these process variables. The concept of triggers was proposed by Sangiorgi and plays a prevalent role in the manipulation of higher-order processes; see [19] [25]. We stress that the design of the normal bisimulation in this work requires new forms of triggers due to the presence of parameterization. The work in [15] only needs the simplest form of triggers acting as synchronizers sending handshaking signals, i.e., \bar{m} . However in contrast, in the setting of parameterization, triggers should bear the responsibility of relocating the parameters for an abstraction. This design is non-trivial in general, and we harness the results in the previous work [25] to devise different forms of triggers used by the parameterization. It is not hard to prove that the strong normal bisimilarity is a congruence [19].

Definition 7 (Strong normal bisimilarity). A symmetric binary relation \mathcal{R} over closed Π^{mp} terms is a strong normal bisimulation, if whenever $P \mathcal{R} Q$ the following properties hold.

- (1)** If P is a non-abstraction, then so is Q . **(2)** If P is a process-abstraction $\langle Y \rangle P'$, then Q is a process-abstraction $\langle Y \rangle Q'$, and for every closed A it holds for fresh m that:
- (a)** $P' \{\text{Tr}_m/Y\} \mathcal{R} Q' \{\text{Tr}_m/Y\}$, if Y is a non-abstraction. **(b)** $P' \{\text{Tr}_m^D/Y\} \mathcal{R} Q' \{\text{Tr}_m^D/Y\}$, if Y is a process-abstraction. **(c)** $P' \{\text{Tr}_m^{D,d}/Y\} \mathcal{R} Q' \{\text{Tr}_m^{D,d}/Y\}$, if Y is a name-abstraction.
- (3)** If P is a name-abstraction $\langle y \rangle A$, then Q is a name-abstraction $\langle y \rangle B$, and $A \mathcal{R} B$.
- (4)** If $P \xrightarrow{a(X)} P'$, then $Q \xrightarrow{a(X)} Q'$ and for every closed A , it holds for fresh m that:
- (a)** $P' \{\text{Tr}_m/X\} \mathcal{R} Q' \{\text{Tr}_m/X\}$, if X is a non-abstraction. **(b)** $P' \{\text{Tr}_m^D/X\} \mathcal{R} Q' \{\text{Tr}_m^D/X\}$, if X is a process-abstraction. **(c)** $P' \{\text{Tr}_m^{D,d}/X\} \mathcal{R} Q' \{\text{Tr}_m^{D,d}/X\}$, if X is a name-abstraction.
- (5)** If $P \xrightarrow{\bar{a}A} P'$, there are three possibilities: **(a)** If A is not an abstraction, then $Q \xrightarrow{\bar{a}B} Q'$ for non-abstraction B , and it holds for fresh m that $m.A | P' \mathcal{R} m.B | Q'$. **(b)** If A is a process-abstraction $\langle Y \rangle A_1$, then $Q \xrightarrow{\bar{a}B} Q'$ for process-abstraction B that is $\langle Y \rangle B_1$, and it holds for fresh m that $m(Z).A \langle Z \rangle | P' \mathcal{R} m(Z).B \langle Z \rangle | Q'$. **(c)** If A is a name-abstraction $\langle y \rangle A_1$, then $Q \xrightarrow{\bar{a}B} Q'$ for name-abstraction B that is $\langle y \rangle B_1$, and it holds for fresh m that $m(Z).Z \langle A \rangle | P' \mathcal{R} m(Z).Z \langle B \rangle | Q'$.

(6) If $P \xrightarrow{\tau} P'$, then $Q \xrightarrow{\tau} Q'$ and $P' \mathcal{R} Q'$. The strong normal bisimilarity, notation \sim_{nr} , is the largest strong normal bisimulation.

We can also extend \sim_{nr} to open terms. For open terms P and Q with $\text{fpv}(P, Q) = \{\tilde{X}_1, \tilde{X}_2, \tilde{X}_3\}$, $P \sim_{\text{nr}} Q$ if and only if $P\{\widetilde{\text{Tr}}_{m_1}/\tilde{X}_1\}\{\widetilde{\text{Tr}}_{m_2}^D/\tilde{X}_2\}\{\widetilde{\text{Tr}}_{m_3}^{D,d}/\tilde{X}_3\} \sim_{\text{nr}} Q\{\widetilde{\text{Tr}}_{m_1}/\tilde{X}_1\}\{\widetilde{\text{Tr}}_{m_2}^D/\tilde{X}_2\}\{\widetilde{\text{Tr}}_{m_3}^{D,d}/\tilde{X}_3\}$, where each variable in \tilde{X}_1, \tilde{X}_2 and \tilde{X}_3 is respectively a non-abstraction, process-abstraction and name-abstraction, and is replaced with the corresponding trigger for that variable type. The corresponding tuples of triggers are respectively denoted by $\widetilde{\text{Tr}}_{m_1}, \widetilde{\text{Tr}}_{m_2}^D$ and $\widetilde{\text{Tr}}_{m_3}^{D,d}$, where the names of all the triggers are fresh.

Open strong normal bisimilarity The following bisimilarity is a variant of the strong normal bisimilarity on open terms. It is basically an extension of the same bisimilarity in [15].

Definition 8 (Open strong normal bisimilarity). A symmetric binary relation \mathcal{R} over Π^{mp} terms is an open strong normal bisimulation, if whenever $P \mathcal{R} Q$ the following properties hold.

- (1) If P is a non-abstraction, then so is Q . (2) If P is a process-abstraction $\langle Y \rangle A$, then Q is a process-abstraction $\langle Y \rangle B$, and $A \mathcal{R} B$. (3) If P is a name-abstraction $\langle y \rangle A$, then Q is a name-abstraction $\langle y \rangle B$, and $A \mathcal{R} B$. (4) If $P \xrightarrow{\bar{a}A} P'$ or $P \xrightarrow{\tau} P'$, then Q matches P in the same way as in strong normal bisimilarity. (5) If $P \xrightarrow{a(X)} P'$, then $Q \xrightarrow{a(X)} Q'$ and $P' \mathcal{R} Q'$. (6) If $P \equiv X | P'$, then $Q \equiv X | Q'$ and $P' \mathcal{R} Q'$. (7) If $P \equiv X \langle A \rangle | P'$, then $Q \equiv X \langle B \rangle | Q'$ and moreover the following is valid for fresh m . (a) If A is not an abstraction, then so is B and $m.A | P' \mathcal{R} m.B | Q'$. (b) If A is a process-abstraction, then so is B and $m(Z).A \langle Z \rangle | P' \mathcal{R} m(Z).B \langle Z \rangle | Q'$. (c) If A is a name-abstraction, then so is B and $m(Z).Z \langle A \rangle | P' \mathcal{R} m(Z).Z \langle B \rangle | Q'$. (8) If $P \equiv X \langle d \rangle | P'$, then $Q \equiv X \langle d \rangle | Q'$ and $P' \mathcal{R} Q'$.

The open strong normal bisimilarity, notation \sim_{nr}° , is the largest open strong normal bisimulation.

3 Deciding the strong bisimilarity for Π^{mp}

In this section, we first establish the decidability of the strong HO-IO bisimilarity. This is the cornerstone of the decidability for other bisimilarities. Then we discuss the relationship between the strong bisimilarities, and eventually obtain the coincidence between them. As such, all of the strong bisimilarities are decidable.

3.1 The decidability and properties of \sim_{hoio}°

To facilitate discussion on decidability, we need a metric of the syntactical structure of a term.

Definition 9 (Depth of a term). The depth $\text{depth}(P)$ of a term P is a mapping from terms to natural numbers defined as follows.

$$\begin{aligned} \text{depth}(0) &= 0, \text{depth}(X) = 1, \text{depth}(m(X).P_1) = \text{depth}(P_1) + 1, \text{depth}(\bar{m}(P_1)) = \text{depth}(P_1) + 1, \\ \text{depth}(P_1 | P_2) &= \text{depth}(P_1) + \text{depth}(P_2), \text{depth}(\langle X \rangle P_1) = \text{depth}(P_1) + 1, \text{depth}(X \langle P_1 \rangle) = \text{depth}(P_1) + 1, \\ \text{depth}(P_1 \langle P_2 \rangle) &= \text{depth}(P_3 \{P_2/Y\}) \text{ (where } P_1 \text{ is } \langle Y \rangle P_3), \text{depth}(\langle x \rangle P_1) = \text{depth}(P_1) + 1, \\ \text{depth}(X \langle n \rangle) &= 1, \text{depth}(P_1 \langle n \rangle) = \text{depth}(P_3 \{n/y\}) \text{ (where } P_1 \text{ is } \langle y \rangle P_3) \end{aligned}$$

An immediate property is that both of $P \equiv Q$ and $P \sim_{\text{hoio}}^\circ Q$ implies $\text{depth}(P) = \text{depth}(Q)$. The proof of this property is by induction over the depth of P . The details are put in Appendix A.

Lemma 10. *If $P \equiv Q$ or $P \sim_{\text{hoio}}^\circ Q$, then $\text{depth}(P) = \text{depth}(Q)$.*

Strong HO-IO bisimulation up-to \equiv Bisimulation up-to \equiv is a useful technique to establish bisimulations. Its definition is obtained by replacing \mathcal{R} with $\equiv \mathcal{R} \equiv$ in every clause of Definition 4. The

advantage of the up-to technique is that if \mathcal{R} is a strong HO-IO bisimulation up-to \equiv , then $\mathcal{R} \subseteq \sim_{\text{hoio}}^\circ$. See [16, 21] for a thorough introduction and discussion.

Congruence Through standard state-diagram-chasing argument, one can prove that \sim_{hoio}° is an equivalence relation. It is also a congruence, as the follow-up lemma reveals. See [15, 19] for a reference of proof; we also provide a proof in Appendix A.

Lemma 11 (Congruence). *On Π^{mp} terms, \sim_{hoio}° is congruence. That is, suppose P and Q are Π^{mp} terms, then $P \sim_{\text{hoio}}^\circ Q$ implies: (1) $a(X).P \sim_{\text{hoio}}^\circ a(X).Q$; (2) $\bar{a}(P) \sim_{\text{hoio}}^\circ \bar{a}(Q)$; (3) $P|R \sim_{\text{hoio}}^\circ Q|R$; (4) $\langle X \rangle P \sim_{\text{hoio}}^\circ \langle X \rangle Q$; (5) $\langle x \rangle P \sim_{\text{hoio}}^\circ \langle x \rangle Q$; (6) $Y\langle P \rangle \sim_{\text{hoio}}^\circ Y\langle Q \rangle$.*

Decidability We now establish the decidability of \sim_{hoio}° . As a premise, we have the following structural property, whose proof is a simple induction over the semantic rules.

Lemma 12. *Suppose P is a Π^{mp} term. Then: (1) If $P \xrightarrow{\bar{a}(A)} P'$, then $P \equiv \bar{a}(A) | P'$. (2) If $P \xrightarrow{a(X)} P'$, then $P \equiv a(X).P_1 | P_2$ and $P' \equiv P_1 | P_2$.*

Lemma 13 (Decidability). *On Π^{mp} terms, \sim_{hoio}° is decidable.*

Proof of Lemma 13. We decide whether $P \sim_{\text{hoio}}^\circ Q$ by induction on $\text{depth}(P)$.

Induction basis. In this case $\text{depth}(P)$ is 0 or 1. The case $\text{depth}(P)$ is 0, i.e., P is 0, is trivial because no action is possible from P and it has no free variables. If $\text{depth}(P)$ is 1, i.e., P is X or $X\langle d \rangle$, then no action is possible from P . One simply checks that Q is also X or $X\langle d \rangle$ respectively.

Induction step. We perform a (finite) check of each clause of \sim_{hoio}° in an inductive way.

(1) If P is a non-abstraction, then check that Q is also a non-abstraction.

(2) If P is a process-abstraction $\langle X \rangle A$, then check that Q is also a process-abstraction $\langle X \rangle B$ (up-to α -conversion), and continue with checking $A \sim_{\text{hoio}}^\circ B$ using induction hypothesis since the depth of A decreases with respect to P , i.e., $\text{depth}(A) < \text{depth}(P)$.

(3) If P is a name-abstraction $\langle x \rangle A$, then check that Q is also a name-abstraction $\langle x \rangle B$ (up-to α -conversion), and continue with checking $A \sim_{\text{hoio}}^\circ B$ using induction hypothesis since the depth of A decreases with respect to P , i.e., $\text{depth}(A) < \text{depth}(P)$.

(4) If $P \xrightarrow{a(X)} P'$, we check that $Q \xrightarrow{a(X)} Q'$ and $P' \sim_{\text{hoio}}^\circ Q'$. There might be a few (but finite) possibilities concerning Q' . If all such checks fail, then we conclude that P and Q are not strong HO-IO bisimilar. For each possible check, we know by Lemma 12 that $P \equiv a(X).P_1 | P_2$ and $P' \equiv P_1 | P_2$. Since the depth of the terms decrease, i.e., $\text{depth}(P') < \text{depth}(P)$, we use induction hypothesis to continue checking $P' \sim_{\text{hoio}}^\circ Q'$.

(5) If $P \xrightarrow{\bar{a}(A)} P'$, we check that $Q \xrightarrow{\bar{a}(B)} Q'$ with $A \sim_{\text{hoio}}^\circ B$ and $P' \sim_{\text{hoio}}^\circ Q'$. There might be a few (but finite) possibilities concerning B and Q' . If all such checks fail, then we conclude that P and Q are not strong HO-IO bisimilar. For each possible check, we know by Lemma 12 that $P \equiv \bar{a}(A) | P'$. Since the depth of the terms decrease, i.e., $\text{depth}(P') < \text{depth}(P)$ and $\text{depth}(A) < \text{depth}(P)$, we use induction hypothesis to continue checking $A \sim_{\text{hoio}}^\circ B$ and $P' \sim_{\text{hoio}}^\circ Q'$.

(6) If $P \equiv X | P'$, we check that $Q \equiv X | Q'$ and $P' \sim_{\text{hoio}}^\circ Q'$. There might be a few (but finite) possibilities concerning Q' . If all such checks fail, then we conclude that P and Q are not strong HO-IO bisimilar. For each possible check, since the depth of the terms decrease, i.e., $\text{depth}(P') < \text{depth}(P)$, we use induction hypothesis to continue checking $P' \sim_{\text{hoio}}^\circ Q'$.

(7) If $P \equiv X\langle A \rangle | P'$, we check that $Q \equiv X\langle B \rangle | Q'$ with $A \sim_{\text{hoio}}^\circ B$ and $P' \sim_{\text{hoio}}^\circ Q'$. There might be a few (but finite) possibilities concerning B and Q' . If all such checks fail, then we conclude that P and Q are not strong HO-IO bisimilar. For each possible check, since the depth of the terms decrease, i.e., $\text{depth}(P') < \text{depth}(P)$ and $\text{depth}(A) < \text{depth}(P)$, we use induction hypothesis to continue checking $A \sim_{\text{hoio}}^\circ B$ and $P' \sim_{\text{hoio}}^\circ Q'$.

(8) If $P \equiv X\langle d \rangle | P'$, we check that $Q \equiv X\langle d \rangle | Q'$ and $P' \sim_{\text{hoio}}^{\circ} Q'$. There might be a few (but finite) possibilities concerning Q' . If all such checks fail, then we conclude that P and Q are not strong HO-IO bisimilar. For each possible check, since the depth of the terms decrease, i.e., $\text{depth}(P') < \text{depth}(P)$, we use induction hypothesis to continue checking $P' \sim_{\text{hoio}}^{\circ} Q'$. \square

Bisimilarity Preservation To connect the strong HO-IO bisimilarity with the strong context bisimilarity and other bisimilarities, we need some preparation. In what follows, we show that the strong HO-IO bisimilarity preserves substitutions and τ simulation, as stated in the following two lemmas in a sequel.

Lemma 14 (Name-substitution-preserving). *Assume $P \sim_{\text{hoio}}^{\circ} Q$. Then $P\{g/m\} \sim_{\text{hoio}}^{\circ} Q\{g/m\}$ for all g, m .*

We note that to keep well-formed, substitutions should not (and are always assumed not to) break the legality of the terms under operation. The following lemma states that $\sim_{\text{hoio}}^{\circ}$ is invariant with respect to process substitution. As mentioned, the proof of this lemma (i.e., the process substitution preserving property) is entirely different from the counterpart in [15], in that we are obliged to conduct an induction on the sizes of the terms because a term in the position of application, say the term A in $X\langle A \rangle$, may introduce extra structures. **That is, the proof of the preservation of process substitution becomes much more involved due to the process parameterization. In the current setting, a process variable can be instantiated by a process abstraction which is in turn fed with a process from the context. This would give rise to certain circular arguments, so the original proof method of [15] no longer works. To work around this difficulty, one has to use induction based approach. This approach somewhat reminds one of the difficulty in proving the congruence properties for higher-order processes.**

Lemma 15 (Process-substitution-preserving). *Let $P \sim_{\text{hoio}}^{\circ} Q$. Then $P\{R/X\} \sim_{\text{hoio}}^{\circ} Q\{R/X\}$ for all R, X .*

The proofs of Lemmas 14 and 15 are put in Appendix A. An observation as a corollary from these two lemmas is that $\sim_{\text{hoio}}^{\circ}$ is closed under abstraction, both name abstraction and process-abstraction.

Corollary 16. *Assume $P \sim_{\text{hoio}}^{\circ} Q$. It holds that $\langle X \rangle P \sim_{\text{hoio}}^{\circ} \langle X \rangle Q$ and $\langle x \rangle P \sim_{\text{hoio}}^{\circ} \langle x \rangle Q$.*

The following lemma is important and directly attributed to the open-style nature of $\sim_{\text{hoio}}^{\circ}$.

Lemma 17 (τ -preserving). *Assume $P \sim_{\text{hoio}}^{\circ} Q$. If $P \xrightarrow{\tau} P'$, then $Q \xrightarrow{\tau} Q'$ and $P' \sim_{\text{hoio}}^{\circ} Q'$.*

Proof of Lemma 17. The proof is by induction on the derivation of $P \xrightarrow{\tau} P'$.

(1) $P \xrightarrow{\tau} P'$ comes from the interaction of components of P . That is, $P \xrightarrow{\bar{a}(A)} \cdot$, $P \xrightarrow{a(X)} \cdot$, and $P \xrightarrow{\tau} P'$. One can assume X to be fresh as it is bound. So this can be rewritten as $P \xrightarrow{\bar{a}(A)} \cdot \xrightarrow{a(X)} P_1$ where $P_1\{A/X\} \equiv P'$.

