A Higher-Order Distributed Calculus with Name Creation

Adrien Piérard
Tohoku University
Email: adrien@kb.ecei.tohoku.ac.jp

Eijiro Sumii
Tohoku University
Email: sumii@kb.ecei.tohoku.ac.jp

***Abstract*—This paper introduces HOπPn, the higher-order π-calculus with *passivation* and *name creation*, and develops an equivalence theory for this calculus. Passivation [Schmitt and Stefani] is a language construct that elegantly models higher-order distributed behaviours like failure, migration, or duplication (e.g. when a running process or virtual machine is copied), and name creation consists in generating a fresh name instead of hiding one. Combined with higher-order distribution, name creation leads to different semantics from name hiding, and is closer to implementations of distributed systems. We define for this new calculus a theory of sound and complete environmental bisimulation to prove reduction-closed barbed equivalence and (a reasonable form of) congruence. We furthermore define environmental *simulations* to prove behavioural *approximation*, and use these theories to show non-trivial examples of equivalence or approximation. Those examples could not be proven with previous theories, which were either unsound or incomplete under the presence of process duplication and name restriction, or else required universal quantification over general contexts.**

I. Introduction

Background: With the increasing call for fault tolerance, on-demand computational power and better responsiveness, higher-order and distribution are pervasive in today's computing environment. In this paper, we call *higher-order* the ability to send and receive processes through communication channels, and *distribution* the possibility of location-dependant behaviour. For example, Dell and Hewlett Packard sell products with virtual machine live migration [17], [5], and Gmail relies on remote execution of JavaScript in the users' browsers. Yet, despite the ubiquity and importance of such higher-order distributed systems, the inherent complexity of these systems makes them difficult to analyse, and thus subject to bugs. Therefore, formal models and methods that help reason about higher-order distribution are sought after.

Passivation [18], [6], [8], [11] is a language abstraction for elegantly modelling higher-order distributed systems in process calculi based on the higher-order π-calculus [12], [15] (with which we assume our reader's familiarity). In its simplest form, passivation consists of a syntax of *located processes* $l[P]$, where l is a name called a *location* and P is a process *located at* l, and two labelled transition rules, $l[P] \xrightarrow{\alpha} l[P']$ if $P \xrightarrow{\alpha} P'$ (TRANSP), and $l[P] \xrightarrow{\overline{l}\langle P\rangle} 0$ (PASSIV), where the relation $P \xrightarrow{\alpha} Q$ in general reads "P does action α and becomes Q." The TRANSP rule states that locations are *transparent*, i.e. do not hide any transition α of the processes they are hosting. The PASSIV rule shows how a located process can be *passivated*, i.e. stopped and output to a channel of the same name as the location.

Despite its simplicity, passivation is yet powerful enough to model complex higher-order distributed behaviours. For example, one can conveniently model failure of a process P at location l as $l[P] \,|\, l(X).\overline{fail}\, l \to 0 \,|\, \overline{fail}\, l$, migration from location l to location m as $l[P] \,|\, l(X).m[X] \to 0 \,|\, m[P]$, or duplication as $l[P] \,|\, l(X).(l[X] \,|\, l[X]) \to 0 \,|\, l[P] \,|\, l[P]$.

Name creation versus restriction: To our knowledge, previous process calculi with passivation—or, more generally, with higher-order distribution (i.e. communication of processes through channels across locations)—all used so-called *name restriction* [6], [18], [8], [11]. It hides names, forbidding reactions like $\overline{a}.Q \mid \nu a.(a.R_1 \mid \overline{a}.R_2) \to Q \mid \nu a.(R_1 \mid \overline{a}.R_2)$, where the syntax $\nu a.P$ in general means that name a is local to process P, hidden from the outside. Although the name a is textually present in the process $\overline{a}.Q$ above, the a under the ν operator is only visible to $a.R_1$ and $\overline{a}.R_2$, hence not usable for synchronisation with $\overline{a}.Q$.

Nonetheless, sharing of hidden names is possible via name *extrusion*, as in the reaction $l[\nu a.\overline{c}\langle a\rangle.Q] \mid m[c(x).R] \to \nu a.(l[Q]|m[R\{a/x\}])$. This reaction shows that the name a, which was local to location l, can be sent on another channel c and shared with the receiver outside l. In other words, extrusion extends the scope of the sent name to contain the receiver too, possibly crossing location boundaries outwards.

This makes name restriction harder to implement in higher-order distributed settings, as one needs to maintain the scope of extruded names across physically different locations. For example, guaranteeing that the process $\nu a.(l[Q] \,|\, m[R\{a/x\}])$ above cannot interfere with another process that coincidentally uses the same name a seems to require somehow keeping global information about its scope.

By contrast, real implementations of distributed systems often use *name creation* [19], which (perhaps against common belief) leads to a different equivalence theory from that of name restriction. Our name creation consists in forbidding transitions under a ν operator and in generating a fresh name as an internal transition step, shown in the following rule

$$\frac{a \notin s}{s \vdash \nu a.P \xrightarrow{\tau} s \cup \{a\} \vdash P}\textsc{Create}$$

where the syntax $s \vdash Q$ in general reads "process Q, given

a set s of already created names." This rule means that the process $\nu a.P$ can, in an internal transition step, create a name a that is stored immediately in the set of created names, and continue as process P. Assuming that we identify processes up-to alpha-conversion of names bound by the ν operator, the clause $a \notin s$ is a simple precaution to guarantee that freshly created names are indeed unique.

The creationist treatment of names makes the semantics closer to implementations: for example, suppose name creation is the generation of random numbers (arguably unique). Then, two different names in the model would actually be different numbers in the implementation too, thus ruling out interferences between processes and eliminating the need for explicit scope information.

Equivalence of higher-order distributed processes with name creation differs from that with name restriction. Concretely, consider the process

$$P = \nu l.(l[\nu a.(a \mid \overline{a}.\overline{a}.\overline{\omega})] \mid l(X).(X \mid X))$$

which, with name restriction semantics, can at best reduce (in several steps) to $\nu l.(0 \mid \nu a.(0 \mid \overline{a}.\overline{\omega}) \mid \nu a.(0 \mid \overline{a}.\overline{\omega}))$. With name creation semantics, there is also a reduction sequence that leads to the exhibition of name ω by having creation of name a happen *before* passivation and duplication:

$$\begin{array}{lll}
 & \{\omega\} & \vdash \nu l.(l[\nu a.(a \mid \overline{a}.\overline{a}.\overline{\omega})] \mid l(X).(X \mid X)) \\
\rightarrow & \{\omega, l\} & \vdash l[\nu a.(a \mid \overline{a}.\overline{a}.\overline{\omega})] \mid l(X).(X \mid X) \quad \text{(create } l\text{)} \\
\rightarrow & \{\omega, l, a\} & \vdash l[a \mid \overline{a}.\overline{a}.\overline{\omega}] \mid l(X).(X \mid X) \quad \text{(create } a\text{)} \\
\rightarrow & \{\omega, l, a\} & \vdash 0 \mid a \mid \overline{a}.\overline{a}.\overline{\omega} \mid a \mid \overline{a}.\overline{a}.\overline{\omega} \quad \text{(react on } l\text{)} \\
\rightarrow & \{\omega, l, a\} & \vdash 0 \mid 0 \mid \overline{a}.\overline{\omega} \mid a \mid \overline{a}.\overline{a}.\overline{\omega} \quad \text{(react on } a\text{)} \\
\rightarrow & \{\omega, l, a\} & \vdash 0 \mid 0 \mid \overline{\omega} \mid 0 \mid \overline{a}.\overline{a}.\overline{\omega} \quad \text{(react on } a\text{)}
\end{array}$$

Similarly, another (perhaps surprising) difference is the non-bisimilarity between $l[\nu a.\nu b.P]$ and $l[\nu b.\nu a.P]$ with $P = a.b.a.\overline{\omega_1} | \overline{a}.\overline{b}.\overline{b}.\overline{\omega_2}$, which are indistinguishable under name restriction. To see it, suppose $s \vdash l[\nu a.\nu b.P]$ with $s = \{l, \omega_1, \omega_2\}$ creates name a and is duplicated (i.e. is passivated and then spawned twice), after which the b of each copy is created, giving $s' \vdash (a.b_1.a.\overline{\omega_1} \mid \overline{a}.\overline{b_1}.\overline{b_1}.\overline{\omega_2}) \mid (a.b_2.a.\overline{\omega_1} \mid \overline{a}.\overline{b_2}.\overline{b_2}.\overline{\omega_2})$ with $s' = s \cup \{a, b_1, b_2\}$. Then, this process can exhibit $\overline{\omega_1}$ using both $\overline{a}$'s and a $\overline{b_1}$, giving $s' \vdash (\overline{\omega_1} \mid \overline{b_1}.\overline{\omega_2}) \mid (a.b_2.a.\overline{\omega_1} \mid \overline{b_2}.\overline{b_2}.\overline{\omega_2})$. Yet, it cannot exhibit $\overline{\omega_2}$ which is guarded by two $\overline{b_2}$'s while there is only one b_2. In order for $l[\nu b.\nu a.P]$ to weakly follow and exhibit $\overline{\omega_1}$ too, it must also share a, i.e. create it before duplication, forcing the creation and sharing of b as well. This gives $s \cup \{a, b\} \vdash (a.b.a.\overline{\omega_1} \mid \overline{a}.\overline{b}.\overline{b}.\overline{\omega_2}) \mid (a.b.a.\overline{\omega_1} \mid \overline{a}.\overline{b}.\overline{b}.\overline{\omega_2})$ which weakly exhibits not only $\overline{\omega_1}$ but also $\overline{\omega_2}$, therefore telling apart $l[\nu a.\nu b.P]$ and $l[\nu b.\nu a.P]$. More non-bisimilarities because of name creation semantics are discussed in Section VI. (We will note, however, that these processes are still mutually similar.)

In this paper, we argue that name creation is a realistic alternative to name restriction when modelling higher-order distribution. We recall that a restriction-based semantics is harder to implement, because of the difficulty of implementing distributed scope (which is inherent to the communication of bound names). Here, we discuss several of such semantics and their additional differences from name creation in a higher-order distributed setting. (i) A structural congruence rule $a[\nu c.P] \equiv \nu c.a[P]$ (cf. [1]). Under the presence of process duplication, it is "unsound," i.e. makes some inequivalent processes structurally congruent. For example, it allows $a[\nu c.(\overline{c}|c.c.\bar{\omega})] \equiv \nu c.a[\overline{c}|c.c.\bar{\omega}]$, but the two processes are distinguished by an observer $R = a(X).(X \mid X)$. (ii) Enforcing extrusion [18] before passivation like $l[\nu c.P] \mid l(X).(m[X] \mid n[X]) \rightarrow \nu c.(m[P] \mid n[P])$. It does not allow duplication without sharing private channels, keeping passivation from being used as a general device for copying. (iii) Forbidding passivation when ν is in evaluation position, i.e. $l[\nu c.P] \stackrel{\overline{l}\langle \nu c.P \rangle}{\nrightarrow} 0$. It hinders duplication with private channels as well. Moreover, expectedly equivalent processes $l[\nu a.a[P]]$ and $l[P]$ are distinguished by $l(X).\overline{\omega}$, which reacts only with $l[P]$. (iv) Vertical extrusion with an extra rule like $l[\nu c.P] \stackrel{\tau}{\rightarrow} \nu c.l[P]$. It differs from name creation too: consider $Q = l[m[\nu a.P] \mid m(X).(X \mid X)] \mid l(Y).(Y \mid Y)$ which can become $Q' = l[\nu a.m[P] \mid m(X).(X \mid X)] \mid l(Y).(Y \mid Y)$ in a step, and then (weakly) become either $\nu a.(P \mid P \mid P \mid P)$ with one bound name, or $\nu a.(P \mid P) \mid \nu a.(P \mid P)$ with two. With name creation, there is no reduction $Q \Rightarrow Q'$ such that only $Q' \Rightarrow P\{a_1/a\} \mid P\{a_1/a\} \mid P\{a_1/a\} \mid P\{a_1/a\}$ and $Q' \Rightarrow (P\{a_1/a\} \mid P\{a_1/a\}) \mid (P\{a_2/a\} \mid P\{a_2/a\})$ with a_1 and a_2 fresh, whence the difference.