Because $P \sim_{\text{hoio}}^{\circ} Q$, Q can simulate by $Q \xrightarrow{\bar{a}(B)} \cdot \xrightarrow{a(X)} Q_1 \sim_{\text{hoio}}^{\circ} P_1$, where $A \sim_{\text{hoio}}^{\circ} B$, and $Q_1\{B/X\} \stackrel{\text{def}}{=} Q'$. Now since the higher-order output is non-blocking, the two consecutive actions can contribute to forming a τ action, i.e., $Q \xrightarrow{\tau} Q'$, and we are left with showing $P' \sim_{\text{hoio}}^{\circ} Q'$. From $P_1 \sim_{\text{hoio}}^{\circ} Q_1$, Lemma 15, we know $P' \equiv P_1\{A/X\} \sim_{\text{hoio}}^{\circ} Q_1\{A/X\} \stackrel{\text{def}}{=} Q''$. By the congruence properties, we can derive due to $A \sim_{\text{hoio}}^{\circ} B$ that $Q'' \equiv Q_1\{A/X\} \sim_{\text{hoio}}^{\circ} Q_1\{B/X\} \equiv Q'$. Hence we conclude $P' \sim_{\text{hoio}}^{\circ} Q'$.

(2) $P \xrightarrow{\tau} P'$ comes from a component of P alone. That is, $P \equiv P_1 | P_2$, $P_1 \xrightarrow{\tau} P'_1$, and $P' \equiv P'_1 | P_2$. Then we conclude by induction hypothesis. \square

3.2 Relating the strong bisimilarities

We represent detailed relationship between the strong bisimilarities defined so far. Such relationship will be established step by step. Eventually, as our ultimate goal, it will be demonstrated that all these strong bisimilarities coincide with each other. This coincidence immediately entails that every and each

of them is decidable, and moreover paves way for further discussion on the axiomatization and algorithm. We first establish the coincidence between \sim_{ho} and \sim_{hoio}° , and then move on to the remainder parts.

\sim_{ho} and \sim_{hoio}° coincide The following lemma gives a characteristic of the strong HO bisimilarity. Its proof is a standard bisimulation deduction, with details in Appendix A.

Lemma 18. *Suppose $\text{fpv}(P, Q) = \tilde{X} = X_1, \dots, X_n$. For fresh h and any closed \tilde{R} , it holds that $P\{\tilde{R}/\tilde{X}\} \sim_{ho} Q\{\tilde{R}/\tilde{X}\}$ if and only if $h(X_1) \cdot \dots \cdot h(X_n) \cdot P \sim_{ho} h(X_1) \cdot \dots \cdot h(X_n) \cdot Q$.*

As Lemma 19 states, the strong HO bisimilarity and the strong HO-IO bisimilarity are actually coincident. With the help of Lemma 18, and Lemmas 14, 15, 17 as well, we can prove the mutual inclusion of the two strong bisimilarities. The details are provided in Appendix A.

Lemma 19. *On Π^{mp} terms, $\sim_{ho} = \sim_{hoio}^\circ$.*

Relating \sim_{ho} and the other strong bisimilarities We now tackle the relationship between the strong HO bisimilarity and other strong bisimilarities, including the strong context bisimilarity (\sim_{ctx}), strong normal bisimilarity (\sim_{nr}), and open strong normal bisimilarity (\sim_{nr}°), as well as the remaining part of the overall picture of coincidence. To do this, we need some preparation. The following lemma will be useful. The proof employs the usual bisimulation-establishing method. The details are in Appendix A.

Lemma 20. *Assume that m is fresh with respect to P_1, Q_1, P and Q , said otherwise $m \notin \text{fn}(P_1, Q_1, P, Q)$. (1) If $m.P_1 \mid P \sim_{nr}^\circ m.Q_1 \mid Q$, then $P_1 \sim_{nr}^\circ Q_1$ and $P \sim_{nr}^\circ Q$. (2) If $m(Z).P_1\langle Z \rangle \mid P \sim_{nr}^\circ m(Z).Q_1\langle Z \rangle \mid Q$, then $P_1 \sim_{nr}^\circ Q_1$ and $P \sim_{nr}^\circ Q$. (3) If $m(Z).Z\langle P_1 \rangle \mid P \sim_{nr}^\circ m(Z).Z\langle Q_1 \rangle \mid Q$, then $P_1 \sim_{nr}^\circ Q_1$ and $P \sim_{nr}^\circ Q$.*

Using a similar proof strategy as Lemma 20, one can prove similarly the result for \sim_{nr} .

Corollary 21. *The result of Lemma 20 also holds if one replaces \sim_{nr}° with \sim_{nr} in the statement.*

Next in Lemma 22, we present the first two implications about the strong bisimilarities. Basically, the proof of Lemma 22(1) utilizes the congruence of \sim_{ho} and the proof of Lemma 22(2) uses the fact that the requirements of \sim_{nr} are actually special cases of \sim_{ctx} . The details can be found in Appendix A.

Lemma 22. (1) \sim_{ho} implies \sim_{ctx} on Π^{mp} processes. (2) \sim_{ctx} implies \sim_{nr} on Π^{mp} processes.

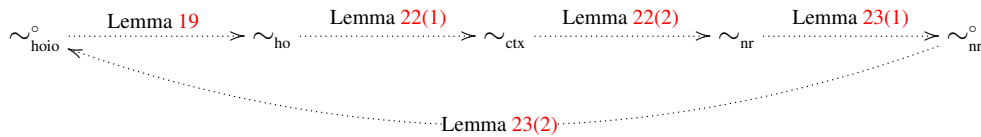
We now demonstrate, in Lemma 23, the two last inclusions about the strong bisimilarities, so as to finalize the jigsaw. The proofs of them use the usual bisimulation construction approach, by exploiting Lemmas 2, 20 and Corollary 21. The details are placed in Appendix A.

Lemma 23. (1) \sim_{nr} implies \sim_{nr}° on Π^{mp} processes. (2) \sim_{nr}° implies \sim_{hoio}° on Π^{mp} processes.

The follow-up lemma essentially fills what is left in the relationship between the strong bisimilarities.

Lemma 24. $\sim_{ho}, \sim_{nr}^\circ$, and \sim_{ctx} coincide on open and closed Π^{mp} processes.

Proof of Lemma 24. The following circular implications prove this lemma.



□

The main theorem From Lemma 24 and Lemma 13, we now have the main result of this section.

Theorem 25. *All the strong bisimilarities, that is, $\sim_{hoio}^\circ, \sim_{ho}, \sim_{ctx}, \sim_{nr}$, and \sim_{nr}° , coincide on open and closed Π^{mp} processes, and are all decidable.*

4 Axiomatization

In this section, we make an axiom system for the strong bisimilarities based on the decidability result. For simplicity, we denote by \sim the strong bisimilarity, since all the strong bisimilarities coincide. Basically, the equation set of the axiom system is composed of the extended structural congruence. Compared with the setting without parameterization [15], we have extra equations describing the application operation. We use a similar approach to prove the correctness of the axiom system.

The axiom system of [15] consists of the basic structural congruence laws and an extended distribution law *DIS*: $a(x). (P \mid \prod_{i=1}^{k-1} a(x).P) = \prod_{i=1}^k a(x).P$. Recall that the rules for application are modeled as a part of the structural congruence. To admit parameterization, we introduce the following two more laws *APP1* and *APP2*: $(\langle X \rangle P) \langle Q \rangle = P\{Q/X\}$, $(\langle x \rangle P) \langle m \rangle = P\{m/x\}$. We will show that the basic laws for the structural congruence, together with law *DIS* and moreover laws *APP1* and *APP2*, are sufficient for a complete axiom system. More specifically, we first prove that any term P has a unique *prime* decomposition $\prod_{i=1}^k P_i$, and then that any term can be simplified to a normal form (denoted as $\text{nf}(P)$) by the laws above. By the soundness of the laws, we infer $P \sim \text{nf}(P)$. Finally we prove that for any P and Q , $P \sim Q$ if and only if $\text{nf}(P) \equiv \text{nf}(Q)$.

Let \mathcal{A} be the axiom system containing *DIS*, *APP1*, *APP2* and the commutative monoid laws for parallel composition. We will prove the completeness of \mathcal{A} in the remainder of this section. We start by the cancellation property. The point of proving this property is to deem \sim as \sim_{hoio} [15], and the most involved cases are those concerning the abstractions. We provide the details in Appendix B.

Proposition 26 (Cancellation). *For all P, Q and R , if $P \mid R \sim Q \mid R$ then $P \sim Q$.*

The notion of prime processes is due to [15, 17]. A process P is *prime* if $P \not\sim 0$ and $P \sim P_1 \mid P_2$ implies $P_1 \sim 0$ or $P_2 \sim 0$. If $P \sim \prod_{i=1}^n P_i$ where each P_i is prime, we call $\prod_{i=1}^n P_i$ a *prime decomposition* of P . The following proposition states that for any process, there is a unique prime decomposition up to the strong bisimilarity and permutation of indices. Instantiating \sim as \sim_{hoio} , one can prove this proposition by induction on the size of the given process. We give the proof in Appendix B.

Proposition 27 (Unique prime decomposition). *Given a process P , if there are two prime decompositions $P \sim \prod_{i=1}^k P_i$ and $P \sim \prod_{j=1}^l Q_j$, then $n = m$ and there is a permutation $\sigma : \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\}$, such that $P_i \sim Q_{\sigma(i)}$ for each $i \in \{1, 2, \dots, n\}$.*

We write $P \rightsquigarrow Q$ if there are P' and Q' such that $P \equiv P'$, $Q \equiv Q'$, and Q' can be obtained from P' by rewriting a subterm of P' by laws *DIS*, *APP1*, *APP2* from left to right. A process P is in *normal form* if it cannot be simplified by using \rightsquigarrow . Any process P has a unique normal form up to \equiv , denoted as $\text{nf}(P)$. It is not hard to derive the following property.

Lemma 28. *If $P \rightsquigarrow Q$, then $P \sim Q$. For any P , $P \sim \text{nf}(P)$.*

Next we give a lemma crucial for the completeness proof. Its counterpart in non-parameterization setting was first presented in [15]. The proof of Lemma 29 is put in Appendix B.

Lemma 29. *If $a(X).P \sim Q \mid Q'$ ($Q, Q' \not\sim 0$), then $a(X).P \sim \prod_{i=1}^k a(X).A$ ($k > 1$) with $a(X).A$ in normal form.*

Now we can prove the completeness of the axiom system \mathcal{A} . Basically, the proof of the completeness uses a similar approach as that of [15]. The main novelty here is to accommodate the parameterization in the equation system and the corresponding parts in the induction, i.e., those parts concerning the abstraction and application for names and processes.

Lemma 30 (Completeness). *For any P, Q , if $P \sim Q$ then $\text{nf}(P) \equiv \text{nf}(Q)$.*

Proof of Lemma 30. We first show the following two properties simultaneously: **1.** If A is a prefixed process in normal form, then A is prime; **2.** For any A, B in normal form, $A \sim B$ implies $A \equiv B$. We proceed by induction on $\text{depth}(A)$. The case $\text{depth}(A) = 0$ is immediate as the only term of this size is 0. Suppose the property holds for all $\text{depth}(A) < n$ with $n \geq 1$.

(1) Assume A is of the form $a(X).A'$. Suppose A is not prime, $A \sim P_1 | P_2$. By Lemma 29, $A \sim \prod_{i=1}^k a(X).B$ with $k > 1$ and $a(X).B$ in normal form. Then $A' \sim B | \prod_{i=1}^{k-1} a(X).B$. By ind. hyp. for 2, we have $A' \equiv B | \prod_{i=1}^{k-1} a(X).B$. Then $A \equiv a(X).(B | \prod_{i=1}^{k-1} a(X).B)$, a contradiction to that A is in normal form.

(2) Suppose $A \sim B$, we proceed by a case analysis on the structure of A .

- A is X . We have that B should be the same variable X .
- A is $m(X).P$. Assume B is not prime, $B \sim P_1 | P_2$. By Lemma 29, we know $A \sim \prod_{i=1}^k a(X).Q$ with $k > 1$ and $a(X).Q$ in normal form. But according to property 1, A is prime, a contradiction. We thus have B is $m(X).Q$ with $P \sim Q$. By ind. hyp., $P \equiv Q$. We thus have $A \equiv B$.
- A is $\bar{m}(Q)$. We have that B is $\bar{m}(Q')$ with $Q \sim Q'$. By ind. hyp., $Q \equiv Q'$. We thus have $A \equiv B$.
- A is $\langle X \rangle P$. We have that B is $\langle X \rangle Q$ and $P \sim Q$. By ind. hyp., $P \equiv Q$. We thus have $A \equiv B$.
- A is $X \langle Q \rangle$. We have that B is $X \langle Q' \rangle$ and $Q \sim Q'$. By ind. hyp., $Q \equiv Q'$. We thus have $A \equiv B$.
- A is $\langle x \rangle P$. We have that B is $\langle x \rangle Q$ and $P \sim Q$. By ind. hyp., $P \equiv Q$. We thus have $A \equiv B$.
- A is $X \langle n \rangle$. We have that B is $X \langle n \rangle$, and then $A \equiv B$.
- A is $\prod_{i=1}^k P_i$ with $k > 1$ and P_i is not a parallel composition. We discuss over the possible shape of P_i .
 - If there exists j s.t. $P_j = X$, then $B \equiv X | B'$. Thus $A \equiv B$ follows by ind. hyp. on $\prod_{1 \leq i \leq k, i \neq j} P_i$ and B' .
 - If there exists j s.t. $P_j = X \langle Q \rangle$, then $B \equiv X \langle Q' \rangle | B'$ with $Q \sim Q'$ and $B' \sim \prod_{1 \leq i \leq k, i \neq j} P_i$. By ind. hyp., $A \equiv B$.
 - If there exists j s.t. $P_j = X \langle n \rangle$, then $B \equiv X \langle n \rangle | B'$ with $B' \equiv \prod_{1 \leq i \leq k, i \neq j} P_i$. By ind. hyp., $A \equiv B$.
 - If there exists j s.t. $P_j = \bar{m}(Q)$, then B must contain an output component on the same channel. We thus have $B = \bar{m}(Q') | B'$ with $Q' \sim Q$ and $B' \sim \prod_{1 \leq i \leq k, i \neq j} P_i$. By ind. hyp., $Q' \equiv Q$ and $B' \equiv \prod_{1 \leq i \leq k, i \neq j} P_i$, which implies $A \equiv B$.
 - The last case is $A = \prod_{i=1}^k m_i(X_i).P_i$. According to property 1, each component $m_i(X_i).P_i$ is prime. Similarly, $B \equiv \prod_{i=1}^l n_i(Y_i).Q_i$ and each component $n_i(Y_i).Q_i$ is prime. By Proposition 27, $k = l$ and $m_i(X_i).P_i \sim n_i(Y_i).Q_i$ for $1 \leq i \leq k$ (up to a permutation of indices). By ind. hyp. $P_i \equiv Q_i$ for all i , which finally implies $A \equiv B$.

Now for P, Q , assume $P \sim Q$. Let $A \stackrel{\text{def}}{=} \text{nf}(P)$ and $B \stackrel{\text{def}}{=} \text{nf}(Q)$. By Lemma 28, $A \sim P \sim Q \sim B$. As A, B are in normal form, have $A \equiv B$, and then $\text{nf}(P) \equiv \text{nf}(Q)$, as needed. \square

5 Algorithm for the bisimilarity checking

In this section, based on the results in the previous sections, we develop an algorithm for checking the strong bisimilarity. We utilize the tree approach proposed in [15], i.e., encoding a Π^{mp} process as a tree, normalizing this tree to be compared up-to syntax. Differently now, the tree and the normalization takes parameterization into consideration. We define a function db that assigns De Bruijn indices to variables [4, 15]. Here the variables include the ones introduced by input prefixed processes, name abstraction and process abstraction. Following [15], we introduce the representation of a term by a tree. We write $t[m_1, \dots, m_k]$ for a tree with label t and subtrees m_1, \dots, m_k .

Definition 31 (Tree representation). The tree representation of P is defined inductively as follows.

- (1) $\text{Tree}(0) = 0[]$, (2) $\text{Tree}(X) = \text{db}(X)[]$, (3) $\text{Tree}(a(X).P) = a[\text{Tree}(P)]$,
- (4) $\text{Tree}(\bar{a}(Q)) = \bar{a}^O[\text{Tree}(Q)]$, (5) $\text{Tree}(x(X).P) = \text{db}(x)[\text{Tree}(P)]$, (6) $\text{Tree}(\bar{x}(Q)) = \text{db}(x)^O[\text{Tree}(Q)]$,
- (7) $\text{Tree}(\prod_{i=1}^n P_i) = \text{par}[\text{Tree}(P_1), \dots, \text{Tree}(P_n)]$, (8) $\text{Tree}(\langle X \rangle P) = \text{abs}[\text{Tree}(P)]$, (9) $\text{Tree}(\langle P \rangle Q) = \text{app}[\text{Tree}(P), \text{Tree}(Q)]$, (10) $\text{Tree}(\langle x \rangle P) = \text{abs}[\text{Tree}(P)]$, (11) $\text{Tree}(\langle P \rangle n) = \text{app}[\text{Tree}(P), n[]]$.

The algorithm deciding the strong bisimilarity depends on the following 3 normalization steps:

Normalization: (1) In the first step, the term is rewritten by the application rules *APP1*, *APP2* if possible. (2) The second step focuses on the normalization of parallel composition. W.l.o.g., we can assume that the children of parallel composition nodes are not parallel composition nodes. After this step, every parallel composition node has at least two sorted child nodes, and none of them is 0. (3) The last step aims to apply *DIS* from left to right if possible.

Now we explain the detailed algorithms given as pseudocodes below. A tree node n has the following attributes: $n.type$ for the type of corresponding process, the values can be *zero*, *var*, *inp*, *out*, *par*, *abs*, *app*; $n.label$ for the label of the tree node; $n.numChildren$ for the number of children nodes; $n.children$ for the lists of all child nodes. The algorithm *App* realizes the application operation. It requires three parameters: n_{raw} , ind , and n_{eval} . The tree is traversed top-down and all variables from term n_{raw} are replaced with process n_{eval} if the De Bruijn index matches ind . In the process of application, if there are more than one occurrence of an abstracted variable, say X , to be replaced, there will be more than one duplications of n_{eval} . The nests of application may result in an exponential explosion on the number of tree nodes. However, we can make optimization by reusing n_{eval} , that is, each occurrence of X points to the same tree of n_{eval} . Then it is guaranteed that the space cost for normalized terms is still linear, leading to acceptable time complexity of the algorithm.

Algorithm *NS1* deals with terms for application. The tree is traversed bottom-up. Every term in the form of $\langle X \rangle P \langle Q \rangle$ or $\langle x \rangle P \langle m \rangle$ are rewritten as $P\{Q/X\}$ or $P\{m/x\}$ respectively. Terms in the form of $X \langle Q \rangle$ and $X \langle n \rangle$ remain unchanged. Algorithm *NS2* deals with parallel composition. First all zero processes are removed. Then, if attribute $numChildren$ is 0, the tree is collapsed to a zero node. If attribute $numChildren$ is 1, the tree is collapsed to its single child. After this, all children nodes are sorted. In algorithm *NS3*, the tree is traversed bottom-up to find subtrees which can apply *DIS* from left to right. Lines 11-24 decides if node n matches the pattern with the left-hand side of *DIS*. It harnesses the property that all children nodes have been sorted in normalization step 2. If it fails to match the pattern, the node n remains unchanged and the function returns in line 18 or 22. Otherwise, the term is rewritten at lines 25-26. As a consequence of Lemma 30, the following lemma shows that, if two terms are strongly bisimilar, they can be normalized to the same tree by the three normalization steps. By checking the equalities of the two trees, we can decide the strong bisimilarity between Π^{mp} terms.

Lemma 32. *Let P, Q be two terms. Let T_P, T_Q be the tree representations of P, Q respectively. Assume that T'_P, T'_Q are the normalized trees after the normalization steps 1-3. Then $P \sim Q$ if and only if $T'_P = T'_Q$.*

We now analyze the complexity of the algorithm. Given processes P and Q , let n be the sum of the number of nodes in the tree representations of P and Q . The algorithm *App* and *NS1* traverse the tree for one time and can be done in $O(n)$ time. The most time-consuming part of *NS2* is sorting, which can be done in $O(n \log(n))$ time. The algorithm *NS3* can be performed in $O(n)$ time. Therefore, bisimilarity checking takes in $O(n \log(n))$ time in total. As explained above, the space complexity is $O(n)$.

Application $App(n_{raw}, ind, n_{eval})$

<p>Require: Tree nodes n_{raw}, n_{eval}, an integer ind.</p> <p>1: if ($n_{raw}.type == 'var'$ or $n_{raw}.type == 'inp'$) and $n_{raw}.label == ind$ then</p> <p>2: $n_{raw} = n_{eval}$</p> <p>3: end if</p> <p>4: if $n_{raw}.type == 'out'$ and $n_{raw}.label == ind^O$ then</p> <p>5: $n_{raw} = n_{eval}$</p> <p>6: $n_{raw}.label = (n_{raw}.label)^O$</p>	<p>7: end if</p> <p>8: if $n_{raw}.type == 'inp'$ or $n_{raw}.type == 'abs'$ then</p> <p>9: $ind = ind + 1$</p> <p>10: end if</p> <p>11: for $i = 1$ to $n.numChildren$ do</p> <p>12: $App(n_{raw}.children[i], ind, n_{eval})$</p> <p>13: end for</p>
--	---

Normalization Step 1 NS1(n)

Require: A tree node n 1: for $i = 1$ to n .numChildren do 2: NS1(n .children[i]) 3: end for	4: if n .type == 'app' then 5: if n .children[1].type == 'abs' then 6: App(n .children[1].children[1], 1, n .children[2]) 7: end if 8: end if
---	--

Normalization Step 2 NS2(n)

Require: A tree node n 1: for $i = 1$ to n .numChildren do 2: NS2(n .children[i]) 3: end for 4: if n .type == 'par' then 5: $j = 1$ 6: for $i = 1$ to n .numChildren do 7: if n .children[i].type \neq 'zero' then 8: n .children[j] = n .children[i] 9: $j = j + 1$ 10: end if	11: end for 12: n .numChildren = $j - 1$ 13: if n .numChildren == 0 then 14: n .type = 'zero' 15: end if 16: if n .numChildren == 1 then 17: $n = n$.children[1] 18: end if 19: end if 20: sortChildren(n)
---	--

Normalization Step 3 NS3(n)

Require: A tree node n 1: for $i = 1$ to n .numChildren do 2: NS3(n .children[i]) 3: end for 4: if n .type == 'inp' then 5: $p = n$.children[1] 6: if p .type == 'par' then 7: $smallIndex = -1$ 8: $small, big = null$ 9: $pc1 = p$.children[1] 10: $pc2 = p$.children[p .numChildren] 11: if $pc1$.type == 'inp' and $pc1$.label == n .label and 12: $pc1$.children[1] == $pc2$ then 13: $small = pc2, big = pc1,$ 14: $smallIndex = p$.numChildren	14: else if $pc2$.type == 'inp' and $pc2$.label == n .label and 15: $pc2$.children[1] == $pc1$ then 16: $small = pc1, big = pc2$ 17: $smallIndex = 1$ 18: else 19: return 20: end if 21: for $i = 2$ to n .numChildren-1 do 22: if p .children[i] \neq big then 23: return 24: end if 25: end for 26: p .children[$smallIndex$] = big 27: $n = n$.children[1] 28: end if
---	--

6 Conclusion

In this paper, we have exhibited that even in presence of parameterization, which can increase the expressiveness of higher-order processes, the strong bisimilarity is still decidable for Π^{mp} . The proving approach extends the previous one for HOCORE, with several significant distinctions due to parameterization. This decidability result comes with the more powerful modelling capability of the process model, and is thus of both fundamental and practical importance to some extent. Besides, an axiom system and an algorithm are provided. They can be used as an intermediate prototype for potential application of the higher-order process model, in particular the bisimilarity checking. A further work is to try expanding the model to allow more convenient modelling capability, e.g., locations, while maintaining the decidability result. A far more challenging job is to consider the decidability of the weak bisimilarity.

Acknowledgements We are grateful to the anonymous referees for their useful comments on this paper.

References

- [1] G. Ambal, S. Lenglet & A. Schmitt (2021): *HO π in Coq*. *Journal of Automated Reasoning* 65, pp. 75–124, doi:[10.1007/s10817-020-09553-0](https://doi.org/10.1007/s10817-020-09553-0).
- [2] H. P. Barendregt (1984): *The Lambda Calculus—Its Syntax and Semantics*. North-Holland.
- [3] M. Biernacka, D. Biernacki, S. Lenglet, P. Polesiuk, D. Pous & A. Schmitt (2017): *Fully Abstract Encodings of λ -Calculus in HOcore through Abstract Machines*. In: *Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2017)*, pp. 1–12, doi:[10.1109/LICS.2017.8005118](https://doi.org/10.1109/LICS.2017.8005118).
- [4] N.G. De Bruijn (1972): *Lambda Calculus Notation with Nameless Dummies: A Tool for Automatic Formula Manipulation, with Application to the Church-Rosser Theorem*. *Indagationes Mathematicae* 34, pp. 381–392, doi:[10.1016/1385-7258\(72\)90034-0](https://doi.org/10.1016/1385-7258(72)90034-0).
- [5] M. Bundgaard, J. C. Godskesen & T. Hildebrandt (2004): *Bisimulation Congruences for Homer ? A Calculus of Higher Order Mobile Embedded Resources*. Technical Report, IT University of Copenhagen.
- [6] M. Bundgaard, J. Chr. Godskesen, B. Haagsen & H. Hüttel (2009): *Decidable Fragments of a Higher Order Calculus with Locations*. In: *Proceedings of the 15th International Workshop on Expressiveness in Concurrency (EXPRESS 2008)*, *Electronic Notes in Theoretical Computer Science* 242(1), pp. 113–138, doi:[10.1016/j.entcs.2009.06.016](https://doi.org/10.1016/j.entcs.2009.06.016).
- [7] N. Busi, M. Gabbriellini & G. Zavattaro (2009): *On the Expressive Power of Recursion, Replication and Iteration in Process Calculi*. *Mathematical Structures in Computer Science* 19, pp. 1191–1222, doi:[10.1017/S096012950999017X](https://doi.org/10.1017/S096012950999017X).
- [8] W. Charatonik & J.-M. Talbot (2001): *The Decidability of Model Checking Mobile Ambients*. In: *In Proceedings of the 15th International Workshop on Computer Science Logic (CSL 2001)*, *Lecture Notes in Computer Science* 2142, pp. 339–354, doi:[10.1007/3-540-44802-0_24](https://doi.org/10.1007/3-540-44802-0_24).
- [9] S. Christensen, Y. Hirshfeld & F. Moller (1994): *Decidable Subsets of CCS*. *The Computer Journal* 37, pp. 233–242, doi:[10.1093/comjnl/37.4.233](https://doi.org/10.1093/comjnl/37.4.233).
- [10] C. Di Giusto, J. A. Prez & G. Zavattaro (2009): *On the Expressiveness of Forwarding in Higher-Order Communication*. In: *In Proceedings of the 6th International Colloquium on Theoretical Aspects of Computing (ICTAC 2009)*, *Lecture Notes in Computer Science* 5684, pp. 155–169, doi:[10.1007/978-3-642-03466-4_10](https://doi.org/10.1007/978-3-642-03466-4_10).
- [11] P. Jančar (1995): *Undecidability of Bisimilarity for Petri Nets and Some Related Problems*. *Theoretical Computer Science* 148, pp. 281–301, doi:[10.1016/0304-3975\(95\)00037-W](https://doi.org/10.1016/0304-3975(95)00037-W).
- [12] V. Koutavas & M. Hennessy (2012): *First-order Reasoning for Higher-order Concurrency*. *Computer Languages, Systems and Structures* 38, pp. 242–277, doi:[10.1016/j.cl.2012.04.003](https://doi.org/10.1016/j.cl.2012.04.003).
- [13] A. Kučera & P. Jančar (2006): *Equivalence-checking on Infinite-state Systems: Techniques and Results*. *Theory and Practice of Logic Programming* 6, pp. 227–264, doi:[10.1017/S1471068406002651](https://doi.org/10.1017/S1471068406002651).
- [14] I. Lanese, J. A. Pérez, D. Sangiorgi & A. Schmitt (2010): *On the Expressiveness of Polyadic and Synchronous Communication in Higher-order Process Calculi*. In: *In Proceedings of 37th International Colloquium on Automata, Languages and Programming (ICALP 2010)*, *Lecture Notes in Computer Science* 6199, pp. 442–453, doi:[10.1007/978-3-642-14162-1_37](https://doi.org/10.1007/978-3-642-14162-1_37).
- [15] I. Lanese, J. A. Pérez, D. Sangiorgi & A. Schmitt (2011): *On the Expressiveness and Decidability of Higher-Order Process Calculi*. *Information and Computation* 209(2), pp. 198–226, doi:[10.1016/j.ic.2010.10.001](https://doi.org/10.1016/j.ic.2010.10.001).
- [16] R. Milner (1989): *Communication and Concurrency*. Prentice Hall.
- [17] R. Milner & F. Moller (1993): *Unique Decomposition of Processes*. *Theoretical Computer Science* 107(2), pp. 357–363, doi:[10.1016/0304-3975\(93\)90176-t](https://doi.org/10.1016/0304-3975(93)90176-t).
- [18] J.A. Prez (2010): *Higher-Order Concurrency: Expressiveness and Decidability Results*. Phd thesis, University of Bologna, Italy.
- [19] D. Sangiorgi (1992): *Expressing Mobility in Process Algebras: First-order and Higher-order Paradigms*. Phd thesis, University of Edinburgh.

- [20] D. Sangiorgi (1996): *Pi-calculus, Internal Mobility and Agent-Passing Calculi*. *Theoretical Computer Science* 167(2), doi:[10.1016/0304-3975\(96\)00075-8](https://doi.org/10.1016/0304-3975(96)00075-8). Extracts of parts of the material contained in this paper can be found in the Proceedings of TAPSOFT'95 and ICALP'95.
- [21] D. Sangiorgi & D. Walker (2001): *The Pi-calculus: a Theory of Mobile Processes*. Cambridge University Press.
- [22] A. Schmitt & J. B. Stefani (2004): *The Kell Calculus: a Family of Higher-order Distributed Process Calculi*. In: *In IST/FET International Workshop on Global Computing, Lecture Notes in Computer Science* 3267, pp. 146–178, doi:[10.1007/978-3-540-31794-4_9](https://doi.org/10.1007/978-3-540-31794-4_9).
- [23] P. Schnoebelen (2001): *Bisimulation and Other Undecidable Equivalences for Lossy Channel Systems*. In: *In Proceedings of 4th International Symposium on Theoretical Aspects of Computer Software (TACS'01), Lecture Notes in Computer Science* 2215, pp. 385–399, doi:[10.1007/3-540-45500-0_19](https://doi.org/10.1007/3-540-45500-0_19).
- [24] B. Thomsen (1993): *Plain CHOCS, a Second Generation Calculus for Higher-Order Processes*. *Acta Informatica* 30(1), pp. 1–59, doi:[10.1007/BF01200262](https://doi.org/10.1007/BF01200262).
- [25] Xian Xu (2020): *Parameterizing Higher-order Processes on Names and Processes*. *RAIRO - Theoretical Informatics and Applications* 53(3-4), pp. 153–206, doi:[10.1051/ita/2019005](https://doi.org/10.1051/ita/2019005).
- [26] Xian Xu & Wenbo Zhang (2021): *On Decidability of the Bisimilarity on Higher-order Processes with Parameterization (long version)*. Available at http://basics.sjtu.edu.cn/~xuxian/main_HO_DEC_with_appendix.pdf.
- [27] Qiang Yin, Xian Xu & Huan Long (2017): *On Parameterization of Higher-order Processes*. *International Journal of Computer Mathematics* 94(7), pp. 1451–1478, doi:[10.1080/00207160.2016.1210793](https://doi.org/10.1080/00207160.2016.1210793).

Appendix

A Proofs for Section 3

In this appendix, we provide the detailed proofs for Section 3.

Proof of Lemma 10. If $P \equiv Q$, we make a case analysis on the derivation of $P \equiv Q$. In the rest of the proof, we focus on the situation when $P \sim_{\text{hoio}}^{\circ} Q$. W.l.o.g, we assume that $\text{depth}(P) \leq \text{depth}(Q)$, and the proof is conducted by an induction on $\text{depth}(P)$. If $\text{depth}(P) = 0$, P is 0 and Q must also be 0 and $\text{depth}(P) = \text{depth}(Q)$. If $\text{depth}(P) > 0$, there are several possibilities.

- If P is a process-abstraction $\langle Y \rangle A$, then Q is a process-abstraction $\langle Y \rangle B$, and $A \sim_{\text{hoio}}^{\circ} B$. We have $\text{depth}(A) = \text{depth}(B)$ by $\text{depth}(A) < \text{depth}(P)$ and induction hypothesis. Then $\text{depth}(P) = \text{depth}(A) + 1 = \text{depth}(B) + 1 = \text{depth}(Q)$.
- If P is a name-abstraction $\langle y \rangle A$, then it is similar with the last case.
- If $P \xrightarrow{\bar{a}A} P'$, then $Q \xrightarrow{\bar{a}B} Q'$ with $A \sim_{\text{hoio}}^{\circ} B$ and $P' \sim_{\text{hoio}}^{\circ} Q'$. We have both $\text{depth}(A) = \text{depth}(B)$ and $\text{depth}(P') = \text{depth}(Q')$ by induction hypothesis. Then $\text{depth}(P) = \text{depth}(A) + \text{depth}(P') + 1 = \text{depth}(B) + \text{depth}(Q') + 1 = \text{depth}(Q)$.
- If $P \xrightarrow{a(X)} P'$, then $Q \xrightarrow{a(X)} Q'$ and $P' \sim_{\text{hoio}}^{\circ} Q'$. We have $\text{depth}(P') = \text{depth}(Q')$ by induction hypothesis. Then $\text{depth}(P) = \text{depth}(P') + 1 = \text{depth}(Q') + 1 = \text{depth}(Q)$.
- If $P \equiv X | P'$, then $Q \equiv X | Q'$ and $P' \sim_{\text{hoio}}^{\circ} Q'$. We have $\text{depth}(P') = \text{depth}(Q')$ by induction hypothesis. Then $\text{depth}(P) = \text{depth}(P') + 1 = \text{depth}(Q') + 1 = \text{depth}(Q)$.
- If $P \equiv X \langle A \rangle | P'$ or $P \equiv X \langle d \rangle | P'$, then it is similar with the last case.

□

Proof of Lemma 11. We prove these closure properties at the same time by proving the following relation to be a strong HO-IO bisimulation.

$$\mathcal{R} \stackrel{\text{def}}{=} \left\{ \begin{array}{l} (a(X).P, a(X).Q), \quad (\bar{a}(P), \bar{a}(Q)), \quad (P | R, Q | R), \\ (\langle X \rangle P, \langle X \rangle Q), \quad (\langle x \rangle P, \langle x \rangle Q), \quad (Y \langle P \rangle, Y \langle Q \rangle), \end{array} \left| P \sim_{\text{hoio}}^{\circ} Q \right. \right\} \cup \sim_{\text{hoio}}^{\circ}$$

Suppose $(P_1, Q_1) \in \mathcal{R}$, we tackle each form (P_1, Q_1) can take. Symmetric cases are skipped.

1. (P_1, Q_1) is $(a(X).P, a(X).Q)$. Then $a(X).P \xrightarrow{a(X)} P$ can be matched by $a(X).Q \xrightarrow{a(X)} Q$ and $P \mathcal{R} Q$.
2. (P_1, Q_1) is $(\bar{a}(P), \bar{a}(Q))$. Then $\bar{a}(P) \xrightarrow{\bar{a}(P)} 0$ can be matched by $\bar{a}(Q) \xrightarrow{\bar{a}(Q)} 0$, and $P \mathcal{R} Q$ and $0 \mathcal{R} 0$.
3. The cases (P_1, Q_1) takes the forms $(\langle X \rangle P, \langle X \rangle Q)$, $(\langle x \rangle P, \langle x \rangle Q)$, or $(Y \langle P \rangle, Y \langle Q \rangle)$ are trivial since they do not have actions (the semantics does not allow action inside an abstraction or application).
4. The case that (P_1, Q_1) is $(P | R, Q | R)$ is most involved, and we make further analysis. Clearly neither of $P | R$ and $Q | R$ can be an abstraction.
 - (a) Suppose $P | R \xrightarrow{\bar{a}(A)} \cdot$ in which \cdot represents a certain term.