Equivalence and inequivalence in higher-order distribution: We have just seen that equivalence differs depending on the semantics of names. Consequently, the equivalence theory under the presence of name creation needs to be rethought. Behavioural equivalence can be characterised as reduction-closed barbed equivalence (or congruence) [7] which has a simple definition but is impractical as a proof method because of a universal quantification on observer processes (or contexts) in its definition. Therefore, more convenient relations like bisimulations, whose membership implies reduction-closed barbed equivalence and which come with a co-inductive proof method, are sought after.

Accordingly, we define a theory of (environmental [20], [21], [14], [16], [11]) *bisimulation* for a higher-order π-calculus with both passivation and name creation. The theory is proven to be sound and, thanks to name creation semantics, complete. (In contrast, environmental bisimulations for higher-order π-calculus with passivation [11] were far from complete under name restriction semantics, being unsound without severe constraints on environments.) It can then be used to prove non-trivial equivalences that could not be shown previously [11], [8], like that of distributed left and right list folds (simplified versions of "MapReduce" [3]), detailed in Section V.

One may also want to prove bisimilarity of distributed programs that are more structurally different than the two fold functions, e.g. linear and logarithmic implementations of power functions. Perhaps surprisingly again, these implementations are *not* bisimilar. The reason is that the linear distributed implementation uses more hosts than the logarithmic one, and is therefore more likely to fail (under either of name

creation and restriction). Thus, bisimilarity may sometimes be too strong an equivalence. Instead, mutual *simulation* can be desirable, so as to provide a coarser equivalence (cf. [9, p. 20, Exercise 3.10]) still useful for comparing such programs. Environmental simulations can be proven, for example, between distributed linear and logarithmic power functions as detailed in Section VI-B.

Summary of our contributions: In this paper, we introduce the higher-order π-calculus with passivation and name creation (henceforth HOπPn) a dialect of HOπP (the higher-order π-calculus with passivation and name restriction) [8]. We then provide environmental bisimulations that are sound and complete with respect to reduction-closed barbed equivalence and (a reasonable form of) congruence, and use them to prove a non-trivial equivalence that could not be shown with previous methods. We also provide sound environmental simulations that can be used to show reduction-closed barbed *approximation*, and give a non-trivial simulation proof as well.

Overview of the paper: The rest of the paper is structured as follows. Section II defines HOπPn. Section III formalises our environmental bisimulations and simulations, and Section IV establishes their soundness and completeness. Section V shows the example bisimilarity proof of distributed left and right folds. Section VI discusses non-bisimilarity in higher-order distribution and shows the simulation proof of distributed power functions. Finally, Section VII considers previous and future work.

II. Higher-Order π-Calculus with Passivation and Name Creation

We formally introduce HOπPn through its syntax and labelled transition system. The syntax of HOπPn processes P, Q and terms M, N is given by the following grammar (the same as in [11]):

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

Briefly, X ranges over the set of variables, and a over names. 0 is a stuck process, $a(X)$ and $\overline{a}\langle M\rangle$ input and output prefixes, $\mid$ the parallel composition operator, and $a[P]$ the process P located at location a. νa is the name creation prefix, ! the replication operator, *run* the thawing operator which is used to turn a term into a process, X a variable and ${\scriptstyle `}P$ a process as a term. As in [11], the distinction between processes and terms is needed for our generic up-to context technique (see Section III).

The semantics of HOπPn is given by the following labelled transitions system which is based on that of the higher-order π-calculus with passivation [8]—itself being based on that of the higher-order π-calculus [12]—and is now defined on *configurations*. A configuration $s \vdash P$ is the pair of a set s of names and a process P such that $\mathit{fn}(P) \subseteq s$. We casually write sx or s, x for $s \cup \{x\}$ or $s \cup x$ when x is a name or a set of names. Omitting symmetric rules Par-R and React-R, the transition relation is defined inductively by the rules in

$$\frac{}{s \vdash a(X).P \xrightarrow{a(M)} s \vdash P\{M/X\}}\textsc{Ho-In} \qquad \frac{}{s \vdash \overline{a}\langle M\rangle.P \xrightarrow{\overline{a}\langle M\rangle} s \vdash P}\textsc{Ho-Out}$$

$$\frac{s \vdash P \xrightarrow{\alpha} s' \vdash P' \quad (s' \setminus s) \cap \mathit{fn}(Q) = \emptyset}{s \vdash P \mid Q \xrightarrow{\alpha} s' \vdash P' \mid Q}\textsc{Par-L}$$

$$\frac{s \vdash P \xrightarrow{\overline{a}\langle M\rangle} s \vdash P' \quad s \vdash Q \xrightarrow{a(M)} s \vdash Q'}{s \vdash P \mid Q \xrightarrow{\tau} s \vdash P' \mid Q'}\textsc{React-L}$$

$$\frac{s \vdash !P \mid P \xrightarrow{\alpha} s' \vdash P'}{s \vdash !P \xrightarrow{\alpha} s' \vdash P'}\textsc{Rep} \qquad \frac{a \notin s}{s \vdash \nu a.P \xrightarrow{\tau} s, a \vdash P}\textsc{Create}$$

$$\frac{s \vdash P \xrightarrow{\alpha} s' \vdash P}{s \vdash a[P] \xrightarrow{\alpha} s' \vdash a[P']}\textsc{Transp}$$

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

Fig. 1. Labelled transitions system of HOπPn

Figure 1. Assuming knowledge of the standard higher-order π-calculus [15], [12], [13], we skim over the distribution-related transitions and comment on the notable changes coming from name creation. The Transp rule expresses the *transparency* of locations—the fact that transitions can happen inside a location and be observed outside its boundaries. The Passiv rule shows how a process running inside a location can be *passivated*, i.e. stopped, turned into a term, and sent along a channel whose name corresponds to that of the location. The Run rule shows how to retrieve a process from a term at the cost of an internal transition.

The rule Create shows how a process $\nu a.P$ can create a name a—which is added to the configuration's set of names—and become P in an internal transition step. As we identify processes up-to alpha-conversion of bound names, progress is guaranteed. The rule Par-L shows that a transition can happen in a sub-process provided it does not create a name that is free in another sub-process put in parallel (the function fn, which returns the set of free names of a process or a term, is standard). Again, alpha-conversion is used for guaranteeing that no free name of Q will be captured.

The other rules are straightforward even in the presence of name creation and will not be discussed further. As usual with small-step semantics, when the assumptions cannot be satisfied or when a case is undefined (as in $\mathit{run}(X)$), transition does not progress and the process is stuck.

Henceforth, we shall write $\overline{a}.P$ for $\overline{a}\langle 0\rangle.P$ and $a.P$ for $a(X).P$ when X is not free in P. We define structural congruence $\equiv$ as the smallest congruence on processes with $P \equiv P \mid 0$, $P_1 \mid (P_2 \mid P_3) \equiv (P_1 \mid P_2) \mid P_3$, $P_1 \mid P_2 \equiv P_2 \mid P_1$ and $!P \equiv !P \mid P$. Notice that this definition is *not* standard: it allows neither $(\nu a.P) \mid Q \equiv \nu a.(P \mid Q)$, $\nu a.\nu b.P \equiv \nu b.\nu a.P$, nor $\nu a.0 \equiv 0$.

III. Environmental Bisimulation and Simulation for HOπPN

We define an *environmental relation* $\mathcal{X}$ as a set of sextuples $(\mathcal{E},\ r,\ s,\ P,\ t,\ Q)$ where $\mathcal{E}$ is a binary relation (called the *environment*) on terms with no free variables and finitely many free names, r is a finite set of names (the *public* names), s and t too are finite sets of names (the *created* names) such that $r \subseteq s \cap t$, and P and Q are variable-closed processes (the *tested processes*). We often write $(s \vdash P)\,\mathcal{X}_{\mathcal{E};r}\,(t \vdash Q)$ to mean $(\mathcal{E}, r, s, P, t, Q) \in \mathcal{X}$ for an environmental relation $\mathcal{X}$.

Definition 1. *We define multi-hole contexts for terms C (contexts that have holes for terms) and multi-hole contexts for processes C_p (contexts that have holes for processes) as:*

$$\begin{array}{l} C\ ::= 0 \mid a(X).C \mid \overline{a}\langle D\rangle.C \mid (C \mid C) \mid a[C] \mid \nu a.C \mid\ !C \mid \mathit{run}(D) \\ D\ ::= [\cdot]_i \mid X \mid\ `C \\ C_p ::= [\cdot]_i \mid 0 \mid a(X).C_p \mid \overline{a}\langle D_p\rangle.C_p \mid (C_p \mid C_p) \mid a[C_p] \mid \nu a.C_p \\ \qquad \mid\ !C_p \mid \mathit{run}(D_p) \\ D_p ::= X \mid\ `C_p \end{array}$$

Definition 2. *We define* process context closure *and* term context closure *as:*

$$\begin{array}{l} (\mathcal{E};r)^\circ = \{(C[\widetilde{M}], C[\widetilde{N}]) \mid \\ \qquad bn(C) \cap fn(\widetilde{M},\widetilde{N}) = \emptyset, fn(C) \subseteq r,\ (\widetilde{M},\widetilde{N}) \in \mathcal{E}\} \\ (\mathcal{E};r)^\star = \{(D[\widetilde{M}], D[\widetilde{N}]) \mid \\ \qquad bn(D) \cap fn(\widetilde{M},\widetilde{N}) = \emptyset, fn(D) \subseteq r,\ (\widetilde{M},\widetilde{N}) \in \mathcal{E}\} \end{array}$$

The process context closure $(\mathcal{E};r)^\circ$ intuitively represents the processes that an attacker can craft given some terms and public names. It allows him to create processes using terms $(\widetilde{M},\widetilde{N})$ from the environment and names from r. Capture of names is forbidden, hence the condition on names bound by the context. As this closure uses a context for terms, it will necessarily put its terms in an output prefix or under a *run*. The term context closure $(\mathcal{E};r)^\star$ intuitively corresponds to all the terms that the attacker can craft from his knowledge. We point out that these closure operations are monotonic on all their arguments, and thus for any $\mathcal{E}$ and r, $(\mathcal{E};r)^\star$ includes the identity $(\emptyset;r)^\star$.