- i. The action comes from P , i.e., $P \xrightarrow{\bar{a}(A)} P'$ and $P|R \xrightarrow{\bar{a}(A)} P'|R$. Since $P \sim_{\text{hoio}}^\circ Q$, Q matches by $Q \xrightarrow{\bar{a}(B)} Q'$ with $A \sim_{\text{hoio}}^\circ B$ and $P' \sim_{\text{hoio}}^\circ Q'$. So $Q|R \xrightarrow{\bar{a}(B)} Q'|R$. This simulation works because $A \mathcal{R} B$ and $P'|R \mathcal{R} Q'|R$ due to $P' \sim_{\text{hoio}}^\circ Q'$.
 - ii. The action comes from R , i.e., $R \xrightarrow{\bar{a}(A)} R'$ and $P|R \xrightarrow{\bar{a}(A)} P|R'$. So one simulates by $Q|R \xrightarrow{\bar{a}(A)} Q|R'$, with ARA and $P|R' \mathcal{R} Q|R'$ due to $P \sim_{\text{hoio}}^\circ Q$.
- (b) Suppose $P|R \xrightarrow{a(X)} \cdot$.
- i. The action comes from P , i.e., $P \xrightarrow{a(X)} P'$ and $P|R \xrightarrow{a(X)} P'|R$. Since $P \sim_{\text{hoio}}^\circ Q$, Q matches by $Q \xrightarrow{a(X)} Q'$ with $P' \sim_{\text{hoio}}^\circ Q'$. So $Q|R \xrightarrow{a(X)} Q'|R$. This finishes the simulation because $P'|R \mathcal{R} Q'|R$ due to $P' \sim_{\text{hoio}}^\circ Q'$.
 - ii. The action comes from R , i.e., $R \xrightarrow{a(X)} R'$ and $P|R \xrightarrow{a(X)} P|R'$. So one simulates by $Q|R \xrightarrow{a(X)} Q|R'$, with $P|R' \mathcal{R} Q|R'$ due to $P \sim_{\text{hoio}}^\circ Q$.
- (c) Suppose $P|R \equiv X|O$.
- i. The form results from P , i.e., $P \equiv X|P'$ and $P|R \equiv X|P'|R$. Since $P \sim_{\text{hoio}}^\circ Q$, we have $Q \equiv X|Q'$ with $P' \sim_{\text{hoio}}^\circ Q'$. So $Q|R \equiv X|Q'|R$. This does the simulation because $P'|R \mathcal{R} Q'|R$ due to $P' \sim_{\text{hoio}}^\circ Q'$.
 - ii. The form results from R , i.e., $R \equiv X|R'$ and $P|R \equiv X|P|R'$. Also one knows that $Q|R \equiv X|Q|R'$. This closes the simulation because $P|R' \mathcal{R} Q|R'$ due to $P \sim_{\text{hoio}}^\circ Q$.
- (d) Suppose $P|R \equiv X\langle A \rangle|O$.
- i. The form results from P , i.e., $P \equiv X\langle A \rangle|P'$ and $P|R \equiv X\langle A \rangle|P'|R$. Since $P \sim_{\text{hoio}}^\circ Q$, we have $Q \equiv X\langle B \rangle|Q'$ with $A \sim_{\text{hoio}}^\circ B$ and $P' \sim_{\text{hoio}}^\circ Q'$. So $Q|R \equiv X\langle B \rangle|Q'|R$. This does the simulation because $A \mathcal{R} B$ and $P'|R \mathcal{R} Q'|R$ due to $P' \sim_{\text{hoio}}^\circ Q'$.
 - ii. The form results from R , i.e., $R \equiv X\langle A \rangle|R'$ and $P|R \equiv X\langle A \rangle|P|R'$. Also one knows that $Q|R \equiv X\langle A \rangle|Q|R'$. This closes the simulation because ARA and $P|R' \mathcal{R} Q|R'$ due to $P \sim_{\text{hoio}}^\circ Q$.
- (e) Suppose $P|R \equiv X\langle d \rangle|O$.
- i. The form results from P , i.e., $P \equiv X\langle d \rangle|P'$ and $P|R \equiv X\langle d \rangle|P'|R$. Since $P \sim_{\text{hoio}}^\circ Q$, we have $Q \equiv X\langle d \rangle|Q'$ with $P' \sim_{\text{hoio}}^\circ Q'$. So $Q|R \equiv X\langle d \rangle|Q'|R$. This does the simulation because $P'|R \mathcal{R} Q'|R$ due to $P' \sim_{\text{hoio}}^\circ Q'$.
 - ii. The form results from R , i.e., $R \equiv X\langle d \rangle|R'$ and $P|R \equiv X\langle d \rangle|P|R'$. Also one knows that $Q|R \equiv X\langle d \rangle|Q|R'$. This closes the simulation because $P|R' \mathcal{R} Q|R'$ due to $P \sim_{\text{hoio}}^\circ Q$.

□

Proof of Lemma 14. We define \mathcal{R} to be

$$\mathcal{R} \stackrel{\text{def}}{=} \{(P\{g/m\}, Q\{g/m\}) \mid P \sim_{\text{hoio}}^\circ Q\}$$

and show it to be a strong HO-IO bisimulation up-to \equiv . Suppose $(P\{g/m\}, Q\{g/m\}) \in \mathcal{R}$. Below we verify each clause of HO-IO bisimulation up-to \equiv (?). Lemma 2 and Lemma 3 would be helpful.

1. If $P\{g/m\}$ is not an abstraction, then P is not either. Since $P \sim_{\text{hoio}}^\circ Q$, Q must not be an abstraction, neither is $Q\{g/m\}$.
2. If $P\{g/m\}$ is a process-abstraction, then it must be the case that $P \equiv \langle X \rangle M_1$ and $P\{g/m\} \equiv \langle X \rangle (M_1\{g/m\})$. Since $P \sim_{\text{hoio}}^\circ Q$, Q must be of the form $\langle X \rangle M_2$ and $M_1 \sim_{\text{hoio}}^\circ M_2$. So $Q\{g/m\} \equiv \langle X \rangle (M_2\{g/m\})$, and $M_1\{g/m\} \mathcal{R} M_2\{g/m\}$.

3. If $P\{g/m\}$ is a name-abstraction, then it must be the case that $P \equiv \langle x \rangle M_1$ and $P\{g/m\} \equiv \langle x \rangle (M_1\{g/m\})$ (we can assume neither of m and g is x up-to α -conversion). Since $P \sim_{\text{hoio}}^\circ Q$, Q must be of the form $\langle x \rangle M_2$ and $M_1 \sim_{\text{hoio}}^\circ M_2$. So $Q\{g/m\} \equiv \langle x \rangle (M_2\{g/m\})$, and $M_1\{g/m\} \mathcal{R} M_2\{g/m\}$.

4. We now tackle input action simulation, as output is similar. Suppose $P\{g/m\} \xrightarrow{a(Y)} P_1$.

In terms of Lemma 2, the action that comes from $P\{g/m\}$ has several possibilities: (1) $P \xrightarrow{h(Y)} P'$ in which h is neither m nor g , a is h , and $P_1 \equiv P'\{g/m\}$. (2) $P \xrightarrow{g(Y)} P'$, a is g , and $P_1 \equiv P'\{g/m\}$. (3) $P \xrightarrow{m(Y)} P'$, a is g , and $P_1 \equiv P'\{g/m\}$.

We focus on the last case, as the others are similar. In that case, since $P \sim_{\text{hoio}}^\circ Q$, $Q \xrightarrow{m(Y)} Q' \sim_{\text{hoio}}^\circ P'$. So $Q\{g/m\} \xrightarrow{g(Y)} Q_1 \equiv Q'\{g/m\}$. We have $P_1 \equiv P'\{g/m\} \mathcal{R} Q'\{g/m\} \equiv Q_1$ because $P' \sim_{\text{hoio}}^\circ Q'$.

5. Suppose $P\{g/m\} \xrightarrow{\bar{a}(A)} P_1$.

Like the input case, in terms of Lemma 2, the action that comes from $P\{g/m\}$ has several possibilities: (1) $P \xrightarrow{\bar{h}(A_1)} P'$ in which h is neither m nor g , a is h , A is $A_1\{g/m\}$, and $P_1 \equiv P'\{g/m\}$. (2) $P \xrightarrow{\bar{g}(A_1)} P'$, a is g , A is $A_1\{g/m\}$, and $P_1 \equiv P'\{g/m\}$. (3) $P \xrightarrow{\bar{m}(A_1)} P'$, a is g , A is $A_1\{g/m\}$, and $P_1 \equiv P'\{g/m\}$.

We focus on the last case, as the others are similar. In that case, since $P \sim_{\text{hoio}}^\circ Q$, $Q \xrightarrow{\bar{m}(B_1)} Q'$ with $A_1 \sim_{\text{hoio}}^\circ B_1$ and $P' \sim_{\text{hoio}}^\circ Q'$. So $Q\{g/m\} \xrightarrow{\bar{g}(B)} Q_1 \equiv Q'\{g/m\}$ in which $B \equiv B_1\{g/m\}$. We have $P_1 \equiv P'\{g/m\} \mathcal{R} Q'\{g/m\} \equiv Q_1$ and $A \equiv A_1\{g/m\} \mathcal{R} B_1\{g/m\} \equiv B$ because $P' \sim_{\text{hoio}}^\circ Q'$ and $A_1 \sim_{\text{hoio}}^\circ B_1$.

6. Suppose $P\{g/m\} \equiv Y|M$.

The form stems from $P \equiv Y|M_1$ where $M \equiv M_1\{g/m\}$. Since $P \sim_{\text{hoio}}^\circ Q$, $Q \equiv Y|N_1$ and $M_1 \sim_{\text{hoio}}^\circ N_1$. So $Q\{g/m\} \equiv Y|N$ where $N \equiv N_1\{g/m\}$. We have $M \equiv M_1\{g/m\} \mathcal{R} N_1\{g/m\} \equiv N$ because $M_1 \sim_{\text{hoio}}^\circ N_1$.

7. Suppose $P\{g/m\} \equiv Y\langle A \rangle | M$.

The form stems from $P \equiv Y\langle A_1 \rangle | M_1$ where $A \equiv A_1\{g/m\}$ and $M \equiv M_1\{g/m\}$. Since $P \sim_{\text{hoio}}^\circ Q$, $Q \equiv Y\langle B_1 \rangle | N_1$, and $A_1 \sim_{\text{hoio}}^\circ B_1$ and $M_1 \sim_{\text{hoio}}^\circ N_1$. So $Q\{g/m\} \equiv Y\langle B \rangle | N$ where $B \equiv B_1\{g/m\}$ and $N \equiv N_1\{g/m\}$. We have $A \equiv A_1\{g/m\} \mathcal{R} B_1\{g/m\} \equiv B$ and $M \equiv M_1\{g/m\} \mathcal{R} N_1\{g/m\} \equiv N$ because $A_1 \sim_{\text{hoio}}^\circ B_1$ and $M_1 \sim_{\text{hoio}}^\circ N_1$.

8. Suppose $P\{g/m\} \equiv Y\langle d \rangle | M$.

The form that $P\{g/m\}$ takes has a few possibilities: (1) $P \equiv Y\langle h \rangle | M_1$ where h is neither m nor g , d is h , and $M \equiv M_1\{g/m\}$. (2) $P \equiv Y\langle g \rangle | M_1$, d is g , and $M \equiv M_1\{g/m\}$. (3) $P \equiv Y\langle m \rangle | M_1$, d is g , and $M \equiv M_1\{g/m\}$.

We focus on the last case, for the others are similar. In that case, since $P \sim_{\text{hoio}}^\circ Q$, $Q \equiv Y\langle m \rangle | N_1$ and $M_1 \sim_{\text{hoio}}^\circ N_1$. So $Q\{g/m\} \equiv Y\langle g \rangle | N$ where $N \equiv N_1\{g/m\}$. We have $M \equiv M_1\{g/m\} \mathcal{R} N_1\{g/m\} \equiv N$ because $M_1 \sim_{\text{hoio}}^\circ N_1$.

□

Proof of Lemma 15. On the whole, we proceed by induction on the sum of the depth of R and the depth of P . Notice that X can be a non-abstraction, process-abstraction, or name-abstraction.

Induction Basis. The basis comprises the cases that $\text{depth}(R)$ is 0 or 1, and $\text{depth}(P)$ is 0 or 1.

One notable point is that being either 0 or Y (a free variable), R does not contribute to the action from $P\{R/X\}$ at all. In other words, the transition by $P\{R/X\}$ must result from the original process P (similar for $Q\{R/X\}$). In the case of $R \equiv Y$, it basically amounts to renaming variable X to another variable Y . Now we expand the cases for the basis.

1. R is 0. In this case, the type of X is non-abstraction. We define \mathcal{R}_1 as follows.

$$\mathcal{R}_1 \stackrel{\text{def}}{=} \{(P\{0/X\}, Q\{0/X\}) \mid P \sim_{\text{hoio}}^\circ Q\} \cup \sim_{\text{hoio}}^\circ$$

We show that \mathcal{R}_1 is a strong HO-IO bisimulation (up-to \equiv). Suppose $(P\{0/X\}, Q\{0/X\}) \in \mathcal{R}_1$.

- (a) If $P\{0/X\}$ is not an abstraction, neither is P . Since $P \sim_{\text{hoio}}^\circ Q$, Q is not an abstraction, neither is $Q\{0/X\}$.
- (b) If $P\{0/X\}$ is a process-abstraction $\langle Z \rangle P_1$, then it must be the case that P is of the form $\langle Z \rangle P_2$ and $P_1 \equiv P_2\{0/X\}$. Since $P \sim_{\text{hoio}}^\circ Q$, Q is also a process-abstraction, say $\langle Z \rangle Q_2$ with $P_2 \sim_{\text{hoio}}^\circ Q_2$. Then $Q\{0/X\}$ is a process-abstraction $\langle Z \rangle Q_1$ in which $Q_1 \equiv Q_2\{0/X\}$. So we have

$$P_1 \equiv P_2\{0/X\} \mathcal{R}_1 Q_2\{0/X\} \equiv Q_1$$

because $P_2 \sim_{\text{hoio}}^\circ Q_2$.

- (c) If $P\{0/X\}$ is a name-abstraction $\langle z \rangle P_1$, then it must be the case that P is of the form $\langle z \rangle P_2$ and $P_1 \equiv P_2\{0/X\}$. Since $P \sim_{\text{hoio}}^\circ Q$, Q is also a name-abstraction, say $\langle z \rangle Q_2$ with $P_2 \sim_{\text{hoio}}^\circ Q_2$. Then $Q\{0/X\}$ is a process-abstraction $\langle z \rangle Q_1$ in which $Q_1 \equiv Q_2\{0/X\}$. So we have

$$P_1 \equiv P_2\{0/X\} \mathcal{R}_1 Q_2\{0/X\} \equiv Q_1$$

because $P_2 \sim_{\text{hoio}}^\circ Q_2$.

- (d) If $P\{0/X\} \xrightarrow{a(Z)} P_1$, then it must stem from P . That is, $P \xrightarrow{a(Z)} P'$ and $P_1 \equiv P'\{0/X\}$. Since $P \sim_{\text{hoio}}^\circ Q$, $Q \xrightarrow{a(Z)} Q'$ and $P' \sim_{\text{hoio}}^\circ Q'$. So $Q\{0/X\} \xrightarrow{a(Z)} Q_1 \equiv Q'\{0/X\}$. Hence we have

$$P_1 \equiv P'\{0/X\} \mathcal{R}_1 Q'\{0/X\} \equiv Q_1$$

because $P' \sim_{\text{hoio}}^\circ Q'$.

- (e) If $P\{0/X\} \xrightarrow{\bar{a}(A_1)} P_1$, then it must stem from P . That is, $P \xrightarrow{\bar{a}(A)} P'$, and $A_1 \equiv A\{0/X\}$ and $P_1 \equiv P'\{0/X\}$. Since $P \sim_{\text{hoio}}^\circ Q$, $Q \xrightarrow{\bar{a}(B)} Q'$, and $A \sim_{\text{hoio}}^\circ B$ and $P' \sim_{\text{hoio}}^\circ Q'$. So $Q\{0/X\} \xrightarrow{\bar{a}(B_1)} Q_1$, where $B_1 \equiv B\{0/X\}$ and $Q_1 \equiv Q'\{0/X\}$. Hence we have

$$\begin{aligned} A_1 \equiv A\{0/X\} \mathcal{R}_1 B\{0/X\} &\equiv B_1 \\ P_1 \equiv P'\{0/X\} \mathcal{R}_1 Q'\{0/X\} &\equiv Q_1 \end{aligned}$$

because $A \sim_{\text{hoio}}^\circ B$ and $P' \sim_{\text{hoio}}^\circ Q'$.

- (f) If $P\{0/X\} \equiv Y \mid P_1$, then it must stem from P as follows: $P \equiv Y \mid P'$, and $P_1 \equiv P'\{0/X\}$. Since $P \sim_{\text{hoio}}^\circ Q$, $Q \equiv Y \mid Q'$, with $P' \sim_{\text{hoio}}^\circ Q'$. Thus $Q\{0/X\} \equiv Y \mid Q_1$, where $Q_1 \equiv Q'\{0/X\}$. Hence we have

$$P_1 \equiv P'\{0/X\} \mathcal{R}_1 Q'\{0/X\} \equiv Q_1$$

because $P' \sim_{\text{hoio}}^\circ Q'$.

- (g) If $P\{0/X\} \equiv Y\langle A_1 \rangle | P_1$, then it must stem from P as follows: $P \equiv Y\langle A \rangle | P'$, and $A_1 \equiv A\{0/X\}$ and $P_1 \equiv P'\{0/X\}$. Since $P \sim_{\text{hoio}}^\circ Q$, $Q \equiv Y\langle B \rangle | Q'$, with $A \sim_{\text{hoio}}^\circ B$ and $P' \sim_{\text{hoio}}^\circ Q'$. So $Q\{0/X\} \equiv Y\langle B_1 \rangle | Q_1$, where $B_1 \equiv B\{0/X\}$ and $Q_1 \equiv Q'\{0/X\}$. Hence we have

$$\begin{aligned} A_1 &\equiv A\{0/X\} \mathcal{R}_1 B\{0/X\} \equiv B_1 \\ P_1 &\equiv P'\{0/X\} \mathcal{R}_1 Q'\{0/X\} \equiv Q_1 \end{aligned}$$

because $A \sim_{\text{hoio}}^\circ B$ and $P' \sim_{\text{hoio}}^\circ Q'$.