A few extra notations are used in this paper. We define the weak transitions $\stackrel{\tau}{\Rightarrow}$ (or $\Rightarrow$) as the reflexive transitive closure of $\stackrel{\tau}{\rightarrow}$ (or $\rightarrow$), and $\stackrel{\alpha}{\Rightarrow}$ as $\stackrel{\tau}{\Rightarrow}\stackrel{\alpha}{\rightarrow}\stackrel{\tau}{\Rightarrow}$ for any $\alpha \neq \tau$. Finally, we write $a \oplus b$ to express the union $\{a\} \cup b$.

We now formally define environmental bisimulations (a subset of environmental relations):

Definition 3. *$\mathcal{X}$ is an environmental bisimulation if for all $(s \vdash P)\,\mathcal{X}_{\mathcal{E};r}\,(t \vdash Q)$,*

1) *if $s \vdash P \stackrel{\tau}{\rightarrow} s' \vdash P'$ then there is $t' \vdash Q'$ such that $t \vdash Q \stackrel{\tau}{\Rightarrow} t' \vdash Q'$ and $(s' \vdash P')\,\mathcal{X}_{\mathcal{E};r}\,(t' \vdash Q')$,*
2) *if $s \vdash P \stackrel{a(M)}{\longrightarrow} s \vdash P'$ with $a \in r$ and $(M,N) \in (\mathcal{E};r)^\star$, then there is $t' \vdash Q'$ such that $t \vdash Q \stackrel{a(N)}{\Longrightarrow} t' \vdash Q'$ and $(s \vdash P')\,\mathcal{X}_{\mathcal{E};r}\,(t' \vdash Q')$,*
3) *if $s \vdash P \stackrel{\overline{a}\langle M\rangle}{\longrightarrow} s \vdash P'$ with $a \in r$, then there are $t' \vdash Q'$ and N such that $t \vdash Q \stackrel{\overline{a}\langle N\rangle}{\Longrightarrow} t' \vdash Q'$ and $(s \vdash P')\,\mathcal{X}_{(M,N)\oplus\mathcal{E};r}\,(t' \vdash Q')$,*
4) *for all $l \in r$ and $(`P_1, `Q_1) \in \mathcal{E}$, we have $(s \vdash P \mid l[P_1])\,\mathcal{X}_{\mathcal{E};r}\,(t \vdash Q \mid l[Q_1])$,*
5) *for all $n \notin s \cup t$, we have $(s, n \vdash P)\,\mathcal{X}_{\mathcal{E};r,n}\,(t, n \vdash Q)$, and*
6) *the converse of the three first clauses, on Q's transitions.*

Clause 1 requires weak reduction closure and is fairly usual; clause 2 requires tested processes in a bisimulation to be able to input on a public channel any related terms that the attacker may create (hence the use of the term context closure), and to have their continuations in the bisimulation; clause 3 enlarges the knowledge of the attacker with terms that were output on a public channel, and requires the continuations to be bisimilar under this new knowledge; clause 4 allows the attacker to spawn and immediately run terms from the environment as processes in parallel to the tested processes (this allows to virtually consider an arbitrary process from the process context closure, while being much more tractable [11, Section 1]); clause 5 means that the attacker can create fresh names at will; finally, clause 6 is just the symmetric of the first three ones.

We remind that, because the set r of names is included in both s and t, we know that no name created by P nor Q will clash with r; this is why we do not need extra constraints on names like $(s' \setminus s) \cap r = \emptyset$ (in clause 1) and $(t' \setminus t) \cap r = \emptyset$ (in clauses 1, 2, 3), and why we do not have to require $n \notin r$ in clause 5. Also, using clause 5, the attacker can always generate fresh names before creating terms (that will use these new names) for input in clause 2.

As all the clauses of environmental bisimulations are monotonic on $\mathcal{X}$, the union of all bisimulations exists and is itself an environmental bisimulation. We call it environmental bisimilarity and write it $\sim$. For proving the equivalence of two processes P and Q, we show that $(f \vdash P) \sim_{\emptyset;r} (f \vdash Q)$ for some $r \subseteq f = fn(P,Q)$. It corresponds to equivalence where the attacker can send and receive messages over the public channels r of P and Q, but is yet to learn and put any term into the environment. Since $\sim$ is the union of all bisimulations, to prove this equivalence, it suffices to find an environmental bisimulation $\mathcal{X}$ such that $(f \vdash P)\,\mathcal{X}_{\emptyset;r}\,(f \vdash Q)$ with public names $r \subseteq f = fn(P,Q)$.

For improving the practicality of our proof method, we define an up-to context technique. Let us write $\mathcal{X}^\star$ for an environmental relation $\mathcal{X}$, such that:

$$\begin{array}{l} \mathcal{X}^\star = \{(\mathcal{E}, r, s, P, t, Q) \mid P \equiv P_0 \mid P_1,\ Q \equiv Q_0 \mid Q_1, \\ \qquad r' \cap (s \cup t) = \emptyset, \\ \qquad (s, r' \vdash P_0)\,\mathcal{X}_{\mathcal{E}';r,r'}\,(t, r' \vdash Q_0), \\ \qquad (P_1, Q_1) \in (\mathcal{E}'; rr')^\circ,\ \mathcal{E} \subseteq (\mathcal{E}'; rr')^\star\} \end{array}$$

Even though we call it "up-to context" for simplicity, it is in fact the combination of several up-to techniques: (i) "up-to context" since we allow the spawning of any related processes (P_1, Q_1) taken from the process context closure $(\mathcal{E}'; rr')^\circ$ of the knowledge $\mathcal{E}'$, rr' in parallel to the tested processes P_0 and Q_0 related by $(s, r' \vdash \cdot)\mathcal{X}_{\mathcal{E}';rr'}(t, r' \vdash \cdot)$; (ii) "up-to environment" since we allow, through the condition with the term context closure, the use of environments that are

larger than immediately necessary; (iii) "up-to name creation" since we allow the use of extra new names r'; and (iv) "up-to structural congruence" since we identify processes structurally congruent to $P_0 \mid P_1$ and $Q_0 \mid Q_1$. This convenient notation allows us to define environmental bisimulations up-to context:

Definition 4. *$\mathcal{X}$ is an environmental bisimulation up-to context if for all $(s \vdash P)\ \mathcal{X}_{\mathcal{E};r}\ (t \vdash Q)$,*

1) *if $s \vdash P \xrightarrow{\tau} s' \vdash P'$ then there is $t' \vdash Q'$ such that $t \vdash Q \stackrel{\tau}{\Rightarrow} t' \vdash Q'$ and $(s' \vdash P')\ \mathcal{X}^{\star}_{\mathcal{E};r}\ (t' \vdash Q')$,*
2) *if $s \vdash P \xrightarrow{a(M)} s \vdash P'$ with $a \in r$ and $(M, N) \in (\mathcal{E}; r)^{\star}$, then there is $t' \vdash Q'$ such that $t \vdash Q \stackrel{a(N)}{\Longrightarrow} t' \vdash Q'$ and $(s \vdash P')\ \mathcal{X}^{\star}_{\mathcal{E};r}\ (t' \vdash Q')$,*
3) *if $s \vdash P \xrightarrow{\overline{a}\langle M\rangle} s \vdash P'$ with $a \in r$, then there are $t' \vdash Q'$ and N such that $t \vdash Q \stackrel{\overline{a}\langle N\rangle}{\Longrightarrow} t' \vdash Q'$ and $(s \vdash P')\ \mathcal{X}^{\star}_{(M,N)\oplus\mathcal{E};r}\ (t' \vdash Q')$,*
4) *for all $l \in r$ and $(`P_1, `Q_1) \in \mathcal{E}$, we have $(s \vdash P \mid l[P_1])\ \mathcal{X}^{\star}_{\mathcal{E};r}\ (t \vdash Q \mid l[Q_1])$,*
5) *for all $n \notin s \cup t$, we have $(s, n \vdash P)\ \mathcal{X}_{\mathcal{E};r,n}\ (t, n \vdash Q)$, and*
6) *the converse of the three first clauses, on Q's transitions.*

This is basically the same definition as Definition 3 but all the positive instances of $\mathcal{X}$ became $\mathcal{X}^{\star}$ (except in clause 5 for technical reasons). Clause 4 is not a tautology since it spawns terms immediately as processes while the definition of $\mathcal{X}^{\star}$ allows only quoted processes. This distinction between quoted and non-quoted processes enables the use of generic contexts (as in [16], [11]) instead of redex contexts (as in [14]). Similarly to $\sim$, we define environmental bisimilarity up-to context and write it $\simeq$.

Finally, we define environmental similarity $\prec$ and similarity up-to context $\preceq$ by removing the converse conditions from the appropriate definitions.

IV. Soundness and Completeness of Environmental Bisimulation and Simulation

We outline here main results and proofs concerning the soundness and completeness of our proof method. More details are found in the appendix [10].

A. Behavioural Equivalences

We say process P has or exhibits barb a (resp. $\overline{a}$), written $P \downarrow_a$ (resp. $P \downarrow_{\overline{a}}$), whenever $P \xrightarrow{a(\cdot)} \cdot$ (resp. $P \xrightarrow{\overline{a}\langle\cdot\rangle} \cdot$). We say process P weakly exhibits barb μ, written $P \Downarrow \mu$, whenever $P \Rightarrow \downarrow \mu$ for a name or a co-name μ.

We can now formally define the equivalence predicates of our language, based on that of [7] (see also [15, Section 2.4.4]) with extensions for name creation.

Definition 5. *Reduction-closed barbed equivalence $\approx$ is the largest binary relation on variable-closed configurations, indexed with a set of names $r \subseteq s \cap t$, such that when $s \vdash P \approx_r t \vdash Q$,*

- *$s \vdash P \to s' \vdash P'$ implies there are Q' and t' such that $t \vdash Q \Rightarrow t' \vdash Q'$ and $s' \vdash P' \approx_r t' \vdash Q'$,*
- *$s \vdash P \downarrow_\mu$ implies $t \vdash Q \Downarrow_\mu$ if μ or $\overline{\mu}$ is in r,*
- *the converse of the above two on Q, and*
- *for all R with $fn(R) \cap ((s \cup t) \setminus r) = \emptyset$, we have $s \cup fn(R) \vdash P \mid R \approx_{r \cup fn(R)} t \cup fn(R) \vdash Q \mid R$.*

Note that we parameterised the equivalence with public names r. This is necessary for distinguishing the public names from private names that are not known to the attacker and cannot be observed nor used. This explains why the clause on barbs (and its symmetric) only considers barbs in r, and why in the last clause private names cannot be free in R. However, the free names created by the attacker are public and must thus be added to r for observation, and to s and t to avoid re-creation.

Definition 6. *Reduction-closed barbed congruence $\dot{\approx}$ is defined similarly to Definition 5, but replacing $\approx$ with $\dot{\approx}$ and the last clause with: for all C_p (context with holes for processes) such that $fn(C_p) \cap ((s \cup t) \setminus r) = bn(C_p) \cap fn(P, Q) = \emptyset$, we have $s \cup fn(C_p) \vdash C_p[P] \dot{\approx}_{r \cup fn(C_p)} t \cup fn(C_p) \vdash C_p[Q]$.*

It might be surprising that we consider a "congruence" which cannot capture public names ($bn(C_p) \cap fn(P, Q) = \emptyset$), but we argue that this is a reasonable definition. Indeed, free names in our language represent already created constants (private or public) in the compared processes; allowing the capture of names would virtually correspond to allowing in-place changes to the constant values in programs. Even though this may well tell some systems apart—as the attacker wishes to do—we doubt it represents a reasonable way to compare the behaviours of systems in execution contexts (in fact, this rather looks like using a binary editor to tell apart programs by modifying their code).

Definition 7. *Reduction-closed barbed approximation $\lessapprox$ is defined similarly to Definition 5, but replacing $\approx$ with $\lessapprox$ and removing the converse clauses. Respectively, reduction-closed barbed pre-congruence $\dot{\lessapprox}$ is defined similarly to Definition 6, but replacing $\dot{\approx}$ with $\dot{\lessapprox}$ and removing the converse clause.*

We say P approximates Q if $f \vdash P \lessapprox_r f \vdash Q$ with some $r \subseteq f = fn(P, Q)$. Intuitively, $P \lessapprox Q$ (or $P \dot{\lessapprox} Q$) whenever Q can do at least as much as P, in parallel with an observer R (or under a non-capturing context C_p).

B. Soundness

Theorem 1. *If $(s \vdash P) \simeq_{\mathcal{E};r} (t \vdash Q)$ then $(s \vdash P) \sim_{\mathcal{E};r} (t \vdash Q)$.*

Outline of proof: knowing $\simeq \subseteq \simeq^{\star}$ by definition, we show that:

1) transitions from $\simeq^{\star}$ lead to $\simeq^{-}$ (a superset of $\simeq^{\star}$ called run-erasure [10]),
2) if $(s \vdash P) \simeq^{-}_{\mathcal{E};r} (t \vdash Q)$ and $s \vdash P \xrightarrow{\overline{a}\langle M\rangle} s \vdash P'$, then $t \vdash Q \stackrel{\overline{a}\langle N\rangle}{\Longrightarrow} t' \vdash Q'$ and $(s \vdash P') \simeq^{-}_{(M,N)\oplus\mathcal{E};r} (t' \vdash Q')$, and using this result,
3) $\simeq^{-}$ is also closed by input and internal transitions.

It is then quite easy to show that $\simeq^-$ verifies all the clauses of environmental bisimulations, that is, $\simeq^- \subseteq \sim$. □

Corollary 1. *If* $(f \vdash P) \sim_{\emptyset;r} (f \vdash Q)$ *with* $r \subseteq f = fn(P,Q)$, *then* $f \vdash P \approx_r f \vdash Q$.

Outline of proof: we show that $\sim$ is reduction-closed (by definition), that it weakly exhibits the same barbs (by definition of bisimulation, ignoring the continuations after input or output transitions), and that it is preserved by parallel composition of arbitrary processes (that do not use private names) using the up-to context technique (with $\sim \subseteq \simeq$). □

It is interesting to remark that reduction-closed barbed *congruence* can easily be shown as follows. Let us define $P \mathrel{\dot{=}}_r Q$ if $(f \vdash 0) \sim_{\{(\text{`}P,\text{`}Q)\};r} (f \vdash 0)$ with $r \subseteq f = fn(P,Q)$. Then:

Theorem 2. *If* $P \mathrel{\dot{=}}_r Q$, *then* $f \vdash P \mathrel{\dot{\approx}}_r f \vdash Q$ *with* $r \subseteq f = fn(P,Q)$.

Outline of proof: We first show that a set relating run-erasures of $(C[\widetilde{M}], C[\widetilde{N}])$ for any non-capturing context C, with $(s \vdash 0) \sim_{\mathcal{E};r} (t \vdash 0)$ and $(\widetilde{M}, \widetilde{N}) \in \mathcal{E}$, is reduction-closed and verifies the conditions on barbs of reduction-closed barbed congruence. Then, by $P \mathrel{\dot{=}}_r Q$, i.e. $(f \vdash 0) \sim_{\{(\text{`}P,\text{`}Q)\};r} (f \vdash 0)$ with $r \subseteq f = fn(P,Q)$, we have $f \vdash P \mathrel{\dot{\approx}}_r f \vdash Q$. □

We emphasise that a capturing congruence cannot (and should not) be shown with this method. Omitting ‘ and *run* for brevity, we prove this by crafting a counter-example such that $P \mathrel{\dot{=}}_r Q$ but P and Q are not related by a name-capturing version of $\mathrel{\dot{\approx}}$.

Let $P_1 = a(X).(X \mid i(Y).\overline{m}) \mid m.\overline{\omega}$ and $Q_1 = a(X).(X \mid i(Y).Y) \mid m.\overline{\omega}$. We then consider the two processes $P = \nu i.\overline{b}\langle P_1\rangle \mid \overline{c}\langle\overline{a}\langle\overline{i}\langle\overline{m}\rangle\rangle\rangle$ and $Q = \nu i.\overline{b}\langle Q_1\rangle \mid \overline{c}\langle\overline{a}\langle\overline{i}\langle\overline{m}\rangle\rangle\rangle$ which are such that $P \mathrel{\dot{=}}_r Q$ for $r = \{a, m, b, c, \omega\}$. P and Q have been designed such that P can exhibit barb $\overline{\omega}$ if it receives anything on private channel i, while Q can exhibit barb $\overline{\omega}$ if it receives process $\overline{m}$ on private channel i.

By creating name i and then capturing the public name m with a context like $\overline{a}\langle\nu m.[\,]_1\rangle \mid a(X).(X \mid X)$, it is possible to reach a state where $\overline{\omega}$ and $\overline{m_1} \mid m_2.\overline{\omega}$ with private and different m_1 and m_2 would be related. However, as only the former process has barb $\overline{\omega}$, the equivalence cannot hold. This shows that comparing processes in a bisimulation environment is not enough to guarantee name-capturing reduction-closed barbed congruence.

It is in fact no problem that two processes in a bisimulation environment are not necessarily related by a name-capturing version of reduction-closed barbed congruence. Indeed, it is consistent with our idea that allowing capture of already created names is not a good basis for a congruence in languages with name creation like HOπPn.

Soundness also holds for simulations:

Theorem 3. *Let* $r \subseteq f = fn(P,Q)$. *If* $(f \vdash P) \preceq_{\emptyset;r} (f \vdash Q)$, *then* $(f \vdash P) \lesssim_r (f \vdash Q)$. *Respectively, if* $(f \vdash 0) \preceq_{\{(\text{`}P,\text{`}Q)\};r} (f \vdash 0)$, *then* $(f \vdash P) \mathrel{\dot{\lesssim}}_r (f \vdash Q)$.

The proof is immediate as our soundness proofs for environmental bisimulations do not use the symmetry condition and therefore can automatically be applied to environmental simulations too.

C. Completeness

Theorem 4. *If* $f \vdash P \approx_r f \vdash Q$ *with* $r \subseteq f = fn(P,Q)$, *then* $(f \vdash P) \sim_{\emptyset;r} (f \vdash Q)$.

Outline of proof: we find an environmental bisimulation $\mathcal{X}$ (up-to context) relating reduction-closed barbed equivalent P and Q. The trick is to use a parallel product of outputting processes to represent the environment. Roughly,

$$(s \vdash P \mid \textstyle\prod_i l_i[P_i])\ \mathcal{X}_{\mathcal{E};r}\ (t \vdash Q \mid \prod_i l_i[Q_i])$$

with $\{(\text{`}\widetilde{P}, \text{`}\widetilde{Q})\} \subseteq \mathcal{E}$ and $\widetilde{l} \in r$ is defined by

$$(s, \widetilde{g} \vdash P \mid \textstyle\prod_j !\overline{g_j}\langle\text{`}P_j\rangle) \approx_{r,\widetilde{g}} (t, \widetilde{g} \vdash Q \mid \prod_j !\overline{g_j}\langle\text{`}Q_j\rangle)$$

with $\{(\text{`}\widetilde{P}, \text{`}\widetilde{Q})\} = \mathcal{E}$. By using the last clause of reduction-closed barbed equivalence, one can create processes that will fetch the necessary $(\text{`}\widetilde{P}, \text{`}\widetilde{Q})$ and use them for crafting elements of the context closure $(\mathcal{E}; r)^\star$ needed in the input clause. The spawn clause is satisfied by construction. When accounting for a reaction like $P \mid l_1[P_1] \xrightarrow{\tau} P \mid l_1[P_1']$, one uses the last clause of reduction-closed barbed equivalence to create a receiver $l_1[g_1(X).X]$, spawns P_1 (and Q_1) inside this location l_1, mimics the reduction of P_1 (and the weak reactions of Q and Q_1 to Q' and Q_1'), and then passivates the contents of location l_1 to put $(\text{`}P_1', \text{`}Q_1')$ immediately in the representation of the "environment" under fresh name g_{j+1}, giving

$$s', \widetilde{g} \vdash P \mid \textstyle\prod_j !\overline{g_j}\langle\text{`}P_j\rangle \mid !\overline{g_{j+1}}\langle\text{`}P_1'\rangle \approx_{r,\widetilde{g}}$$
$$t', \widetilde{g} \vdash Q' \mid \textstyle\prod_j !\overline{g_j}\langle\text{`}Q_j\rangle \mid !\overline{g_{j+1}}\langle\text{`}Q_1'\rangle.$$

Therefore, processes $P \mid l_1[P_1']$ and $Q' \mid l[Q_1']$ are now related as wanted. □

Corollary 2. *If* $f \vdash P \mathrel{\dot{\approx}}_r f \vdash Q$ *with* $r \subseteq f = fn(P,Q)$, *then* $P \mathrel{\dot{=}}_r Q$.

Outline of proof: by the last clause of reduction-closed barbed congruence, we know that $(f \vdash P) \mathrel{\dot{\approx}}_r (f \vdash Q)$ implies $(a, f \vdash \overline{a}\langle P\rangle) \mathrel{\dot{\approx}}_{a,r} (a, f \vdash \overline{a}\langle Q\rangle)$ for fresh a, which itself implies $(a, f \vdash \overline{a}\langle P\rangle) \approx_{a,r} (a, f \vdash \overline{a}\langle Q\rangle)$. By Theorem 4, we thus have $(a, f \vdash \overline{a}\langle P\rangle) \sim_{\emptyset;a,r} (a, f \vdash \overline{a}\langle Q\rangle)$. We then output to a, get $(a, f \vdash 0) \sim_{\{(\text{`}P,\text{`}Q)\};a,r} (a, f \vdash 0)$, remove a up-to name creation, and we are done. □

We do not know yet whether completeness of simulation holds, since our current proofs rely on the symmetry conditions of the relations.

V. Bisimilarity Example

We present an example of equivalence that could not be proven with previous methods, remotely inspired by MapReduce [3] and abstracting the "reduce" part of it. More precisely, we show the bisimilarity between distributed left- and right-fold computations for arbitrary list l, associative function f,

and initial value i (the identity element of f). With car and cdr functions that return the head and tail of a list, we define the "fold servers" as:

$$L = !fl(f,i,l,k).\mathsf{if}\ null\ l\ \mathsf{then}\ \overline{k}\langle i\rangle\ \mathsf{else}\ \nu c.\overline{fl}\langle f, f\ i\ (car\ l), cdr\ l, c\rangle.c(m).\overline{k}\langle m\rangle$$
$$R = !fr(f,l,i,k).\mathsf{if}\ null\ l\ \mathsf{then}\ \overline{k}\langle i\rangle\ \mathsf{else}\ \nu c.\overline{fr}\langle f, cdr\ l, i, c\rangle.c(m).\overline{k}\langle f\ (car\ l)\ m\rangle$$

They are parameterised by (in addition to f, l and i) a channel k to return their results to clients (although omitted in the preceding sections, we remind our reader that first-order names and constants are easily added to the theory of environmental bisimulation [16]).