- (h) If $P\{0/X\} \equiv Y\langle d \rangle | P_1$, then it must stem from P as follows: $P \equiv Y\langle d \rangle | P'$, and $P_1 \equiv P'\{0/X\}$. Since $P \sim_{\text{hoio}}^\circ Q$, $Q \equiv Y\langle d \rangle | Q'$, with $P' \sim_{\text{hoio}}^\circ Q'$. Thus $Q\{0/X\} \equiv Y\langle d \rangle | Q_1$, where $Q_1 \equiv Q'\{0/X\}$. Hence we have

$$P_1 \equiv P'\{0/X\} \mathcal{R}_1 Q'\{0/X\} \equiv Q_1$$

because $P' \sim_{\text{hoio}}^\circ Q'$.

2. R is Y . In this case, X can be of any type, but the discussion is similar. To taste the argument, we define \mathcal{R}_2 as follows.

$$\mathcal{R}_2 \stackrel{\text{def}}{=} \{(P\{Y/X\}, Q\{Y/X\}) | P \sim_{\text{hoio}}^\circ Q\} \cup \sim_{\text{hoio}}^\circ$$

We show that \mathcal{R}_2 is a strong HO-IO bisimulation (up-to \equiv). Suppose $(P\{Y/X\}, Q\{Y/X\}) \in \mathcal{R}_2$.

- (a) If $P\{Y/X\}$ is not an abstraction, neither is P . Since $P \sim_{\text{hoio}}^\circ Q$, Q is not an abstraction, neither is $Q\{Y/X\}$.
- (b) If $P\{Y/X\}$ is a process-abstraction $\langle Z \rangle P_1$, then it must be the case that P is of the form $\langle Z \rangle P_2$ and $P_1 \equiv P_2\{Y/X\}$. Since $P \sim_{\text{hoio}}^\circ Q$, Q is also a process-abstraction, say $\langle Z \rangle Q_2$ with $P_2 \sim_{\text{hoio}}^\circ Q_2$. Then $Q\{Y/X\}$ is a process-abstraction $\langle Z \rangle Q_1$ in which $Q_1 \equiv Q_2\{Y/X\}$. So we have

$$P_1 \equiv P_2\{Y/X\} \mathcal{R}_2 Q_2\{Y/X\} \equiv Q_1$$

because $P_2 \sim_{\text{hoio}}^\circ Q_2$.

- (c) If $P\{Y/X\}$ is a name-abstraction $\langle z \rangle P_1$, then it must be the case that P is of the form $\langle z \rangle P_2$ and $P_1 \equiv P_2\{Y/X\}$. Since $P \sim_{\text{hoio}}^\circ Q$, Q is also a name-abstraction, say $\langle z \rangle Q_2$ with $P_2 \sim_{\text{hoio}}^\circ Q_2$. Then $Q\{Y/X\}$ is a process-abstraction $\langle z \rangle Q_1$ in which $Q_1 \equiv Q_2\{Y/X\}$. So we have

$$P_1 \equiv P_2\{Y/X\} \mathcal{R}_2 Q_2\{Y/X\} \equiv Q_1$$

because $P_2 \sim_{\text{hoio}}^\circ Q_2$.

- (d) If $P\{Y/X\} \xrightarrow{a(Z)} P_1$, then it must stem from P . That is, $P \xrightarrow{a(Z)} P'$ and $P_1 \equiv P'\{Y/X\}$. Since $P \sim_{\text{hoio}}^\circ Q$, $Q \xrightarrow{a(Z)} Q'$ and $P' \sim_{\text{hoio}}^\circ Q'$. So $Q\{Y/X\} \xrightarrow{a(Z)} Q_1 \equiv Q'\{Y/X\}$. Hence we have

$$P_1 \equiv P'\{Y/X\} \mathcal{R}_2 Q'\{Y/X\} \equiv Q_1$$

because $P' \sim_{\text{hoio}}^\circ Q'$.

- (e) If $P\{Y/X\} \xrightarrow{\bar{a}(A_1)} P_1$, then it must stem from P . That is, $P \xrightarrow{\bar{a}(A)} P'$, and $A_1 \equiv A\{Y/X\}$ and $P_1 \equiv P'\{Y/X\}$. Since $P \sim_{\text{hoio}}^{\circ} Q$, $Q \xrightarrow{\bar{a}(B)} Q'$, and $A \sim_{\text{hoio}}^{\circ} B$ and $P' \sim_{\text{hoio}}^{\circ} Q'$. So $Q\{Y/X\} \xrightarrow{\bar{a}(B_1)} Q_1$, where $B_1 \equiv B\{Y/X\}$ and $Q_1 \equiv Q'\{Y/X\}$. Hence we have

$$\begin{aligned} A_1 &\equiv A\{Y/X\} \mathcal{R}_2 B\{Y/X\} \equiv B_1 \\ P_1 &\equiv P'\{Y/X\} \mathcal{R}_2 Q'\{Y/X\} \equiv Q_1 \end{aligned}$$

because $A \sim_{\text{hoio}}^{\circ} B$ and $P' \sim_{\text{hoio}}^{\circ} Q'$.

- (f) We consider $P\{Y/X\} \equiv Y|P_1$ in which the Y in $Y|P_1$ results from the substitution $\{Y/X\}$. The other possibilities are similar.

So if $P\{Y/X\} \equiv Y|P_1$, then it must stem from P as follows: $P \equiv X|P'$, and $P_1 \equiv P'\{Y/X\}$. Since $P \sim_{\text{hoio}}^{\circ} Q$, $Q \equiv X|Q'$, with $P' \sim_{\text{hoio}}^{\circ} Q'$. Thus $Q\{Y/X\} \equiv Y|Q_1$, where $Q_1 \equiv Q'\{Y/X\}$. Hence we have

$$P_1 \equiv P'\{Y/X\} \mathcal{R}_2 Q'\{Y/X\} \equiv Q_1$$

because $P' \sim_{\text{hoio}}^{\circ} Q'$.

- (g) We consider $P\{Y/X\} \equiv Y\langle A \rangle|P_1$ in which the Y in $Y\langle A \rangle$ results from the substitution $\{Y/X\}$. The other possibilities are similar.

So if $P\{Y/X\} \equiv Y\langle A \rangle|P_1$, then it must stem from P as follows: $P \equiv X\langle A \rangle|P'$, and $A_1 \equiv A\{Y/X\}$ and $P_1 \equiv P'\{Y/X\}$. Since $P \sim_{\text{hoio}}^{\circ} Q$, $Q \equiv X\langle B \rangle|Q'$, with $A \sim_{\text{hoio}}^{\circ} B$ and $P' \sim_{\text{hoio}}^{\circ} Q'$. Thus $Q\{Y/X\} \equiv Y\langle B \rangle|Q_1$, where $B_1 \equiv B\{Y/X\}$ and $Q_1 \equiv Q'\{Y/X\}$. Hence we have

$$\begin{aligned} A_1 &\equiv A\{Y/X\} \mathcal{R}_2 B\{Y/X\} \equiv B_1 \\ P_1 &\equiv P'\{Y/X\} \mathcal{R}_2 Q'\{Y/X\} \equiv Q_1 \end{aligned}$$

because $A \sim_{\text{hoio}}^{\circ} B$ and $P' \sim_{\text{hoio}}^{\circ} Q'$.

- (h) There are several subcases. We consider $P\{Y/X\} \equiv Y\langle d \rangle|P_1$ in which the Y in $Y\langle d \rangle$ results from the substitution $\{Y/X\}$. The other possibilities are similar.

So if $P\{Y/X\} \equiv Y\langle d \rangle|P_1$, then it must stem from P as follows: $P \equiv X\langle d \rangle|P'$, and $P_1 \equiv P'\{Y/X\}$. Since $P \sim_{\text{hoio}}^{\circ} Q$, $Q \equiv X\langle d \rangle|Q'$, with $P' \sim_{\text{hoio}}^{\circ} Q'$. Thus $Q\{Y/X\} \equiv Y\langle d \rangle|Q_1$, where $Q_1 \equiv Q'\{Y/X\}$. Hence we have

$$P_1 \equiv P'\{Y/X\} \mathcal{R}_2 Q'\{Y/X\} \equiv Q_1$$

because $P' \sim_{\text{hoio}}^{\circ} Q'$.

3. R is $Y\langle e \rangle$. This case is similar to the previous case. Actually it is simpler because confining to the form $Y\langle e \rangle$ (which can be somehow imagined as a special meta-variable, informally) effects to reduce the cases-analysis in establishing the strong HO-IO bisimulation.
4. P is 0, Y , $Y\langle e \rangle$, X , or $X\langle e \rangle$. In all these cases, the premise $P \sim_{\text{hoio}}^{\circ} Q$ regulates that Q must accordingly take the same forms 0, Y , $Y\langle e \rangle$, X , or $X\langle e \rangle$ respectively. So the result follows in a straightforward fashion.

Induction step. Assume the result (of this lemma) for depth no greater than the natural number n and prove it for $n+1$. Overall, still we have several situations depending what X is: non-abstraction, process-abstraction, or name-abstraction. The case for process-abstraction is where we may need the

induction hypothesis. For example, when R is substituted in for X , the sub-terms of the forms $R\langle A \rangle \sim_{\text{hoio}}^\circ R\langle B \rangle$ in which $A \sim_{\text{hoio}}^\circ B$ may call for induction, say, R takes certain shape like $\langle Z \rangle(Z\langle R' \rangle)$. Otherwise we can simply apply congruence properties (Lemma 11) and also Lemma 14.

First we establish the result for guarded X , and then extend to the general situation. We define \mathcal{R} as

$$\mathcal{R} \stackrel{\text{def}}{=} \{(P\{R/X\} | O_1, Q\{R/X\} | O_2) \mid P \sim_{\text{hoio}}^\circ Q, O_1 \sim_{\text{hoio}}^\circ O_2\} \cup \sim_{\text{hoio}}^\circ$$

and show it to be a strong HO-IO bisimulation up-to \equiv . Suppose $(P\{R/X\} | O_1, Q\{R/X\} | O_2) \in \mathcal{R}$. We notice that neither $P\{R/X\} | O_1$ nor $Q\{R/X\} | O_2$ can be an abstraction. In order to prove the result, we verify each clause of HO-IO bisimulation up-to \equiv , in which procedure Lemma 2 and Lemma 3 may be called. The bisimulation checks of the requirements of HO-IO bisimulation up-to \equiv are very much alike so we shall focus on the input case. Importantly, we make use of 'guarded' meaning that X is in an un-firable position.

- We focus on input action simulation, as output and the others are similar. Now suppose we have $P\{R/X\} | O_1 \xrightarrow{a(Y)}$. Up-to α -conversion, we can assume that Y is not X , since the substitution $P\{R/X\}$ is trying to replace those free occurrence of X in P .

(a) The action comes from $P\{R/X\}$.

That is, $P\{R/X\} \xrightarrow{a(Y)} P'$ and $P\{R/X\} | O_1 \xrightarrow{a(Y)} P' | O_1$. Since X is guarded in P , by Lemma 2, $P \xrightarrow{a(Y)} P_1$ and $P' \equiv P_1\{R/X\}$. Because $P \sim_{\text{hoio}}^\circ Q$, $Q \xrightarrow{a(Y)} Q_1$ with $P_1 \sim_{\text{hoio}}^\circ Q_1$. So by Lemma 2, $Q\{R/X\} \xrightarrow{a(Y)} Q_1\{R/X\} \stackrel{\text{def}}{=} Q'$ and $Q\{R/X\} | O_2 \xrightarrow{a(Y)} Q' | O_2$. We want to show that $P' | O_1$ and $Q' | O_2$ are related by \mathcal{R} , i.e., they can be rewritten into the shapes requested by \mathcal{R} . By Lemma 3, there are three possibilities.

- X is non-abstraction, i.e., it is supposed to be instantiated by a term that is not an abstraction. In this case,

$$\begin{aligned} P_1 &\equiv P'_1 | \Pi_{i=1}^n X \\ Q_1 &\equiv Q'_1 | \Pi_{i=1}^m X \end{aligned}$$

where X is guarded in P'_1 and Q'_1 .

Since $P_1 \sim_{\text{hoio}}^\circ Q_1$, for the following two expressions,

$$\begin{aligned} P'_1 | \Pi_{i=1}^n X \\ Q'_1 | \Pi_{i=1}^m X \end{aligned}$$

besides $P'_1 \sim_{\text{hoio}}^\circ Q'_1$ (in which X is guarded), it must be the case that $n=m$. Thus we know

$$\Pi_{i=1}^n R \equiv \Pi_{i=1}^m R$$

Hence, we have due to $P'_1 \sim_{\text{hoio}}^\circ Q'_1$ (in which X is guarded) that

$$\begin{aligned} P'_1\{R/X\} | \Pi_{i=1}^n R | O_1 \\ \mathcal{R} \\ Q'_1\{R/X\} | \Pi_{i=1}^m R | O_2 \end{aligned}$$

That is,

$$P' | O_1 \equiv \mathcal{R} \equiv Q' | O_2$$

This completes the simulation.

- ii. X is process-abstraction, i.e., it is supposed to be instantiated by a term that is a process-abstraction. In this case,

$$\begin{aligned} P_1 &\equiv P'_1 | \Pi_{i=1}^n X \langle A_i \rangle \\ Q_1 &\equiv Q'_1 | \Pi_{i=1}^m X \langle B_i \rangle \end{aligned}$$

where X is guarded in P'_1 and Q'_1 .

For the following two expressions,

$$\begin{aligned} P'_1 | \Pi_{i=1}^n X \langle A_i \rangle \\ Q'_1 | \Pi_{i=1}^m X \langle B_i \rangle \end{aligned}$$

besides $P'_1 \sim_{\text{hoio}}^\circ Q'_1$ (in which X is guarded), it must be the case that $n=m$ and $A_i \sim_{\text{hoio}}^\circ B_i$ for all natural number i in the range of 1 through n . Thus we infer for each i

$$R \langle A_i \{R/X\} \rangle \sim_{\text{hoio}}^\circ R \langle B_i \{R/X\} \rangle$$

by means of congruence properties and induction hypothesis. We notice that the induction hypothesis is called for A_i and B_i . We also note that here one may need induction hypothesis because process-abstraction is capable of using part of R (i.e., its partial code) when being instantiated by some A_i (and B_i), and this in turn brings us to the very first situation. We stress that the induction step applies because the structure of A_i is relatively diminished with respect to P or the structure of R is consumed during the instantiation. So this yields

$$\Pi_{i=1}^n R \langle A_i \{R/X\} \rangle \sim_{\text{hoio}}^\circ \Pi_{i=1}^m R \langle B_i \{R/X\} \rangle$$

Hence from this equality, we have due to $P'_1 \sim_{\text{hoio}}^\circ Q'_1$ (in which X is guarded) that

$$\begin{aligned} P'_1 \{R/X\} | \Pi_{i=1}^n R \langle A_i \{R/X\} \rangle | O_1 \\ \mathcal{R} \\ Q'_1 \{R/X\} | \Pi_{i=1}^m R \langle B_i \{R/X\} \rangle | O_2 \end{aligned}$$

That is,

$$P' | O_1 \equiv \mathcal{R} \equiv Q' | O_2$$

which closes the simulation.

- iii. X is name-abstraction, i.e., it is supposed to be instantiated by a term that is a name-abstraction. In this case,

$$\begin{aligned} P_1 &\equiv P'_1 | \Pi_{i=1}^n X \langle d_i \rangle \\ Q_1 &\equiv Q'_1 | \Pi_{i=1}^m X \langle e_i \rangle \end{aligned}$$

where X is guarded in P'_1 and Q'_1 .

For the following two expressions,

$$\begin{aligned} P'_1 | \Pi_{i=1}^n X \langle d_i \rangle \\ Q'_1 | \Pi_{i=1}^m X \langle e_i \rangle \end{aligned}$$

besides $P'_1 \sim_{\text{hoio}}^\circ Q'_1$ (in which X is guarded), it must be the case that $n=m$ and $d_i=e_i$ for all natural number i in the range of 1 through n . Thus we know for each i

$$R \langle d_i \rangle \equiv R \langle e_i \rangle$$

and thus

$$\Pi_{i=1}^n R \langle d_i \rangle \equiv \Pi_{i=1}^m R \langle e_i \rangle$$

Hence, we have due to $P'_1 \sim_{\text{hoio}}^\circ Q'_1$ (in which X is guarded) that

$$\begin{aligned} P'_1 \{R/X\} | \Pi_{i=1}^n R \langle d_i \rangle | O_1 \\ \mathcal{R} \\ Q'_1 \{R/X\} | \Pi_{i=1}^m R \langle e_i \rangle | O_2 \end{aligned}$$

That is,

$$P' | O_1 \equiv \mathcal{R} \equiv Q' | O_2$$

which completes the simulation.

- (b) The action comes from O_1 . That is, $O_1 \xrightarrow{a(Y)} O'_1$ and $P\{R/X\} | O_1 \xrightarrow{a(Y)} P\{R/X\} | O'_1$. Since $O_1 \sim_{\text{hoio}}^\circ O_2$, $O_2 \xrightarrow{a(Y)} O'_2$ with $O'_1 \sim_{\text{hoio}}^\circ O'_2$. So we have $Q\{R/X\} | O_2 \xrightarrow{a(Y)} Q\{R/X\} | O'_2$, and moreover

$$P\{R/X\} | O'_1 \mathcal{R} Q\{R/X\} | O'_2$$

as required.

Now we deal with the situation when X may occur unguarded. We stress that the entire proof is by induction on total depth of R and P . The detailed arguments are very much like the case we have done for the guarded situation, except that we are directly working on \sim_{hoio}° instead of certain constructed bisimulation \mathcal{R} . Using Lemma 3, we have three possibilities.

1. X is non-abstraction, i.e., it is supposed to be instantiated by a term that is not an abstraction. In this case,

$$\begin{aligned} P &\equiv P'' | \Pi_{i=1}^n X \\ Q &\equiv Q'' | \Pi_{i=1}^m X \end{aligned}$$

where X is guarded in P'' and Q'' .