We then want to prove equivalent the configurations $a, b, fl \vdash P$ and $a, b, fr \vdash Q$ with public a and b, and where:

$$P = b(f,l,i,k).(\overline{fl}\langle f,i,l,k\rangle \mid a[L])$$
$$Q = b(f,l,i,k).(\overline{fr}\langle f,l,i,k\rangle \mid a[R])$$

To prove their equivalence, we provide a (strong) bisimulation $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2 \cup \mathcal{X}_3$ as in Figure 2 (where we use sans-serif fonts to denote Haskell-like functions). We will henceforth use the acronyms LHS and RHS for respectively the left-hand and right-hand sides of the bisimulation, i.e. the tested configurations. In this particular example, the same pairs of transitions verify the bisimulation clauses on both LHS and RHS's transitions; we will therefore only consider the transitions of LHS to avoid redundancy. We now analyse $\mathcal{X}_1$, which contains the configurations we want to identify. First, we observe that the set r in $\mathcal{X}_1$ contains at least the public names of P and Q, as required clause 5 of the bisimulation. Also, since the environment is empty, clause 4 is vacuously satisfied. Then, we consider the transitions, starting with $(fl, r \vdash P) \xrightarrow{b(f,i,l,k)} (fl, r \vdash \overline{fl}\langle f,i,l,k\rangle \mid a[L])$, which is matched by $(fr, r \vdash Q) \xrightarrow{b(f,i,l,k)} (fr, r \vdash \overline{fr}\langle f,l,i,k\rangle \mid a[R])$ so that membership to $\mathcal{X}_2$ is satisfied (by taking $n = 1$ and $a_1 = a$, with up-to environment since $\emptyset \subseteq \{(\text{`}L, \text{`}R)\}$). The name k was added to r in $\mathcal{X}_1$ by clause 5 of the bisimulation, and then input by clause 2.

Then, the elements of $\mathcal{X}_2$ must verify the spawn clause since their environment $\{(\text{`}L, \text{`}R)\}$ is not empty; spawning $l[L]$ and $l[R]$ for some $l \in r$ just enlarges the products $\prod_{j=1}^{n} a_j[\ ]$ by one element, preserving the membership to $\mathcal{X}_2$. Conversely, they can also do a passivation transition (which is a form of higher-order output): a pair $(\text{`}L, \text{`}R)$ is necessarily added to the environment (to which it already belongs) and the products shrink by one element; membership to $\mathcal{X}_2$ is thus preserved again. Finally, the reaction of $\overline{fl}$ with L (resp. $\overline{fr}$ with R) gives $(\{(\text{`}L, \text{`}R)\},\ r,\ r\,fl,\ \prod_{j=1}^{n-1} a_j[L] \mid a_n[L \mid P_0], r\,fr,\ \prod_{j=1}^{n-1} a_j[R] \mid a_n[R \mid Q_0])$ with $P_0 = \mathsf{if}\ null\ l\ \mathsf{then}\ \overline{k}\langle i\rangle\ \mathsf{else}\ \nu c.\overline{fl}\langle f, f\ i\ (car\ l), cdr\ l, c\rangle.c(m).\overline{k}\langle m\rangle$ and $Q_0 = \mathsf{if}\ null\ l\ \mathsf{then}\ \overline{k}\langle i\rangle\ \mathsf{else}\ \nu c.\overline{fr}\langle f, cdr\ l, i, c\rangle.c(m).\overline{k}\langle f\ (car\ l)\ m\rangle$, which belongs to $\mathcal{X}_3$ up-to environment since $\{(\text{`}L, \text{`}R)\} \subseteq \mathcal{E}$.

We now show that $\mathcal{X}_3$ satisfies the clauses of environmental bisimulation. Because all locations a_j in $\mathcal{X}_3$ are public and may thus lead to passivation, all P_j, Q_j must be in the environment (by clause 3 of the bisimulation); conversely, all terms from the environment must be spawnable an arbitrary number of times as located processes (by clause 4). It is straightforward to verify that $\mathcal{X}_3$ satisfies these constraints by definition. Respect of clause 5 is also immediate to verify.

$$\mathcal{X}_1 = \{(\emptyset,\ r,\ r\,fl,\ P,\ r\,fr,\ Q) \mid \{a,b\} \subseteq r,\ fl, fr \notin r\}$$
$$\mathcal{X}_2 = \{(\mathcal{E},\ r,\ r\,fl,\ \overline{fl}\langle f,i,l,k\rangle \mid \textstyle\prod_{j=1}^{n} a_j[L],\ r\,fr,\ \overline{fr}\langle f,l,i,k\rangle \mid \prod_{j=1}^{n} a_j[R]) \mid \mathcal{E} = \{(\text{`}L, \text{`}R)\},\ \{k,a,b,a_1,\ldots,a_n\} \subseteq r,\ fl, fr \notin r\}$$
$$\mathcal{X}_3 = \{(\mathcal{E}, r, r\,fl\,\widetilde{c}, \textstyle\prod_j a_j[P_j], r\,fr\,\widetilde{c}, \prod_j a_j[Q_j]) \mid \mathcal{E} = \{(\text{`}L \mid \textstyle\prod_h A_h,\ \text{`}R \mid \prod_h B_h) \mid (\widetilde{A}, \widetilde{B}) \in E^{f,l,i}(\mathsf{length}\ l,\ \{k\},\ r\,fl\,fr)\}, \{k,a,b,\widetilde{a}\} \subseteq r,\ \{\widetilde{c}\} = fn(\mathcal{E}) \setminus r \setminus \{fl, fr\}, (\text{`}\widetilde{P}, \text{`}\widetilde{Q}) \in \mathcal{E},\ fl, fr \notin r\}$$

$$E^{f,l,i}(0,\ rep,\ cre) = \{$$
$$(\mathsf{if}\ null\ []\ \mathsf{then}\ \overline{c_0}\langle f_d^l\rangle\ \mathsf{else}\ \nu c.\overline{fl}\langle f, f\ f_d^l\ (car\ []), cdr\ [], c\rangle.c(m).\overline{c_0}\langle m\rangle,$$
$$\mathsf{if}\ null\ []\ \mathsf{then}\ \overline{c_0}\langle i\rangle\ \mathsf{else}\ \nu c.\overline{fr}\langle f, cdr\ [], i, c\rangle.c(m).\overline{c_0}\langle f\ (car\ [])\ m\rangle),$$
$$(\overline{c_0}\langle f_d^l\rangle,\ \overline{c_0}\langle i\rangle) \mid c_0 \in rep,\ f_d^l = \mathsf{fold\text{-}left}\ f\ i\ l\}$$
$$E^{f,l,i}(m,\ rep,\ cre)\ (\text{when } m > 0) = \{$$
$$(\mathsf{if}\ null\ l_d\ \mathsf{then}\ \overline{c_m}\langle v_d^l\rangle\ \mathsf{else}\ \nu c.\overline{fl}\langle f, v_d^l, cdr\ l_d, c\rangle.c(o).\overline{c_m}\langle o\rangle,$$
$$\mathsf{if}\ null\ l_d\ \mathsf{then}\ \overline{c_m}\langle i\rangle\ \mathsf{else}\ \nu c.\overline{fr}\langle f, cdr\ l_d, i, c\rangle.c(o).\overline{c_m}\langle f\ v_d^r\ o\rangle),$$
$$(\nu c.\overline{fl}\langle f, v_d^l, cdr\ l_d, c\rangle.c(o).\overline{c_m}\langle o\rangle,$$
$$\nu c.\overline{fr}\langle f, cdr\ l_d, i, c\rangle.c(o).\overline{c_m}\langle f\ v_d^r\ o\rangle),$$
$$(\overline{fl}\langle f, v_d^l, cdr\ l_d, c_{m-1}\rangle.c_{m-1}(o).\overline{c_m}\langle o\rangle,$$
$$\overline{fr}\langle f, cdr\ l_d, i, c_{m-1}\rangle.c_{m-1}(o).\overline{c_m}\langle f\ v_d^r\ o\rangle),$$
$$(c_{m-1}(o).\overline{c_m}\langle o\rangle,\ c_{m-1}(o).\overline{c_m}\langle f\ v_d^r\ o\rangle),$$
$$(\overline{c_m}\langle f_d^l\rangle,\ \overline{c_m}\langle f_d^r\rangle), \quad (P,\ Q) \mid c_m \in rep,\ c_{m-1} \in rep',\ rep' \cap cre = \emptyset,\ rep' \text{ finite},$$
$$d = (\mathsf{length}\ l) - m,\ l_d = \mathsf{drop}\ d\ l,$$
$$v_d^l = \mathsf{fold\text{-}left}\ f\ i\ (\mathsf{take}\ d\ l),\ v_d^r = \mathsf{nth}\ d\ l,$$
$$f_d^l = \mathsf{fold\text{-}left}\ f\ i\ l,\ f_d^r = \mathsf{fold\text{-}right}\ i\ f\ l_d,$$
$$(P,\ Q) \in E^{f,l,i}(m-1,\ rep',\ rep' \cup cre)\}$$

Fig. 2. The partitions of the bisimulation $\mathcal{X}$

We then remark that $\mathcal{X}_3$ contains only locations a_j hosting elements of $\mathcal{E}$, i.e. the fold servers L and R in parallel with their related continuations A_h and B_h (if any). Therefore, in order to analyse the other transitions of elements of $\mathcal{X}_3$, we morally just have to consider the transitions of the elements of $\mathcal{E}$, i.e. the servers and their continuations. Those continuations are members of the set $E^{f,l,i}(\mathsf{length}\ l,\ \{k\},\ r\,fl\,fr)$, where E is a recursive function parametric in several values (see Figure 2). Concretely, the fixed parameters are the function f to fold, the initial list l and the initial value i. The varying parameters are the number m of elements yet to fold (hence $d = (\mathsf{length}\ l) - m$ is the current "depth" in the whole fold), a set rep of channels to return the result of the current recursive call, and a set cre of already created names.

Let us therefore consider first the transitions of $P_d = \mathsf{if}\ null\ l_d\ \mathsf{then}\ \overline{c_m}\langle v_d^l\rangle\ \mathsf{else}\ \ldots$ and related $Q_d = \mathsf{if}\ null\ l_d\ \mathsf{then}\ \overline{c_m}\langle i\rangle\ \mathsf{else}\ \ldots$ in some $E_m = E^{f,l,i}(m,\ rep,\ cre)$. If this is the "last recursive call," i.e. $m = 0$, then $l_d = []$ and $P_d \xrightarrow{\tau} \overline{c_m}\langle f_d^l\rangle$ where $f_d^l = \mathsf{fold\text{-}left}\ f\ i\ l$ is the final value (by definition of left fold), and $Q_d \xrightarrow{\tau} \overline{c_m}\langle i\rangle$ (by definition too). Since $\overline{c_m}\langle f_d^l\rangle$ and $\overline{c_m}\langle i\rangle$ are also related by E_m, these

transitions preserve membership to $\mathcal{X}_3$.