Since $P \sim_{\text{hoio}}^\circ Q$, for the following two expressions,

$$\begin{aligned} P'' | \Pi_{i=1}^n X \\ Q'' | \Pi_{i=1}^m X \end{aligned}$$

it must be that $P'' \sim_{\text{hoio}}^\circ Q''$ (in which X is guarded), and $n=m$. Thus we know from the previous discussion that $P''\{R/X\} \sim_{\text{hoio}}^\circ Q''\{R/X\}$, and moreover

$$\Pi_{i=1}^n R \equiv \Pi_{i=1}^m R$$

Hence by congruence properties, we have

$$P''\{R/X\} | \Pi_{i=1}^n R \sim_{\text{hoio}}^\circ Q''\{R/X\} | \Pi_{i=1}^m R$$

2. X is process-abstraction, i.e., it is supposed to be instantiated by a term that is a process-abstraction.

In this case,

$$\begin{aligned} P &\equiv P'' \mid \prod_{i=1}^n X \langle A_i \rangle \\ Q &\equiv Q'' \mid \prod_{i=1}^m X \langle B_i \rangle \end{aligned}$$

where X is guarded in P'' and Q'' .

Since $P \sim_{\text{hoio}}^\circ Q$, for the following two expressions,

$$\begin{aligned} P'' \mid \prod_{i=1}^n X \langle A_i \rangle \\ Q'' \mid \prod_{i=1}^m X \langle B_i \rangle \end{aligned}$$

it must be the case that $P'' \sim_{\text{hoio}}^\circ Q''$ (in which X is guarded), $n=m$, and $A_i \sim_{\text{hoio}}^\circ B_i$ for all natural number i in the range of 1 through n . Thus we know from the previous discussion that $P''\{R/X\} \sim_{\text{hoio}}^\circ Q''\{R/X\}$, and moreover for each i

$$R \langle A_i \{R/X\} \rangle \sim_{\text{hoio}}^\circ R \langle B_i \{R/X\} \rangle$$

by means of congruence properties and/or induction hypothesis. This yields

$$\prod_{i=1}^n R \langle A_i \{R/X\} \rangle \sim_{\text{hoio}}^\circ \prod_{i=1}^m R \langle B_i \{R/X\} \rangle$$

From this equality, we have by congruence that

$$P''\{R/X\} \mid \prod_{i=1}^n R \langle A_i \{R/X\} \rangle \sim_{\text{hoio}}^\circ Q''\{R/X\} \mid \prod_{i=1}^m R \langle B_i \{R/X\} \rangle$$

3. X is name-abstraction, i.e., it is supposed to be instantiated by a term that is a name-abstraction.

In this case,

$$\begin{aligned} P &\equiv P'' \mid \prod_{i=1}^n X \langle d_i \rangle \\ Q &\equiv Q'' \mid \prod_{i=1}^m X \langle e_i \rangle \end{aligned}$$

where X is guarded in P'' and Q'' .

Since $P \sim_{\text{hoio}}^\circ Q$, for the following two expressions,

$$\begin{aligned} P'' \mid \prod_{i=1}^n X \langle d_i \rangle \\ Q'' \mid \prod_{i=1}^m X \langle e_i \rangle \end{aligned}$$

it must be that $P'' \sim_{\text{hoio}}^\circ Q''$ (in which X is guarded), $n=m$, and $d_i=e_i$ for all natural number i in the range of 1 through n . Thus we know from the previous discussion that $P''\{R/X\} \sim_{\text{hoio}}^\circ Q''\{R/X\}$, and moreover for each i

$$R \langle d_i \rangle \equiv R \langle e_i \rangle$$

and thus

$$\prod_{i=1}^n R \langle d_i \rangle \equiv \prod_{i=1}^m R \langle e_i \rangle$$

Hence, we have by congruence that

$$P''\{R/X\} \mid \prod_{i=1}^n R \langle d_i \rangle \sim_{\text{hoio}}^\circ Q''\{R/X\} \mid \prod_{i=1}^m R \langle e_i \rangle$$

The entire proof is now completed. □

Proof of Lemma 18. We prove this lemma in two directions.

- The "only if" direction.

The initial n consecutive input actions

$$h(X_1) \cdots h(X_n) \cdot P \xrightarrow{h(R_1)} \cdots \xrightarrow{h(R_n)} P\{\tilde{R}/\tilde{X}\}$$

can be matched by

$$h(X_1) \cdots h(X_n) \cdot Q \xrightarrow{h(R_1)} \cdots \xrightarrow{h(R_n)} Q\{\tilde{R}/\tilde{X}\}$$

to reach two bisimulation terms $P\{\tilde{R}/\tilde{X}\} \sim_{\text{ho}} Q\{\tilde{R}/\tilde{X}\}$ for any \tilde{R} . So the result follows. More specifically, one can establish the following bisimulation relation as represented.

$$\begin{aligned} & \{(h(X_1) \cdots h(X_n) \cdot P, h(X_1) \cdots h(X_n) \cdot Q) \mid P\{\tilde{R}/\tilde{X}\} \sim_{\text{ho}} Q\{\tilde{R}/\tilde{X}\}\} \\ \cup & \left\{ (P_i, Q_i) \mid \begin{array}{l} h(X_1) \cdots h(X_n) \cdot P \xrightarrow{h(R_1)} \cdots \xrightarrow{h(R_i)} P_i, \\ h(X_1) \cdots h(X_n) \cdot Q \xrightarrow{h(R_1)} \cdots \xrightarrow{h(R_i)} Q_i, \end{array} \quad i = 1, \dots, n-1 \right\} \\ \cup & \sim_{\text{ho}} \end{aligned}$$

- The "if" direction. From the assumption

$h(X_1) \cdots h(X_n) \cdot P \sim_{\text{ho}} h(X_1) \cdots h(X_n) \cdot Q$, we infer that

$$h(X_1) \cdots h(X_n) \cdot P \xrightarrow{h(R_1)} \cdots \xrightarrow{h(R_n)} P\{\tilde{R}/\tilde{X}\}$$

must be matched by the following.

$$h(X_1) \cdots h(X_n) \cdot Q \xrightarrow{h(R_1)} \cdots \xrightarrow{h(R_n)} Q\{\tilde{R}/\tilde{X}\}$$

So we conclude that $P\{\tilde{R}/\tilde{X}\} \sim_{\text{ho}} Q\{\tilde{R}/\tilde{X}\}$. □

Proof of Lemma 19. We prove two directions.

\sim_{ho} is a strong HO-IO bisimulation. Suppose $P \sim_{\text{ho}} Q$ to prove $P \sim_{\text{hoio}}^\circ Q$. We show that the following relation \mathcal{R}_1 is a strong HO-IO bisimulation.

$$\mathcal{R}_1 \stackrel{\text{def}}{=} \left\{ (P, Q) \mid \begin{array}{l} P\{\tilde{T}/\tilde{X}\} \sim_{\text{ho}} Q\{\tilde{T}/\tilde{X}\}, \\ \text{where } T_i \stackrel{\text{def}}{=} \begin{cases} \text{Tr}_{m_i} & \text{if } X_i \text{ is not of an abstraction type} \\ \text{Tr}_{m_i}^D & \text{if } X_i \text{ is of a process-abstraction type} \\ \text{Tr}_{m_i}^{D,d} & \text{if } X_i \text{ is of a name-abstraction type} \end{cases} \\ \text{and the names } m_i \text{ in } \tilde{m} \text{ are fresh} \end{array} \right\}$$

W.l.o.g., in \mathcal{R}_1 we can assume \tilde{X} to comprise/be all the unguarded free variables in P and Q , that is, the remaining free variables in $P\{\tilde{T}/\tilde{X}\}$ and $Q\{\tilde{T}/\tilde{X}\}$ are all guarded (in other words, P and Q are in a somewhat *saturated* form concerning guarded variables). This makes sense because \sim_{ho} is closed under substitution by its definition of extension to open terms.

Suppose $P \mathcal{R}_1 Q$, and $P \xrightarrow{\lambda} P'$. Let \tilde{Y} be the free *guarded* variables of $P\{\tilde{T}/\tilde{X}\}$ and $Q\{\tilde{T}/\tilde{X}\}$, and \tilde{R} be closed terms. We make a case analysis.

1. P is not an abstraction. Trivial.
2. P is a name-abstraction. Trivial.
3. P is a process-abstraction. This case holds because \sim_{ho} is closed under substitution by its definition concerning process-abstraction and open terms (and Lemma 18 as well), and the variables are guarded. Specifically, assume that P is $\langle Z \rangle P'$ in which Z is unguarded in P' (the case Z is guarded is simpler). We consider the case that Z is not of an abstraction type, and the other cases are similar (the only difference is the form of the trigger to replace the abstracted variable corresponding to the type of the variable).

Since $P \sim_{\text{ho}} Q$, Q is also a process abstraction, say, $\langle Z \rangle Q'$, and moreover $P'\{R/Z\} \sim_{\text{ho}} Q'\{R/Z\}$ for every closed R . Thus taking R as $\text{Tr}_{m'}$ (m' is fresh) (depending on the type of Z , one may choose the other two triggers in terms of the type of Z respectively), we have

$$P'\{\text{Tr}_{m'}/Z\}\{\tilde{T}/\tilde{X}\} \sim_{\text{ho}} Q'\{\text{Tr}_{m'}/Z\}\{\tilde{T}/\tilde{X}\}$$

Hence we derive $P' \mathcal{R}_1 Q'$ as desired.

4. Output. We first note that the output clauses of strong HO bisimulations and strong HO-IO bisimulations are the same. Specifically, suppose $P \xrightarrow{\bar{a}(A)} P'$ and we argue that Q can match as requested by the strong HO-IO bisimulations. By Lemma 2, we know that

$$P\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \xrightarrow{\bar{a}(A\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\})} P'\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\}$$

for every closed \tilde{R} .

Since $P\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \sim_{\text{ho}} Q\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\}$ (this is by the definition of \sim_{ho} over open terms), we have the bisimulation as

$$\begin{aligned} Q\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} &\xrightarrow{\bar{a}(B_1)} Q_1 \equiv Q'\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \sim_{\text{ho}} P'\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \\ B_1 &\equiv B\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \sim_{\text{ho}} A\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \end{aligned}$$

By Lemma 2 and because \tilde{Y} are guarded, we have

$$Q\{\tilde{T}/\tilde{X}\} \xrightarrow{\bar{a}(B\{\tilde{T}/\tilde{X}\})} Q'\{\tilde{T}/\tilde{X}\}$$

and moreover $P'\{\tilde{T}/\tilde{X}\} \sim_{\text{ho}} Q'\{\tilde{T}/\tilde{X}\}$. Using Lemma 2 once again since \tilde{T} only hold fresh names \tilde{m} , we have $Q \xrightarrow{\bar{a}(B)} Q'$. This also entails that $P' \mathcal{R}_1 Q'$ together with $A \mathcal{R}_1 B$.

5. Input. Suppose $P \xrightarrow{a(X)} P'$ and we argue that Q can match as required by the strong HO-IO bisimulations. Notice that no instantiation of the variable held by the input is needed, so there would (possibly) be a new process variable (i.e., X); this can be handled in the space of \sim_{ho} as in the definitions of it and \mathcal{R}_1 , by following the definitional bisimulation on closed terms. The overall approach of analysis is similar to the output case.

By Lemma 2, we know that

$$P\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \xrightarrow{a(X)} P'\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\}$$

for every closed \tilde{R} .

Since $P\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \sim_{\text{ho}} Q\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\}$ (this is by the definition of \sim_{ho} over open terms), we have the bisimulation as the following.

$$\begin{aligned} Q\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} &\xrightarrow{a(X)} Q_1 \equiv Q'\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \\ &\text{and for every closed } R' \\ Q'\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\}\{R'/X\} &\sim_{\text{ho}} P'\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\}\{R'/X\} \end{aligned}$$

By Lemma 2 and noticing that \tilde{Y} are guarded, we have

$$Q\{\tilde{T}/\tilde{X}\} \xrightarrow{a(X)} Q'\{\tilde{T}/\tilde{X}\}$$

and moreover $P'\{\tilde{T}/\tilde{X}\} \sim_{\text{ho}} Q'\{\tilde{T}/\tilde{X}\}$. Using Lemma 2 once again since \tilde{T} merely hold fresh names \tilde{m} , we have $Q \xrightarrow{a(X)} Q'$. Also this yields $P' \mathcal{R}_1 Q'$.

6. Suppose $P \equiv X | O$. So $P\{\tilde{T}/\tilde{X}\} \equiv \text{Tr}_{m_i} | O\{\tilde{T}/\tilde{X}\}$ in which m_i is from \tilde{T} . For convenience we still use \tilde{X} (which will not raise confusion). We will use the assumption that each m_i is fresh.

Then $P\{\tilde{T}/\tilde{X}\} \xrightarrow{\tilde{m}_i} O\{\tilde{T}/\tilde{X}\}$ and we argue that Q can match as requested by the strong HO-IO bisimilarity. By Lemma 2, we know that

$$P\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \xrightarrow{\tilde{m}_i} O\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\}$$

for every closed \tilde{R} .

Since $P\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \sim_{\text{ho}} Q\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\}$ (this is by the definition of \sim_{ho} over open terms), we have the bisimulation as

$$\begin{aligned} Q\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} &\xrightarrow{\tilde{m}_i(B_1)} O_1 \equiv O'\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \sim_{\text{ho}} O\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \\ B_1 &\equiv B\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \sim_{\text{ho}} 0 \end{aligned}$$

By Lemma 2 and because \tilde{Y} are guarded, we have

$$Q\{\tilde{T}/\tilde{X}\} \xrightarrow{\tilde{m}_i(B)} O'\{\tilde{T}/\tilde{X}\}$$

and moreover, $O\{\tilde{T}/\tilde{X}\} \sim_{\text{ho}} O'\{\tilde{T}/\tilde{X}\}$ and $B\{\tilde{T}/\tilde{X}\} \sim_{\text{ho}} 0$. Since m_i is fresh, we infer that Q must have a free variable X in the parallel firable position, and $B \equiv 0$ (otherwise the above equivalence with would fail, resulting in a contradiction). Thus we have the following.

$$Q \equiv X | O'$$

Furthermore, we have $O \mathcal{R}_1 O'$ as required.

7. Suppose $P \equiv X \langle A \rangle | O$. Again we use the assumption that m_i is fresh. So we have $P\{\tilde{T}/\tilde{X}\} \equiv (\text{Tr}_{m_i}^D) \langle A\{\tilde{T}/\tilde{X}\} \rangle | O\{\tilde{T}/\tilde{X}\}$ in which m_i is from \tilde{T} . Then $P\{\tilde{T}/\tilde{X}\} \xrightarrow{\tilde{m}_i(A\{\tilde{T}/\tilde{X}\})} O\{\tilde{T}/\tilde{X}\}$ and we argue that Q can match as requested by the strong HO-IO bisimilarity. By Lemma 2, we know that

$$P\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \xrightarrow{\tilde{m}_i(A\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\})} O\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\}$$

for every closed \tilde{R} .

Since $P\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \sim_{\text{ho}} Q\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\}$ (this is by the definition of \sim_{ho} over open terms), we have the bisimulation as

$$\begin{aligned} Q\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} &\xrightarrow{\overline{m}_i(B_1)} O_1 \equiv O'\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \sim_{\text{ho}} O\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \\ B_1 &\equiv B\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \sim_{\text{ho}} A\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \end{aligned}$$

By Lemma 2 and because \tilde{Y} are guarded, we have

$$Q\{\tilde{T}/\tilde{X}\} \xrightarrow{\overline{m}_i(B\{\tilde{T}/\tilde{X}\})} O'\{\tilde{T}/\tilde{X}\}$$

and moreover, $O\{\tilde{T}/\tilde{X}\} \sim_{\text{ho}} O'\{\tilde{T}/\tilde{X}\}$ and $A\{\tilde{T}/\tilde{X}\} \sim_{\text{ho}} B\{\tilde{T}/\tilde{X}\}$. Since m_i is fresh, we infer that Q must have a free variable X in the parallel firable position, and also the following.

$$Q \equiv X\langle B \rangle | O'$$

Furthermore, we have $O\mathcal{R}_1 O'$ as well as $A\mathcal{R}_1 B$.

8. Suppose $P \equiv X\langle d \rangle | O$. Once again the assumption that m_i is fresh will be used. So $P\{\tilde{T}/\tilde{X}\} \equiv (\text{Tr}_{m_i}^{D,d})\langle d \rangle | O\{\tilde{T}/\tilde{X}\}$ in which m_i is from \tilde{T} .

Then $P\{\tilde{T}/\tilde{X}\} \xrightarrow{\overline{m}_i(\langle Z \rangle(Z\langle d \rangle))} O\{\tilde{T}/\tilde{X}\}$ and we argue that Q can match as requested by the strong HO-IO bisimilarity. By Lemma 2, we know that

$$P\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \xrightarrow{\overline{m}_i(\langle Z \rangle(Z\langle d \rangle))} O\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\}$$

for every closed \tilde{R} .

Since $P\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \sim_{\text{ho}} Q\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\}$ (this is by the definition of \sim_{ho} over open terms), we have the bisimulation as

$$\begin{aligned} Q\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} &\xrightarrow{\overline{m}_i(B_1)} O_1 \equiv O'\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \sim_{\text{ho}} O\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \\ B_1 &\equiv B\{\tilde{T}/\tilde{X}\}\{\tilde{R}/\tilde{Y}\} \sim_{\text{ho}} \langle Z \rangle(Z\langle d \rangle) \end{aligned}$$

By Lemma 2 and because \tilde{Y} are guarded, we have

$$Q\{\tilde{T}/\tilde{X}\} \xrightarrow{\overline{m}_i(B\{\tilde{T}/\tilde{X}\})} O'\{\tilde{T}/\tilde{X}\}$$

and moreover, $O\{\tilde{T}/\tilde{X}\} \sim_{\text{ho}} O'\{\tilde{T}/\tilde{X}\}$ and $\langle Z \rangle(Z\langle d \rangle) \sim_{\text{ho}} B\{\tilde{T}/\tilde{X}\}$. Since m_i is fresh, we infer that Q must have a free variable X in the parallel firable position, and B must be of the form $\langle Z \rangle(Z\langle d \rangle)$ (otherwise the above equivalence with would fail, causing a contradiction). Thus we have the following.

$$Q \equiv X\langle d \rangle | O'$$

Furthermore, we have $O\mathcal{R}_1 O'$ as desired.

\sim_{hoio}° is a strong HO bisimulation. Suppose $P \sim_{\text{hoio}}^\circ Q$ to prove $P \sim_{\text{ho}} Q$. We show that the following relation \mathcal{R}_2 is a strong HO bisimulation.

$$\mathcal{R}_2 \stackrel{\text{def}}{=} \{(P, Q) \mid P \sim_{\text{hoio}}^\circ Q\} \cup \sim_{\text{hoio}}^\circ$$

We notice that \sim_{hoio}° is closed under substitution (Lemma 15 and Lemma 14), and moreover is τ -preserving (Lemma 17). Also notice that a strong HO bisimulation and a strong HO-IO bisimulation share the same requirement of output simulation. So the analysis becomes somewhat amenable in isolation.