If $m > 0$ (i.e. the recursive call is not the last one), then the else branches are taken in both processes, giving $P'_d = \nu c.\overline{fl}\langle f, v^l_d, cdr\ l_d, c\rangle.c(o).\overline{c_m}\langle o\rangle$ in LHS and $Q'_d = \nu c.\overline{fr}\langle f, cdr\ l_d, i, c\rangle.c(o).\overline{c_m}\langle f\ v^r_d\ o\rangle)$ in RHS, still preserving membership to $\mathcal{X}_3$ since $(P'_d, Q'_d) \in E_m$.

Then, P'_d and Q'_d can both create a name c_{m-1} to become $P''_d = \overline{fl}\langle f, v^l_d, cdr\ l_d, c_{m-1}\rangle.c_{m-1}(o).\overline{c_m}\langle o\rangle$ and $Q''_d = \overline{fr}\langle f, cdr\ l_d, i, c_{m-1}\rangle.c_{m-1}(o).\overline{c_m}\langle f\ v^r_d\ o\rangle)$, provided c_{m-1} has not already been created, i.e. that c_{m-1} is not in cre. But remember that, by definition of $\mathcal{X}_3$ (that verifies clause 4 of the bisimulation), continuations P'_d and Q'_d can be spawned several times since they belong to $\mathcal{E}$ (along with L and R), and that each copy can thus create its own name c_{m-1}. Therefore, we must relate several (P''_d, Q''_d) all with their own fresh c_{m-1}; the set rep' exactly contains every such c_{m-1}. Notice that the names of rep' are free, allowing the definition of $\mathcal{X}_3$ to list them as $\{\widetilde{c}\}$, the set of names created by the folds.

Now, in order for P''_d to do a transition on private name fl, it must react with a server L, modelling a recursive call to the left fold on the rest of the current list. In this case, not only does P''_d reduce to $P'''_d = c_{m-1}(o).\overline{c_m}\langle o\rangle$, but the replication drawn from L turns into $P_{d+1} =$ if $null\ l_{d+1}$ then $\overline{c_{m-1}}\langle v^l_{d+1}\rangle$ else $\ldots$. Naturally, Q''_d follows as well, giving $Q'''_d = c_{m-1}(o).\overline{c_m}\langle f\ v^r_d\ o\rangle$ and $Q_{d+1} =$ if $null\ l_{d+1}$ then $\overline{c_{m-1}}\langle v^r_{d+1}\rangle$ else $\ldots$. Since (P'''_d, Q'''_d) belongs to E_m and (P_{d+1}, Q_{d+1}) as well (since $E_m \supseteq E_{m-1} = E^{f,l,i}(m-1,\ rep',\ rep' \cup cre)$ by definition), membership to $\mathcal{X}_3$ is still preserved. Notice that *any* L (and related R) may be used for the above reaction, even one at a location where some other continuations already exist. Because the P_{d+1} and Q_{d+1} add up next to the server they come from, the definition of $\mathcal{E}$ in $\mathcal{X}_3$ contains products of arbitrary length $\prod_h A_h$ and $\prod_h B_h$ in parallel with L and R. Analysis of the transitions of P_{d+1} and Q_{d+1} is the same as that of the transitions of P_d and Q_d and needs no further development.

Then, in order for P'''_d to do a transition, it must react on $c_{m-1} \in rep'$. By definition, the only processes that can send on c_{m-1} are $\overline{c_{m-1}}\langle f^l_{d+1}\rangle$ and $\overline{c_{m-1}}\langle f^r_{d+1}\rangle$ in $E_{m-1} \subseteq E_m$. Then P'''_d reacts with $\overline{c_{m-1}}\langle f^l_{d+1}\rangle$ and turns into $\overline{c_m}\langle f^l_d\rangle$ (with $f^l_d = f^l_{d+1} =$ fold-left $f\ i\ l$). Similarly, the process $\overline{c_{m-1}}\langle f^r_{d+1}\rangle$ reacts with Q'''_d which then turns into $\overline{c_m}\langle f\ v^r_d\ f^r_{d+1}\rangle$, i.e. $\overline{c_m}\langle f^r_d\rangle$, again preserving membership to $\mathcal{X}_3$.

Finally, the processes $\overline{c_m}\langle f^l_d\rangle$ and $\overline{c_m}\langle f^r_d\rangle$ may behave differently depending on where c_m comes from. If c_m is private, then they can react with some continuations $c_m(o).\overline{c_{m+1}}\langle o\rangle$ and $c_m(o).\overline{c_{m+1}}\langle f\ v_{r_{d-1}}\ o\rangle)$ that are related by $E_{m+1} = E^{f,l,i}(m+1,\ rep'',\ cre \setminus rep)$. Then, $\overline{c_m}\langle f^l_d\rangle$ and $\overline{c_m}\langle f^r_d\rangle$ both turn into 0 while the continuations' continuations are still related by E_{m+1}, like we showed for E_m in the previous paragraph. (There are no other reactions between elements of E_m and E_n with $m \neq n$.) Otherwise, necessarily $c_m = k$ by definition of $\mathcal{X}_3$ and thus, by definition of E, the same value fold-left $f\ i\ l =$ fold-right $f\ l\ i$ is output to public channel k.

This concludes our proof that elements of $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2 \cup \mathcal{X}_3$ satisfy the clauses of environmental bisimulations (up-to environment), and thus that $a, b, fl \vdash P$ and $a, b, fr \vdash Q$ are bisimilar with public names a and b.

VI. Non-Bisimilar Examples

A. Non-Bisimilarity Due to Different Internal Reactions

In the introduction, we gave an example of perhaps surprising (but rational) non-bisimilarity between located processes $l[\nu a.\nu b.P]$ and $l[\nu b.\nu a.P]$. A possibly even more surprising example would be the following:

$$\{l_1, \omega\} \vdash l_1[\nu a.(a|\overline{a}.\overline{\omega})] \not\approx_{\{\omega, l_1\}} \{l_1, \omega\} \vdash l_1[\nu a.\nu b.(a.b|\overline{a}.\overline{b}.\overline{\omega})]$$

To see why these configurations are not bisimilar, we consider the duplication of the located processes after the name creations; for $s = \{l_1, l_2, a, \omega\}$, we have:

$$s \vdash l_1[a \mid \overline{a}.\overline{\omega}] \mid l_2[a \mid \overline{a}.\overline{\omega}] \not\approx_{\{\omega, l_1, l_2\}}$$
$$s, b \vdash l_1[a.b \mid \overline{a}.\overline{b}.\overline{\omega}] \mid l_2[a.b \mid \overline{a}.\overline{b}.\overline{\omega}]$$

Consider now the transition of the right-hand side:

$$s \vdash l_1[a.b \mid \overline{a}.\overline{b}.\overline{\omega}] \mid l_2[a.b \mid \overline{a}.\overline{b}.\overline{\omega}] \xrightarrow{\tau} s \vdash l_1[b \mid \overline{a}.\overline{b}.\overline{\omega}] \mid l_2[a.b \mid \overline{b}.\overline{\omega}]$$

To match, the left-hand side may do a weak transition to one of the six following processes:

$$\begin{array}{lll} l_1[\overline{\omega}] \mid l_2[\overline{\omega}] & l_1[a \mid \overline{a}.\overline{\omega}] \mid l_2[\overline{\omega}] & l_1[\overline{\omega}] \mid l_2[a \mid \overline{a}.\overline{\omega}] \\ l_1[a \mid \overline{\omega}] \mid l_2[\overline{a}.\overline{\omega}] & l_1[\overline{a}.\overline{\omega}] \mid l_2[a \mid \overline{\omega}] & l_1[a \mid \overline{a}.\overline{\omega}] \mid l_2[a \mid \overline{a}.\overline{\omega}] \end{array}$$

with created names s. Suppose that the attacker then passivates l_1 on the right-hand side:

$$s \vdash l_1[b \mid \overline{a}.\overline{b}.\overline{\omega}] \mid l_2[a.b \mid \overline{b}.\overline{\omega}] \xrightarrow{\overline{l_1}\langle \text{`}b|\overline{a}.\overline{b}.\overline{\omega}\rangle} s \vdash l_2[a.b \mid \overline{b}.\overline{\omega}]$$

The resulting process is stuck, so because of the symmetry of bisimulations, the left-hand side must be able to passivate l_1 and become stuck too. The only way to achieve this is necessarily by doing a transition $\xRightarrow{\overline{l_1}\langle \text{`}a|\overline{\omega}\rangle} l_1, l_2, a, \omega \vdash l_2[\overline{a}.\overline{\omega}]$. The attacker can then passivate the contents of l_2 in both sides of the bisimulation, and be left with processes 0. Now, he can spawn back what was output during the passivation of l_1, and we thus have:

$$s \vdash l_1[a \mid \overline{\omega}] \not\approx_{\{\omega, l_1, l_2\}} s, b \vdash l_1[b \mid \overline{a}.\overline{b}.\overline{\omega}]$$

Obviously, the right-hand side is stuck, but the left one can exhibit $\overline{\omega}$, thus proving that the two processes are not bisimilar.

B. Non-Bisimilarity Due to Different Number of Locations Used

In Section V, we compared processes that recurse the same number of times and built bisimulations relating these processes such that whenever a process uses a location so does the other. We illustrate now that, in our distributed setting, the number of locations used *does* matter to draw some bisimilarity results.