Suppose $P \mathcal{R}_2 Q$ and $P \xrightarrow{\lambda} P'$. We note that we can focus on the case that P and Q are closed. If they are not (i.e., open), say $\text{fpv}(P, Q) = \{X_1, \dots, X_n\}$, we can use Lemma 15 to derive $P\{\tilde{R}/\tilde{X}\} \sim_{\text{hoio}}^\circ Q\{\tilde{R}/\tilde{X}\}$ for all closed \tilde{R} . Then we can use the same method to discuss over $P\{\tilde{R}/\tilde{X}\}$ and $Q\{\tilde{R}/\tilde{X}\}$ to show that they are HO bisimilar, and in turn P and Q are HO bisimilar. Therefore, the case analysis on $P \xrightarrow{\lambda} P'$ below takes P and Q as closed.

1. Cases that P and Q are some kind of abstractions. These are straightforward by simply noticing the salient property that \sim_{hoio}° is closed under substitution (Lemma 15).
2. Input: $P \xrightarrow{a(X)} P'$. Since $P \sim_{\text{hoio}}^\circ Q$, we know

$$Q \xrightarrow{a(X)} Q' \sim_{\text{hoio}}^\circ P'$$

Now because \sim_{hoio}° is closed under substitution (Lemma 15), we have

$$P'\{R/X\} \sim_{\text{hoio}}^\circ Q'\{R/X\}$$

for every closed R . Thus we derive that $P'\{R/X\} \mathcal{R}_2 Q'\{R/X\}$.

3. Output : $P \xrightarrow{\bar{a}(A)} P'$. Since $P \sim_{\text{hoio}}^\circ Q$, we know

$$Q \xrightarrow{\bar{a}(B)} Q' \sim_{\text{hoio}}^\circ P' \quad \text{and} \quad A \sim_{\text{hoio}}^\circ B$$

We thus have $P' \mathcal{R}_2 Q'$ and $A \mathcal{R}_2 B$.

4. τ : $P \xrightarrow{\tau} P'$. Since $P \sim_{\text{hoio}}^\circ Q$ and \sim_{hoio}° is τ -preserving (Lemma 17), we know

$$Q \xrightarrow{\tau} Q' \sim_{\text{hoio}}^\circ P'$$

We thus have $P' \mathcal{R}_2 Q'$.

□

Proof of Lemma 20. We shall focus on the case (2), since cases (3) and (1) are similar and case (1) can also be referred to [15].

In the first part of this proof, we show that the relation \mathcal{R} defined below is an open strong normal bisimulation, and thus establish $P \sim_{\text{nr}}^\circ Q$. Notice that here $\prod_i P_i$ is a shortcut for $\prod_{i=1}^k P_i$ for some $k \in \mathbb{N}$.

Also note that in \mathcal{R} , every m with a subscript is fresh with respect to P , Q , and those P and Q with subscripts as well.

$$\mathcal{R} \stackrel{\text{def}}{=} \left\{ (P, Q) \left| \begin{array}{c} \prod_i m_i.P_i \mid \prod_j m_j(Z_j).P_j\langle Z_j \rangle \mid \prod_k m_k(Z_k).Z_k\langle P_k \rangle \mid P \\ \stackrel{\lambda}{=} \\ \prod_i m_i.Q_i \mid \prod_j m_j(Z_j).Q_j\langle Z_j \rangle \mid \prod_k m_k(Z_k).Z_k\langle Q_k \rangle \mid Q \end{array} \right. \right\}$$

To reach the bisimulation, suppose $P \mathcal{R} Q$ and we make a case analysis.

1. Obviously these terms cannot be abstractions.
2. Suppose $P \xrightarrow{\bar{a}(A)} P'$.

There are three possibilities depending on the type of A . However, they are similar, and we focus on one of them.

- (a) A is a process abstraction. In this case, we know that

$$\xrightarrow{\bar{a}(A)} \begin{array}{c} \prod_i m_i.P_i \mid \prod_j m_j(Z_j).P_j\langle Z_j \rangle \mid \prod_k m_k(Z_k).Z_k\langle P_k \rangle \mid P \\ \prod_i m_i.P_i \mid \prod_j m_j(Z_j).P_j\langle Z_j \rangle \mid \prod_k m_k(Z_k).Z_k\langle P_k \rangle \mid P' \end{array}$$

Since $P \mathcal{R} Q$ and all the m with subscripts are fresh (actually freshness is not essential because these fresh names can only release input actions), in terms of the definition of \mathcal{R} , we have

$$\xrightarrow{\bar{a}(B)} \begin{array}{c} \prod_i m_i.Q_i \mid \prod_j m_j(Z_j).Q_j\langle Z_j \rangle \mid \prod_k m_k(Z_k).Z_k\langle Q_k \rangle \mid Q \\ \prod_i m_i.Q_i \mid \prod_j m_j(Z_j).Q_j\langle Z_j \rangle \mid \prod_k m_k(Z_k).Z_k\langle Q_k \rangle \mid Q' \end{array}$$

and

$$\stackrel{\circ}{\sim}_{\text{nr}} \begin{array}{c} m'(Z').A\langle Z' \rangle \mid \prod_i m_i.P_i \mid \prod_j m_j(Z_j).P_j\langle Z_j \rangle \mid \prod_k m_k(Z_k).Z_k\langle P_k \rangle \mid P' \\ m'(Z').B\langle Z' \rangle \mid \prod_i m_i.Q_i \mid \prod_j m_j(Z_j).Q_j\langle Z_j \rangle \mid \prod_k m_k(Z_k).Z_k\langle Q_k \rangle \mid Q' \end{array} \quad (1)$$

Thus one can observe that

$$Q \xrightarrow{\bar{a}(B)} Q'.$$

and

$$m'(Z').A\langle Z' \rangle \mid P' \mathcal{R} m'(Z').B\langle Z' \rangle \mid Q'$$

because from (1) we have the following.

$$\stackrel{\circ}{\sim}_{\text{nr}} \begin{array}{c} \prod_i m_i.P_i \mid \prod_j m_j(Z_j).P_j\langle Z_j \rangle \mid \prod_k m_k(Z_k).Z_k\langle P_k \rangle \mid m'(Z').A\langle Z' \rangle \mid P' \\ \prod_i m_i.Q_i \mid \prod_j m_j(Z_j).Q_j\langle Z_j \rangle \mid \prod_k m_k(Z_k).Z_k\langle Q_k \rangle \mid m'(Z').B\langle Z' \rangle \mid Q' \end{array}$$

Moreover, Q can actually simulate P in a more tight way. Specifically, from (1) one can readily see that $P' \mathcal{R} Q'$, by incorporating the components $m'(Z').A\langle Z' \rangle$ and $m'(Z').B\langle Z' \rangle$ in the indexed collections respectively.

- (b) A is a name abstraction. Similar.
- (c) A is not an abstraction. Similar.

We stress that, as one can see, the continuations of output by P and Q , i.e., P' and Q' respectively, are also directly related by \mathcal{R} , so they are also open strong normal bisimilar.

3. Suppose $P \xrightarrow{\tau} P'$. We know that

$$\xrightarrow{\tau} \frac{\prod_i m_i.P_i \mid \prod_j m_j(Z_j).P_j\langle Z_j \rangle \mid \prod_k m_k(Z_k).Z_k\langle P_k \rangle \mid P}{\prod_i m_i.P_i \mid \prod_j m_j(Z_j).P_j\langle Z_j \rangle \mid \prod_k m_k(Z_k).Z_k\langle P_k \rangle \mid P'}$$

Since $P \mathcal{R} Q$ and all the m with subscripts are fresh (this is not essential because those prefixes can only fire input actions), we have

$$\xrightarrow{\tau} \frac{\prod_i m_i.Q_i \mid \prod_j m_j(Z_j).Q_j\langle Z_j \rangle \mid \prod_k m_k(Z_k).Z_k\langle Q_k \rangle \mid Q}{\prod_i m_i.Q_i \mid \prod_j m_j(Z_j).Q_j\langle Z_j \rangle \mid \prod_k m_k(Z_k).Z_k\langle Q_k \rangle \mid Q'}$$

and

$$\sim_{\text{nr}}^{\circ} \frac{\prod_i m_i.P_i \mid \prod_j m_j(Z_j).P_j\langle Z_j \rangle \mid \prod_k m_k(Z_k).Z_k\langle P_k \rangle \mid P'}{\prod_i m_i.Q_i \mid \prod_j m_j(Z_j).Q_j\langle Z_j \rangle \mid \prod_k m_k(Z_k).Z_k\langle Q_k \rangle \mid Q'} \quad (2)$$

Thus one can see from the matching above that

$$Q \xrightarrow{\tau} Q'$$

and

$$P' \mathcal{R} Q'$$

thanks to (2).

4. Suppose $P \xrightarrow{a(X)} P'$. We know that

$$\xrightarrow{a(X)} \frac{\prod_i m_i.P_i \mid \prod_j m_j(Z_j).P_j\langle Z_j \rangle \mid \prod_k m_k(Z_k).Z_k\langle P_k \rangle \mid P}{\prod_i m_i.P_i \mid \prod_j m_j(Z_j).P_j\langle Z_j \rangle \mid \prod_k m_k(Z_k).Z_k\langle P_k \rangle \mid P'}$$

Since $P \mathcal{R} Q$ and all the m with subscripts are fresh, by the definition of \mathcal{R} , we have

$$\xrightarrow{a(X)} \frac{\prod_i m_i.Q_i \mid \prod_j m_j(Z_j).Q_j\langle Z_j \rangle \mid \prod_k m_k(Z_k).Z_k\langle Q_k \rangle \mid Q}{\prod_i m_i.Q_i \mid \prod_j m_j(Z_j).Q_j\langle Z_j \rangle \mid \prod_k m_k(Z_k).Z_k\langle Q_k \rangle \mid Q'}$$

and

$$\sim_{\text{nr}}^{\circ} \frac{\prod_i m_i.P_i \mid \prod_j m_j(Z_j).P_j\langle Z_j \rangle \mid \prod_k m_k(Z_k).Z_k\langle P_k \rangle \mid P'}{\prod_i m_i.Q_i \mid \prod_j m_j(Z_j).Q_j\langle Z_j \rangle \mid \prod_k m_k(Z_k).Z_k\langle Q_k \rangle \mid Q'} \quad (3)$$

Thus one can see from the matching above that

$$Q \xrightarrow{a(X)} Q'$$

and

$$P' \mathcal{R} Q'$$

thanks to (3).

5. Suppose $P \equiv X | P'$. We have

$$\begin{aligned} & \prod_i m_i.P_i | \prod_j m_j(Z_j).P_j\langle Z_j \rangle | \prod_k m_k(Z_k).Z_k\langle P_k \rangle | P \\ \equiv & \prod_i m_i.P_i | \prod_j m_j(Z_j).P_j\langle Z_j \rangle | \prod_k m_k(Z_k).Z_k\langle P_k \rangle | X | P' \end{aligned}$$

Since $P \mathcal{R} Q$ and all the m with subscripts are fresh, in terms of the definition of \mathcal{R} , we have

$$\begin{aligned} & \prod_i m_i.Q_i | \prod_j m_j(Z_j).Q_j\langle Z_j \rangle | \prod_k m_k(Z_k).Z_k\langle Q_k \rangle | Q \\ \equiv & \prod_i m_i.Q_i | \prod_j m_j(Z_j).Q_j\langle Z_j \rangle | \prod_k m_k(Z_k).Z_k\langle Q_k \rangle | X | Q' \end{aligned}$$

and

$$\begin{aligned} & \prod_i m_i.P_i | \prod_j m_j(Z_j).P_j\langle Z_j \rangle | \prod_k m_k(Z_k).Z_k\langle P_k \rangle | P' \\ \sim_{\text{nr}}^{\circ} & \prod_i m_i.Q_i | \prod_j m_j(Z_j).Q_j\langle Z_j \rangle | \prod_k m_k(Z_k).Z_k\langle Q_k \rangle | Q' \end{aligned} \quad (4)$$

From this, one can see that

$$Q \equiv X | Q'$$

and

$$P' \mathcal{R} Q'$$

due to (4).

6. Suppose $P \equiv X\langle A \rangle | P'$. We tackle the case that A is a process-abstraction, and the other cases are similar. In this case, we know that

$$\begin{aligned} & \prod_i m_i.P_i | \prod_j m_j(Z_j).P_j\langle Z_j \rangle | \prod_k m_k(Z_k).Z_k\langle P_k \rangle | P \\ \equiv & \prod_i m_i.P_i | \prod_j m_j(Z_j).P_j\langle Z_j \rangle | \prod_k m_k(Z_k).Z_k\langle P_k \rangle | X\langle A \rangle | P' \end{aligned}$$

Since $P \mathcal{R} Q$ and all the m with subscripts are fresh, in terms of the definition of \mathcal{R} , we have

$$\begin{aligned} & \prod_i m_i.Q_i | \prod_j m_j(Z_j).Q_j\langle Z_j \rangle | \prod_k m_k(Z_k).Z_k\langle Q_k \rangle | Q \\ \equiv & \prod_i m_i.Q_i | \prod_j m_j(Z_j).Q_j\langle Z_j \rangle | \prod_k m_k(Z_k).Z_k\langle Q_k \rangle | X\langle B \rangle | Q' \end{aligned}$$

and

$$\begin{aligned} & m'(Z').A\langle Z' \rangle | \prod_i m_i.P_i | \prod_j m_j(Z_j).P_j\langle Z_j \rangle | \prod_k m_k(Z_k).Z_k\langle P_k \rangle | P' \\ \sim_{\text{nr}}^{\circ} & m'(Z').B\langle Z' \rangle | \prod_i m_i.Q_i | \prod_j m_j(Z_j).Q_j\langle Z_j \rangle | \prod_k m_k(Z_k).Z_k\langle Q_k \rangle | Q' \end{aligned} \quad (5)$$

From this matching, one can observe that

$$Q \equiv X\langle B \rangle | Q'$$

and

$$m'(Z').A\langle Z' \rangle | P' \mathcal{R} m'(Z').B\langle Z' \rangle | Q'$$

because from (5) we have the following.

$$\begin{aligned} & \prod_i m_i.P_i | \prod_j m_j(Z_j).P_j\langle Z_j \rangle | \prod_k m_k(Z_k).Z_k\langle P_k \rangle | m'(Z').A\langle Z' \rangle | P' \\ \sim_{\text{nr}}^{\circ} & \prod_i m_i.Q_i | \prod_j m_j(Z_j).Q_j\langle Z_j \rangle | \prod_k m_k(Z_k).Z_k\langle Q_k \rangle | m'(Z').B\langle Z' \rangle | Q' \end{aligned}$$

Furthermore, Q can simulate P in a more directly way. By (1) one can easily see that $P' \mathcal{R} Q'$, by means of combining the components $m'(Z').A\langle Z' \rangle$ and $m'(Z').B\langle Z' \rangle$ in the indexed collections respectively.

corresponding trigger for the variable types. Accordingly, the tuples of triggers are represented by $\widetilde{\text{Tr}}_{m_1}$, $\widetilde{\text{Tr}}_{m_2}^D$ and $\widetilde{\text{Tr}}_{m_3}^{D,d}$, respectively. We note that the names of all the triggers are fresh.

Assuming $\{\widetilde{X}_1, \widetilde{X}_2, \widetilde{X}_3\} = \text{fpv}(P, Q)$, it is not hard to have the following equivalence from the definition of the strong normal bisimilarity on open processes; see [15, 19] for a reference. We notice that the first equality is based on open processes whereas the second is on closed ones.

$$\begin{aligned}
& P \sim_{\text{nr}} Q \\
& \text{if and only if} \\
& P\{\widetilde{\text{Tr}}_{m_1}/\widetilde{X}_1\}\{\widetilde{\text{Tr}}_{m_2}^D/\widetilde{X}_2\}\{\widetilde{\text{Tr}}_{m_3}^{D,d}/\widetilde{X}_3\} \sim_{\text{nr}} Q\{\widetilde{\text{Tr}}_{m_1}/\widetilde{X}_1\}\{\widetilde{\text{Tr}}_{m_2}^D/\widetilde{X}_2\}\{\widetilde{\text{Tr}}_{m_3}^{D,d}/\widetilde{X}_3\}
\end{aligned}$$

Now we proceed to showing the implication claimed by the lemma, by establishing the bisimulation between P and Q using the equivalence above. For the sake of convenience, we abbreviate $\{\widetilde{\text{Tr}}_{m_1}/\widetilde{X}_1\}\{\widetilde{\text{Tr}}_{m_2}^D/\widetilde{X}_2\}\{\widetilde{\text{Tr}}_{m_3}^{D,d}/\widetilde{X}_3\}$ as σ .

1. The matching of Q when P is not an abstraction or is a name-abstraction is trivial. If P is a process-abstraction of the form $\langle Y \rangle A$ in which Y is of the type of process-abstraction (cases when Y is of the type of name-abstraction or non-abstraction are similar), then since $P \sim_{\text{nr}} Q$, Q must take the form $\langle Y \rangle B$, and moreover

$$A\sigma\{\text{Tr}_{m'}^D/Y\} \sim_{\text{nr}} B\sigma\{\text{Tr}_{m'}^D/Y\}$$

Hence we have $A \sim_{\text{nr}} B$, as required.

2. Suppose $P \xrightarrow{a(X)} P'$. We assume X has the type of process abstraction, and the other cases are similar. By Lemma 2, $P\sigma \xrightarrow{a(X)} P'\sigma$. Since $P\sigma \sim_{\text{nr}} Q\sigma$, we know that $Q\sigma \xrightarrow{a(X)} Q_1$ and

$$P'\sigma\{\text{Tr}_{m'}^D/X\} \sim_{\text{nr}} Q_1\{\text{Tr}_{m'}^D/X\}$$

for fresh m' . Using Lemma 2 again, we have $Q \xrightarrow{a(X)} Q'$ where $Q_1 \equiv Q'\sigma$, and moreover the following.

$$P'\sigma\{\text{Tr}_{m'}^D/X\} \sim_{\text{nr}} Q'\sigma\{\text{Tr}_{m'}^D/X\}$$

We thus have $P' \sim_{\text{nr}} Q'$, as required.

3. Suppose $P \xrightarrow{\bar{a}(A)} P'$. We focus on the case A is a process-abstraction, and the others are similar. By Lemma 2, $P\sigma \xrightarrow{\bar{a}(A\sigma)} P'\sigma$. Since $P\sigma \sim_{\text{nr}} Q\sigma$, we know that $Q\sigma \xrightarrow{\bar{a}(B_1)} Q_1$ and

$$m'(Z).(A\sigma)\langle Z \rangle | P'\sigma \sim_{\text{nr}} m'(Z).(B_1)\langle Z \rangle | Q_1$$

for fresh m' . Using Lemma 2 again, we have $Q \xrightarrow{\bar{a}(B)} Q'$ where $B_1 \equiv B\sigma$ and $Q_1 \equiv Q'\sigma$, and moreover the following.

$$m'(Z).(A\sigma)\langle Z \rangle | P'\sigma \sim_{\text{nr}} m'(Z).(B\sigma)\langle Z \rangle | Q'\sigma$$

We thus have

$$m'(Z).(A)\langle Z \rangle | P' \sim_{\text{nr}} m'(Z).(B)\langle Z \rangle | Q'$$

as required.

4. Suppose $P \xrightarrow{\tau} P'$. By Lemma 2, $P\sigma \xrightarrow{\tau} P'\sigma$. Since $P\sigma \sim_{\text{nr}} Q\sigma$, we know that $Q\sigma \xrightarrow{\tau} Q_1$ and

$$P'\sigma \sim_{\text{nr}} Q_1.$$

Using Lemma 2 again, we have $Q \xrightarrow{\tau} Q'$ where $Q_1 \equiv Q'\sigma$, and moreover the following.

$$P'\sigma \sim_{\text{nr}} Q'\sigma$$

We thus have $P' \sim_{\text{nr}} Q'$, as required.

5. If $P \equiv X|P'$, we need to show that $Q \equiv X|Q'$ and $P' \sim_{\text{nr}} Q'$. Here $X \in \tilde{X}_1$ and we suppose the trigger replacing X has name m_1^X . To prove this, we observe that

$$P\sigma \xrightarrow{\overline{m_1^X}} 0|P'\sigma$$

Thanks to $P\sigma \sim_{\text{nr}} Q\sigma$, $Q\sigma$ is supposed to match this action. Now because m_1^X is fresh, Q must contain a free X so as to be able to make an output on m_1^X . So we know that Q must take the form $X|Q'$. Moreover, $Q\sigma$ matches $P\sigma$ by

$$Q\sigma \xrightarrow{\overline{m_1^X}} 0|Q'\sigma$$

with

$$m'.0|P'\sigma \sim_{\text{nr}} m'.0|Q'\sigma$$

where m' is fresh. Using Corollary 21, we derive $P'\sigma \sim_{\text{nr}} Q'\sigma$ and thus $P' \sim_{\text{nr}} Q'$, as needed. Actually this is straightforward because m' is fresh, so one can consume it simultaneously from $m'.0|P'\sigma$ and $m'.0|Q'\sigma$.

6. If $P \equiv X\langle A \rangle|P'$ in which A is a process-abstraction (the other cases are similar), we need to show that $Q \equiv X\langle B \rangle|Q'$ and $m'(Z).A\langle Z \rangle|P' \sim_{\text{nr}} m'(Z).B\langle Z \rangle|Q'$ for fresh m' . Here $X \in \tilde{X}_2$ and we assume the trigger replacing X has name m_2^X . To do this, we observe that

$$P\sigma \xrightarrow{\overline{m_2^X}(A\sigma)} 0|P'\sigma$$

Since $P\sigma \sim_{\text{nr}} Q\sigma$, $Q\sigma$ has to match this action. Now because m_2^X is fresh, Q must have a free X so as to be capable of an output over m_2^X . Therefore, we claim that Q must take the form $X\langle B \rangle|Q'$. Moreover, $Q\sigma$ matches $P\sigma$ by

$$Q\sigma \xrightarrow{\overline{m_2^X}(B\sigma)} 0|Q'\sigma$$

with

$$m'(Z).(A\sigma)\langle Z \rangle|P'\sigma \sim_{\text{nr}} m'(Z).(B\sigma)\langle Z \rangle|Q'\sigma$$

Hence we have

$$m'(Z).A\langle Z \rangle|P' \sim_{\text{nr}} m'(Z).B\langle Z \rangle|Q'$$

as desired.

7. If $P \equiv X\langle d \rangle | P'$, we need to show that $Q \equiv X\langle d \rangle | Q'$ and $P' \sim_{\text{nr}} Q'$. Here $X \in \tilde{X}_3$ and we assume the trigger replacing X has name m_3^X . To do this, we observe that

$$P\sigma \xrightarrow{\overline{m_3^X}(\langle Z \rangle(Z\langle d \rangle))} 0 | P'\sigma$$

Due to $P\sigma \sim_{\text{nr}} Q\sigma$, $Q\sigma$ has to match this action. Now because m_3^X is fresh, Q must have a free X so as to be able to fire an output on m_3^X . Therefore, we know that Q must take the form $X\langle e \rangle | Q'$. Moreover, $Q\sigma$ matches $P\sigma$ by

$$Q\sigma \xrightarrow{\overline{m_3^X}(\langle Z \rangle(Z\langle e \rangle))} 0 | Q'\sigma$$

with

$$m'(Z_1).Z_1\langle\langle Z \rangle(Z\langle d \rangle)\rangle | P'\sigma \sim_{\text{nr}} m'(Z_1).Z_1\langle\langle Z \rangle(Z\langle e \rangle)\rangle | Q'\sigma$$

where m' is fresh.

Using Corollary 21, we derive $\langle Z \rangle(Z\langle d \rangle) \sim_{\text{nr}} \langle Z \rangle(Z\langle e \rangle)$ and $P'\sigma \sim_{\text{nr}} Q'\sigma$.

Actually this is also a straightforward consequence of the congruence properties of \sim_{nr} . Basically, one use the freshness of m' to establish the equivalence between $P'\sigma \sim_{\text{nr}} Q'\sigma$ and then by consuming them in the manner of strong bisimulation, one obtain the equivalence between $\langle Z \rangle(Z\langle d \rangle)$ and $\langle Z \rangle(Z\langle e \rangle)$.

From this, we know that d is the same as e (otherwise it would break the bisimulation equivalence, since the former is potentially capable of doing actions on d but the other cannot), and in the meanwhile $P' \sim_{\text{nr}} Q'$, as desired.

Now the proof is completed. \square

Proof of Lemma 23(2). We show that \sim_{nr}° is a strong HO-IO bisimulation. Suppose $P \sim_{\text{nr}}^\circ Q$. Examining the definitions of the two bisimulations, one can see that the difference consists in two clauses: the output case and the case when P takes an open form with a free variable of the process-abstraction type.

1. If $P \xrightarrow{\bar{a}A} P'$ in which A is a process abstraction (the cases A is of other abstraction types are similar), then $Q \xrightarrow{\bar{a}B} Q'$ for process-abstraction B that is $\langle Y \rangle B_1$, and it holds for fresh m that $m(Z).A\langle Z \rangle | P' \sim_{\text{nr}}^\circ m(Z).B\langle Z \rangle | Q'$. Now using Lemma 20, we have $A \sim_{\text{nr}}^\circ B$ and $P' \sim_{\text{nr}}^\circ Q'$, as required by the strong HO-IO bisimulation.
2. If $P \equiv X\langle A \rangle | P'$ in which A is a process abstraction (the cases A is of other abstraction types are similar), then $Q \equiv X\langle B \rangle | Q'$, and moreover B is also a process abstraction and $m(Z).A\langle Z \rangle | P' \sim_{\text{nr}}^\circ m(Z).B\langle Z \rangle | Q'$ for fresh m . By Lemma 20, we have $A \sim_{\text{nr}}^\circ B$ and $P' \sim_{\text{nr}}^\circ Q'$, as required by the strong HO-IO bisimulation.

\square

B Proofs for Section 4

In this appendix, we provide the detailed proofs for Section 4.

Proof of Proposition 26. The proof idea is similar to the one in [15, 17], but with strict extension tackling abstractions. For the sake of simplicity, we take \sim to be \sim_{hoio}° . If R is 0, $P \sim Q$ follows immediately. So we can assume that R is not 0, and thus neither P nor Q is an abstractions. We prove simultaneously the following two claims by induction on $\text{depth}(P) + \text{depth}(Q) + \text{depth}(R)$:

1. If $P|R \sim Q|R$ then $P \sim Q$.
2. For an input action or output action λ , if $R \xrightarrow{\lambda} R'$ and $P|R \sim Q|R'$, then $Q \xrightarrow{\lambda} Q'$ for some Q' s.t. $P \sim Q'$.

Proof of 1.

- Both P and Q are non-abstractions. Otherwise terms $P|R$ and $Q|R$ are invalid.
- For an input action or output action λ , suppose that $P \xrightarrow{\lambda} P'$, we then have $P|R \xrightarrow{\lambda} P'|R$, which can be matched by either: (a) $Q|R \xrightarrow{\lambda} Q'|R$, or (b) $Q|R \xrightarrow{\lambda} Q|R'$. Every transition decreases the depth of the process strictly, so $\text{depth}(P') < \text{depth}(P)$. For case (a), by induction hypothesis for 1, we have $P' \sim Q'$. For case (b), by induction hypothesis for 2, we have $P' \sim Q'$.
- If $P \equiv X|S$, then $X|S|R \sim Q|R$. We distinguish between the source of the variable X from $Q|R$. (a) If the variable X comes from R , which can be rewritten as $X|S'$ for some S' , then $S|X|S' \sim Q|S'$. Since $\text{depth}(S') < \text{depth}(R)$, by induction hypothesis for 1, we have $S|X \sim Q$, thus $P \sim Q$. (b) Otherwise, $Q \equiv X|S'$ for some S' , then $S|R \sim S'|R$. Since $\text{depth}(S') + \text{depth}(S) < \text{depth}(Q) + \text{depth}(P)$, by induction hypothesis for 1, we have $S \sim S'$, and then $P \sim Q$ follows by the congruence property.
- If $P \equiv X\langle A \rangle|S$, then $X\langle A \rangle|S|R \sim Q|R$. By Definition 4, there is a component $X\langle B \rangle$ in $Q|R$. (a) If $R \equiv X\langle B \rangle|S'$, we thus have $A \sim B$ and $S|X\langle B \rangle|S' \sim Q|S'$. Since $\text{depth}(S) + \text{depth}(X\langle B \rangle) + \text{depth}(S') < \text{depth}(P) + \text{depth}(R)$, by induction hypothesis for 1, we have $S|X\langle B \rangle \sim Q$, and then $P \sim Q$ follows by the congruence property. (b) Otherwise, $Q \equiv X\langle B \rangle|S'$, we thus have $A \sim B$ and $S|R \sim S'|R$. Since $\text{depth}(S') + \text{depth}(S) < \text{depth}(Q) + \text{depth}(P)$, by induction hypothesis for 1, we have $S \sim S'$, and then $P \sim Q$ follows by the congruence property.
- If $P \equiv X\langle d \rangle|S$, then $X\langle d \rangle|S|R \sim Q|R$. By Definition 4, there is a component $X\langle d \rangle$ in $Q|R$. (a) If $R \equiv X\langle d \rangle|S'$, we have $P|S' \sim Q|S'$. Since $\text{depth}(S') < \text{depth}(R)$, by induction hypothesis for 1, we have $P \sim Q$. (b) Otherwise, $Q \equiv X\langle d \rangle|S'$, we have $S|R \sim S'|R$. Since $\text{depth}(S') + \text{depth}(S) < \text{depth}(Q) + \text{depth}(P)$, by induction hypothesis for 1 we have $S \sim S'$, and then $P \sim Q$ follows by the congruence property.

The case when starting from Q is symmetric, thus we conclude that $P \sim Q$.

Proof of 2.

Assume that $R \xrightarrow{\lambda} R'$ and $P|R \sim Q|R'$. Then we have $P|R \xrightarrow{\lambda} P|R'$, and there exists some S s.t. $Q|R' \xrightarrow{\lambda} S$ with $P|R' \sim S$. There are two cases:

- The transition $Q|R' \xrightarrow{\lambda} S$ derives from the component Q , i.e., $Q \xrightarrow{\lambda} Q'$ and $P|R' \sim Q'|R'$. Since $\text{depth}(Q') < \text{depth}(Q)$, by induction hypothesis for 1, we have $P \sim Q'$.
- The transition $Q|R' \xrightarrow{\lambda} S$ derives from the component R' , i.e., $R' \xrightarrow{\lambda} R''$ and $P|R' \sim Q|R''$. Since $\text{depth}(R'') < \text{depth}(R')$, by induction hypothesis for 2, we have $Q \xrightarrow{\lambda} Q'$ for some Q' s.t. $P \sim Q'$.

□

Proof of Proposition 27. We proceed by induction on $\text{depth}(P)$. Again, we take \sim to be \sim_{hoio}° .

- If some $P_i \sim Q_j$ (w.l.o.g., assume that $P_1 \sim Q_1$), then we have the following two prime decompositions for P : $P \sim P_1 | \prod_{i=2}^k P_i$ and $P \sim Q_1 | \prod_{j=2}^l Q_j$. By Proposition 26, we have $\prod_{i=2}^k P_i \sim \prod_{j=2}^l Q_j$. By induction hypothesis, the two prime decompositions $\prod_{i=2}^k P_i$ and $\prod_{j=2}^l Q_j$ are identical up to \sim and permutation of indices. Thus $\prod_{i=1}^k P_i$ and $\prod_{j=1}^l Q_j$ are also identical.

- Assume that for every i, j we have $P_i \approx Q_j$.
 - If either $k = 1$ or $l = 1$, then $k = l = 1$ and $P_1 \equiv Q_1$ by the definition of prime process. This is a contradiction.
 - If $k, l \geq 2$, w.l.o.g., we can assume that $\text{depth}(P_1) \leq \text{depth}(P_i)$ for any $1 \leq i \leq k$ and $\text{depth}(P_1) \leq \text{depth}(Q_j)$ for any $1 \leq j \leq l$.
 - * If P_1 is X , as $P \sim \prod_{j=1}^l Q_j$, then one of Q_j must be X , a contradiction.
 - * If P_1 is $X\langle A \rangle$, as $P \sim \prod_{j=1}^l Q_j$, then one of Q_j must be $X\langle B \rangle$ with $B \sim A$, we thus have $X\langle A \rangle \sim X\langle B \rangle$, a contradiction.
 - * If P_1 is $X\langle d \rangle$, as $P \sim \prod_{j=1}^l Q_j$, then one of Q_j must be $X\langle d \rangle$, a contradiction.
 - * If $P_1 = m(X).R$. Since $\text{depth}(R) < \text{depth}(P)$, by induction hypothesis, R has a unique prime decomposition $R = \prod_{g=1}^h R_g$. We have $P = \prod_{i=2}^k P_i | m(x).(\prod_{g=1}^h R_g) \xrightarrow{m(X)} P'$ with unique prime decomposition $P' \sim \prod_{g=1}^h R_g | \prod_{i=2}^k P_i$. Since $P \sim \prod_{j=1}^l Q_j$, w.l.o.g., the corresponding transition is $\prod_{j=1}^l Q_j \xrightarrow{m(X)} T | \prod_{j=2}^l Q_j \sim P'$. By induction hypothesis, the prime decomposition of $T | \prod_{j=2}^l Q_j$ should be $\prod_{g=1}^h R_g | \prod_{i=2}^k P_i$. As Q_2 is prime, it must be equal with a process in $\bigcup_{g=1}^h R_g \cup \bigcup_{i=2}^k P_i$. Since $\text{depth}(R_g) < \text{depth}(Q_2)$ for any $1 \leq g \leq h$, by Lemma 10, $R_g \approx Q_2$ for any $1 \leq g \leq h$, thus $Q_2 \sim P_i$ for some $2 \leq i \leq k$, a contradiction with the assumption that $P_i \approx Q_j$ for every i, j .
 - * If P_1 is $\bar{m}(R)$. Similar to the last case.

□

Proof of Lemma 29. By Proposition 27, we have

$$\text{nf}(Q|Q') \equiv \prod_{1 \leq i \leq j} a_i(X_i).R_i | \prod_{1 \leq i \leq k} \bar{b}_i(S_i) | \prod_{1 \leq i \leq l} Y_i | \prod_{1 \leq i \leq g} Z_i\langle T_i \rangle | \prod_{1 \leq i \leq h} Z'_i\langle n_i \rangle$$

where the processes $a_i(X_i).R_i, \bar{b}_i(S_i), Y_i, Z_i\langle T_i \rangle, Z'_i\langle n_i \rangle$ are in normal form and prime. By Lemma 28, $a(x).P \sim \text{nf}(Q|Q')$. Then we have $k = 0, l = 0, g = 0$ and $h = 0$. For any $i \in [1, j]$ (meaning that i is an integer s.t. $1 \leq i \leq j$), $a_i = a$ and $X_i = X$. We can summarize that $a(X).P \sim \prod_{1 \leq i \leq j} a(X).R_i$, where $j \geq 2$ as there are at least two processes that are not 0. Let $i_1, i_2 \in [1, j]$ be two indices with $i_1 \neq i_2$. Transition $\prod_{1 \leq i \leq j} a(X).R_i \xrightarrow{a(X)} R_{i_1} | \prod_{1 \leq i \leq j, i \neq i_1} a(X).R_i$ should be bisimulated by the following transition: $a(x).P \xrightarrow{a(x)} P \sim R_{i_1} | \prod_{1 \leq i \leq j, i \neq i_1} a(X).R_i$. Similarly, we have

$$P \sim R_{i_2} | \prod_{1 \leq i \leq j, i \neq i_2} a(X).R_i \sim R_{i_1} | \prod_{1 \leq i \leq j, i \neq i_1} a(X).R_i$$

By Proposition 26, we know $R_{i_2} | a(X).R_{i_1} \sim R_{i_1} | a(X).R_{i_2}$. Since $a(X).R_{i_1}$ and $a(X).R_{i_2}$ are prime, $\text{depth}(a(X).R_{i_1}) > \text{depth}(R_{i_1})$, we have $a(X).R_{i_1} \approx R_{i_1}$. By Proposition 27, we infer $a(X).R_{i_1} \sim a(X).R_{i_2}$. As this holds for any $i_1 \neq i_2$, we conclude that $a(X).P \sim \prod_{1 \leq i \leq j} a(X).R_1$ with $j \geq 2$ and $a(X).R_1$ is in normal form. □