Let us consider two implementations of the power function $pow(a, b) = a^b$, one of linear complexity, and the other of logarithmic complexity, as shown in Figure 3. Suppose that we want to replace the linear implementation P by the logarithmic one Q in a distributed system, and to check if

$P = !pow_{lin}(a,b,k).\mathsf{if}\ b=0$
$\quad \mathsf{then}\ \overline{k}\langle 1\rangle\ \mathsf{else}\ \nu c.\overline{pow_{lin}}\langle a, b{-}1, c\rangle.c(m).\overline{k}\langle a{\times}m\rangle$
$Q = !pow_{log}(a,b,k).\mathsf{if}\ b=0$
$\quad \mathsf{then}\ \overline{k}\langle 1\rangle\ \mathsf{else}\ \mathsf{if}\ b\%2=0\ \mathsf{then}\ \overline{pow_{log}}\langle a{\times}a, b{\div}2, k\rangle$
$\quad\quad \mathsf{else}\ \nu c.\overline{pow_{log}}\langle a, b{-}1, c\rangle.c(m).\overline{k}\langle a{\times}m\rangle$

Fig. 3. Linear and logarithmic power functions

there will be no visible difference. We could model those systems as $l[P] \mid \overline{pow_{lin}}\langle a,b,k\rangle$ and $l[Q] \mid \overline{pow_{log}}\langle a,b,k\rangle$ for integers a, b and public names l, k, and show their equivalence using our proof technique, trying to build a bisimulation $\mathcal{X}$ relating them, starting with $(\emptyset, lk, lk\ pow_{lin}, l[P] \mid \overline{pow_{lin}}\langle a,b,k\rangle, lk\ pow_{log}, l[Q] \mid \overline{pow_{log}}\langle a,b,k\rangle) \in \mathcal{X}$. We consider the situation where location l is passivated and then spawned b times: $(\emptyset, lk, lk\ pow_{lin}, \prod_{i=1}^{b} l_i[P] \mid \overline{pow_{lin}}\langle a,b,k\rangle, lk\ pow_{log}, \prod_{i=1}^{b} l_i[Q] \mid \overline{pow_{log}}\langle a,b,k\rangle) \in \mathcal{X}$. Let us consider now the state where the linear version has unfolded all the b recursive calls across $l_1, \ldots, l_b$, like: $l_1[P \mid c_1(x).\overline{k}\langle a{\times}x\rangle] \mid l_2[P \mid c_2(x).\overline{c_1}\langle a{\times}x\rangle] \mid \ldots \mid l_b[P \mid \overline{c_{b-1}}\langle 1\rangle]$. Similarly, the logarithmic version should (somehow) follow weakly: $l_1[Q \mid \ldots] \mid l_2[Q \mid \ldots] \mid \ \ldots \mid l_b[Q \mid \ldots]$. Suppose now that the attacker passivates the location l_1 that contains $c_1(x).\overline{k}\langle a{\times}x\rangle$, so that the left hand-side of the bisimulation can no longer return its result. Then, by the definition of bisimilarity, the logarithmic implementation too must not return a value if l_1 is passivated. The attacker may now spawn back the passivated contents of l_1, and then repeat the same passivation test on each of l_2, $l_3 \ldots$. In the end, we know that more than $log(b)$ locations were necessary for the recursive calls of the logarithmic version of the power function, which is impossible by design since this implementation can do at most $log(b)$ recursive calls. The two systems thus cannot be bisimilar.

C. Mutual Simulation

Although bisimilarity does not hold, we prove that the distributed linear power function approximates the logarithmic one by crafting a simulation relating them. We build the simulation $\mathcal{Y} = \mathcal{Y}_1 \cup \mathcal{Y}_2$ as in Figure 4, in a manner very similar to that of Section V. Our explanations below will thus focus on main differences from Section V. First, we recall that, because we consider (weak) simulations, we require that transitions by LHS be (weakly) matched by transitions on RHS, but not the converse. Thus, in this example, while the simulation $\mathcal{Y}$ needs to keep in LHS all the intermediate states of the linear power calculation, it suffices to keep in RHS only the initial and final states of the logarithmic power calculation.

The initial states are related by $\mathcal{Y}_1$. As far as intermediate states are concerned, we decide that the processes of LHS that can do an observable action (i.e. an output to k) be related to processes in RHS able to do the same action, so as to guarantee satisfaction of the output clause of simulation. Therefore, we define $\mathcal{Y}_2$ such that subprocesses of LHS that have k free in them (i.e. continuations of the initial call to pow_{lin}) are related to $\overline{k}\langle a^b\rangle$ on RHS (i.e. final state of the call to pow_{log}), and that other subprocesses of LHS are related to 0 on RHS.

$$
\begin{aligned}
\mathcal{Y}_1 = \{&(\mathcal{E}, r, r\ pow_{lin}, \textstyle\prod_i^n l_i[P] \mid \overline{pow_{lin}}\langle a,b,k\rangle,\\
&\quad r\ pow_{log}, \textstyle\prod_i^n l_i[Q] \mid \overline{pow_{log}}\langle a,b,k\rangle) \mid\\
&\quad k, \widetilde{l} \in r,\ \mathcal{E} \subseteq \{(\text{`}P, \text{`}Q)\}\}\\
\mathcal{Y}_2 = \{&(\mathcal{E}, r, \widetilde{c}\, r\ pow_{lin}, \textstyle\prod_i l_i[P_i], \widetilde{d}\, r\ pow_{log}, \prod_i l_i[Q_i]) \mid\\
&\quad \mathcal{E} = \{(P \mid \textstyle\prod_h A_h,\ Q \mid \prod_h B_h) \mid\\
&\qquad (\widetilde{A}, \widetilde{B}) \in E^{a,b}(b, \{k\}, pow_{lin}\ pow_{log}\ r)\},\\
&\quad (\text{`}\widetilde{P}, \text{`}\widetilde{Q}) \in \mathcal{E},\ k, \widetilde{l} \in r,\ \widetilde{c} = fn(\mathcal{E}.1) \setminus r \setminus \{pow_{lin}\},\\
&\quad \widetilde{d} \cap r \setminus \{pow_{log}\} = \emptyset,\ |\widetilde{d}| = depth(b),\\
&\quad depth(x) = \begin{cases} 0 & \text{if } x = 0 \\ depth(x/2) & \text{if } x\%2 = 0 \\ 1{+}depth(x{-}1) & \text{otherwise} \end{cases}\}\\
E^{a,b}(0, rep, cre) = \{&\\
&(\mathsf{if}\ 0=0\ \mathsf{then}\ \overline{c_0}\langle 1\rangle\ \mathsf{else}\ \ldots,\ R),\ (\overline{c_0}\langle 1\rangle,\ R) \mid\\
&c_0 \in rep,\ R = \overline{c_0}\langle 1\rangle \text{ if } b = 0, \text{ otherwise } R = 0\}\\
E^{a,b}(m, rep, cre) \text{ (when } m > 0) = \{&\\
&(\mathsf{if}\ m=0\ \mathsf{then}\ \ldots\ \mathsf{else}\ \nu c.\overline{pow_{lin}}\langle a, m{-}1, c\rangle.c(o).\overline{c_m}\langle a{\times}o\rangle,\ R),\\
&(\nu c.\overline{pow_{lin}}\langle a, m{-}1, c\rangle.c(o).\overline{c_m}\langle a{\times}o\rangle,\ R),\\
&(\overline{pow_{lin}}\langle a, m{-}1, c_{m-1}\rangle.c_{m-1}(o).\overline{c_m}\langle a{\times}o\rangle,\ R),\\
&(c_{m-1}(o).\overline{c_m}\langle a{\times}o\rangle,\ R),\ (\overline{c_m}\langle a^m\rangle,\ R),\ (P_{m-1},\ 0) \mid\\
&\quad c_m \in rep,\ c_{m-1} \in rep',\ rep' \cap cre = \emptyset,\ rep' \text{ finite},\\
&\quad P_{m-1} \in E^{a,b}(m{-}1, rep', rep' \cup cre),\\
&\quad R = \overline{c_m}\langle a^b\rangle \text{ if } m = b, \text{ otherwise } R = 0\}
\end{aligned}
$$

Fig. 4. Simulation $\mathcal{Y} = \mathcal{Y}_1 \cup \mathcal{Y}_2$ between linear and logarithmic power functions

We now show that the set $\mathcal{Y} = \mathcal{Y}_1 \cup \mathcal{Y}_2$ is a weak environmental simulation. We start with $\mathcal{Y}_1$ that relates (for $n = 1$ and $l_1 = l$) the processes we want to prove related: $lk\ pow_{lin} \vdash l[P] \mid \overline{pow_{lin}}\langle a,b,k\rangle$ and $lk\ pow_{log} \vdash l[Q] \mid \overline{pow_{log}}\langle a,b,k\rangle$). Suppose that the client $\overline{pow_{lin}}\langle a,b,k\rangle$ of LHS reacts with located server $l_i[P]$, leaving a process $P_b = (\mathsf{if}\ b = 0\ \mathsf{then}\ \overline{k}\langle 1\rangle\ \mathsf{else}\ \nu c.\overline{pow_{lin}}\langle a, b{-}1, c\rangle.c(m).\overline{k}\langle a{\times}m\rangle)$ at l_i. RHS can follow by reacting weakly (doing all calculations on the spot) with $l[Q]$, leaving process $l_i[\overline{k}\langle a^b\rangle]$. The rests are related by $\mathcal{Y}_2$ up-to environment since $\{(\text{`}P, \text{`}Q),\ (\text{`}P \mid P_b, \text{`}Q \mid \overline{k}\langle a^b\rangle)\} \subseteq \mathcal{E}$.

Then, if $b = 0$, P_b reduces to $\overline{k}\langle 1\rangle$, while RHS's $\overline{k}\langle a^b\rangle = \overline{k}\langle 1\rangle$ follows weakly by not doing any transition. The continuations $\overline{k}\langle 1\rangle$ and $\overline{k}\langle 1\rangle$ preserve membership to $\mathcal{Y}_2$. Moreover, if LHS's $\overline{k}\langle 1\rangle$ outputs 1 to k, then so can RHS's $\overline{k}\langle 1\rangle$ as expected.

Otherwise, if $b > 0$, P_b can reduce successively to $P_b' = \nu c.\overline{pow_{lin}}\langle a, b{-}1, c\rangle.\ldots$ and to $P_b'' = \overline{pow_{lin}}\langle a, b{-}1, c_b\rangle.\ldots$. In RHS, $\overline{k}\langle a^b\rangle$ follows both transitions weakly, still preserving membership to $\mathcal{Y}_2$.

Then P_b'' can react with a server P: P_b'' becomes $P_b''' = c_{b-1}(o).\overline{k}\langle a{\times}o\rangle$, P becomes $P \mid P_{b-1} = P \mid \mathsf{if}\ (b{-}1) = 0\ \mathsf{then}\ \overline{c_{b-1}}\langle 1\rangle\ \mathsf{else}\ \nu c.\overline{pow_{lin}}\langle a, b{-}2, c\rangle.c(m).\overline{c_{b-1}}\langle a{\times}m\rangle$, and $\overline{k}\langle a^b\rangle$ follows weakly. For example, we have $l_1[P \mid P_b''] \mid l_2[P] \xrightarrow{\tau} l_1[P \mid P_b'''] \mid l_2[P \mid P_{b-1}]$ in LHS, and $l_1[Q \mid \overline{k}\langle a^b\rangle] \mid l_2[Q] \xrightarrow{\tau} l_1[Q \mid \overline{k}\langle a^b\rangle] \mid l_2[Q \mid 0]$ in RHS. It results from the above that P_b''' is related to $\overline{k}\langle a^b\rangle$, and P_{b-1} to 0 since no process was added in RHS by the weak transition. Membership to $\mathcal{Y}_2$ is still satisfied.

We skip the analysis of transitions of P_{b-1}, as it similar to that of transitions of P_b.

If P_b''' inputs on c_{b-1}, it becomes $P_b'''' = \overline{k}\langle a^b\rangle$ because the only output to c_{b-1} comes from $\overline{c_{b-1}}\langle a^{b-1}\rangle$ in

$\underline{E}^{a,b}(b-1, rep', rep' \cup cre)$. RHS follows weakly, still giving $\overline{k}\langle a^b \rangle$, and preserving membership to $\mathcal{Y}_2$. Finally, output of a^b to k by both processes can happen, satisfying the simulation's output clause.

This concludes our proof that elements of $\mathcal{Y} = \mathcal{Y}_1 \cup \mathcal{Y}_2$ satisfy the clauses of environmental simulation (up-to environment), and thus that the distributed linear algorithm approximates the logarithmic one.

With the same approach, we may easily show that the linear algorithm simulates the logarithmic one as well (the previous simulation proof does not depend on the number of reduction steps on the left-hand side). This means that the two algorithms simulate each other even though they are not bisimilar, supporting the usefulness of mutual simulation in higher-order distribution.

By the same reasoning, the non-bisimilar processes in the introduction and Section VI-A can also be shown to be mutually similar.

VII. Discussions

We defined HOπPn, the higher-order distributed π-calculus with passivation and name creation, and developed its equivalence and inequivalence theories. Although many of the inequivalences may have been counter-intuitive, we emphasise that they are rational in hindsight and reflect the reality of non-linear higher-order distribution (not necessarily passivation but also duplication in general; cf. [2]).

Recently studied higher-order distributed process calculi include the Kell calculus [18], Homer [6] and the higher-order π-calculus with passivation (HOπP) [8]. They are extensions of the π-calculus with the communication of processes and their execution inside locations, and all have name restriction semantics. Other distributed process calculi such as Ambients [1] and Dpi [4] identify name creation and name restriction semantics, but are not higher-order in our sense (of passing processes through channels).

Research on the Kell calculus and Homer led to defining sound and complete *context bisimulations* [13]. However, they critically rely on universal quantification on contexts and are almost as hard as reduction-closed barbed equivalence as proof methods. Later, Lenglet et al. [8] focused on HOπP, a calculus simpler than both the Kell calculus and Homer. In addition to sound and complete context bisimulations, they provided more practical *normal bisimulations* [13] that are sound and complete in the *absence* of name restriction but are unsound otherwise. Also for HOπP, Piérard and Sumii defined a sound but incomplete environmental bisimulation proof technique [11] with strong constraints (the environments could not contain any restriction operator nor higher-order inputs). Even though non-trivial equivalences of processes which could not be realistically proven with context bisimulations can be proven with this technique, the constraints have a severe impact on the variety of processes that can be considered.

The simplicity of HOπPn, notably the transparency of locations, does not offer enough control over the communications between processes, and therefore hinders natural modelling of real systems where processes cannot freely interact with one another. Such systems can be modelled with non-transparent locations [18], [6], e.g. locations that only allow communications between processes from the same level or one level above/below. Moreover, passivation in HOπPn unifies failure, migration, and duplication as higher-order outputs, therefore mixing different behaviours. Even though identifying them keeps the model simple, their distinction may enable a more realistic modelling of higher-order distributed systems.

References

[1] L. Cardelli and A. D. Gordon. Mobile ambients. In *Foundations of Software Science and Computation Structures*, volume 1378 of *Lecture Notes in Computer Science*, pages 140–155. Springer, 1998.

[2] G. Castagna, J. Vitek, and F. Z. Nardelli. The Seal Calculus. *Information and Computation*, 201(1):1–54, 2005.

[3] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. In *Proceedings of the 6th conference on Symposium on Operating Systems Design and Implementation*, volume 6, pages 137–150. USENIX Association, 2004.

[4] M. Hennessy and J. Riely. Resource access control in systems of mobile agents. *Information and Computation*, 173:82–120, 1998.

[5] Hewlett-Packard. Live migration across data centers and disaster tolerant virtualization architecture with HP storageworks cluster extension and Microsoft Hyper-V. http://h20195.www2.hp.com/V2/GetPDF.aspx/4AA2-6905ENW.pdf.

[6] T. Hildebrandt, J. C. Godskesen, and M. Bundgaard. Bisimulation congruences for Homer: a calculus of higher-order mobile embedded resources. Technical Report TR-2004-52, IT University of Copenhagen, 2004.

[7] K. Honda and N. Yoshida. On reduction-based process semantics. *Theoretical Computer Science*, 151(2):437–486, 1995.

[8] S. Lenglet, A. Schmitt, and J.-B. Stefani. Normal bisimulations in calculi with passivation. In *Foundations of Software Science and Computational Structures*, volume 5504 of *Lecture Notes in Computer Science*, pages 257–271. Springer, 2009.

[9] R. Milner. *Communicating and Mobile Systems: the Pi-Calculus*. Cambridge University Press, 1999.

[10] A. Piérard and E. Sumii. Appendix to a higher-order distributed calculus with name creation. http://www.kb.ecei.tohoku.ac.jp/~adrien/pubs/AppendixCreation.pdf.

[11] A. Piérard and E. Sumii. Sound bisimulations for higher-order distributed process calculus. In *Foundations of Software Science and Computational Structures*, volume 6604 of *Lecture Notes in Computer Science*, pages 123–137. Springer, 2011.

[12] D. Sangiorgi. *Expressing Mobility in Process Algebras: First-Order and Higher-Order Paradigms*. PhD thesis, University of Edinburgh, 1992.

[13] D. Sangiorgi. Bisimulation for higher-order process calculi. *Information and Computation*, 131:141–178, 1996.

[14] D. Sangiorgi, N. Kobayashi, and E. Sumii. Environmental bisimulations for higher-order languages. *ACM Transactions on Programming Languages and Systems*, 33:5:1–5:69, 2011.

[15] D. Sangiorgi and D. Walker. *The π-calculus: a Theory of Mobile Processes*. Cambridge University Press, 2001.

[16] N. Sato and E. Sumii. The higher-order, call-by-value applied pi-calculus. In *Asian Symposium on Programming Languages and Systems*, volume 5904 of *Lecture Notes in Computer Science*, pages 311–326. Springer, 2009.

[17] D. Schmidt and P. Dhawan. Live migration with Xen virtualization software. http://www.dell.com/downloads/global/power/ps2q06-20050322-Schmidt-OE.pdf.

[18] A. Schmitt and J.-B. Stefani. The Kell calculus: A family of higher-order distributed process calculi. In *Global Computing*, volume 3267 of *Lecture Notes in Computer Science*, pages 146–178. Springer, 2004.

[19] I. Stark. *Names and Higher-Order Functions*. PhD thesis, University of Cambridge, 1994. Also available as Technical Report 363, University of Cambridge Computer Laboratory.

[20] E. Sumii and B. C. Pierce. A bisimulation for dynamic sealing. *Theoretical Computer Science*, 375(1-3):169–192, 2007.

[21] E. Sumii and B. C. Pierce. A bisimulation for type abstraction and recursion. *Journal of the ACM*, 54:1–43, 2007.

A Higher-order π-calculus with passivation and name creation

1 Syntax

The syntax of HOπPn processes P, Q is given by the following grammar:

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

We define the functions that returns the free names and free variables respectively as:

$$\begin{array}{llll} fn(0) & = \emptyset & fv(0) & = \emptyset \\ fn(a(X).P) & = \{a\} \cup fn(P) & fv(a(X).P) & = fv(P) \setminus \{X\} \\ fn(\overline{a}\langle M\rangle.P) & = \{a\} \cup fn(M) \cup fn(P) & fv(\overline{a}\langle M\rangle.P) & = fv(M) \cup fv(P) \\ fn(P_1 \mid P_2) & = fn(P_1) \cup fn(P_2) & fv(P_1 \mid P_2) & = fv(P_1) \cup fv(P_2) \\ fn(a[P]) & = \{a\} \cup fn(P) & fv(a[P]) & = fv(P) \\ fn(\nu a.P) & = fn(P) \setminus \{a\} & fv(\nu a.P) & = fv(P) \\ fn(!P) & = fn(P) & fv(!P) & = fv(P) \\ fn(run(M)) & = fn(M) & fv(run(M)) & = fv(M) \\ fn(X) & = \emptyset & fv(X) & = \{X\} \\ fn(`P) & = fn(P) & fv(`P) & = fv(P) \end{array}$$

We conveniently write $fn(X, Y, \ldots, Z)$ (resp. $fv(X, Y, \ldots, Z)$) to denote $\bigcup_{S \in \{X,Y,\ldots,Z\}} fn(S)$ (resp. $\bigcup_{S \in \{X,Y,\ldots,Z\}} fv(S)$).

2 Labelled transitions system

Definition A.1. [Configuration]
A configuration $s \vdash P$ is the pair of a set s of names and a process P such that $fn(P) \subseteq s$. We casually write sx or s, x for $s \cup \{x\}$ or $s \cup x$ when x is a name or a set of names.

The transitions semantics of HOπPn is given by the following labelled transitions system:

$$\frac{}{s \vdash a(X).P \xrightarrow{a(M)} s \vdash P\{M/X\}}\text{Ho-In} \qquad \frac{}{s \vdash \overline{a}\langle M\rangle.P \xrightarrow{\overline{a}\langle M\rangle} s \vdash P}\text{Ho-Out}$$

$$\frac{s \vdash P \xrightarrow{\alpha} s' \vdash P' \quad (s' \setminus s) \cap fn(Q) = \emptyset}{s \vdash P \mid Q \xrightarrow{\alpha} s' \vdash P' \mid Q}\text{Par-L}$$

$$\frac{s \vdash P \xrightarrow{\alpha} s' \vdash P' \quad (s' \setminus s) \cap fn(Q) = \emptyset}{s \vdash Q \mid P \xrightarrow{\alpha} s' \vdash Q \mid P'}\textsc{Par-R}$$

$$\frac{s \vdash P \xrightarrow{\overline{a}\langle M\rangle} s \vdash P' \quad s \vdash Q \xrightarrow{a(M)} s \vdash Q'}{s \vdash P \mid Q \xrightarrow{\tau} s \vdash P' \mid Q'}\textsc{React-L}$$

$$\frac{s \vdash P \xrightarrow{a(M)} s \vdash P' \quad s \vdash Q \xrightarrow{\overline{a}\langle M\rangle} s \vdash Q'}{s \vdash P \mid Q \xrightarrow{\tau} s \vdash P' \mid Q'}\textsc{React-R}$$

$$\frac{s \vdash !P \mid P \xrightarrow{\alpha} s' \vdash P'}{s \vdash !P \xrightarrow{\alpha} s' \vdash P'}\textsc{Rep} \qquad \frac{a \notin s}{s \vdash \nu a.P \xrightarrow{\tau} s, a \vdash P}\textsc{Create}$$

$$\frac{s \vdash P \xrightarrow{\alpha} s' \vdash P}{s \vdash a[P] \xrightarrow{\alpha} s' \vdash a[P']}\textsc{Transp} \qquad \frac{}{s \vdash a[P] \xrightarrow{\overline{a}\langle `P\rangle} s \vdash 0}\textsc{Passiv}$$

$$\frac{}{s \vdash run(`P) \xrightarrow{\tau} s \vdash P}\textsc{Run}$$

with the following function on labels

$$n(\alpha) = \begin{cases} \emptyset & \text{if } \alpha = \tau \\ \{a\} \cup fn(M) & \text{if } \alpha = a(M) \text{ or } \alpha = \overline{a}\langle M\rangle \end{cases}$$

and the notation $\widetilde{x}$ to denote the sequence $x_0, x_1, \ldots x_n$.

Definition A.2. *Structural congruence $\equiv$ is the smallest relation on processes such that:*

$$\frac{Q \equiv P}{P \equiv Q}\text{S-Sym} \qquad \frac{}{P \equiv P}\text{S-Refl} \qquad \frac{P \equiv R \quad R \equiv Q}{P \equiv Q}\text{S-Trans}$$

$$\frac{}{P \equiv P \mid 0}\text{S-Empty} \qquad \frac{}{P_1 \mid (P_2 \mid P_3) \equiv (P_1 \mid P_2) \mid P_3}\text{S-Assoc}$$

$$\frac{}{P_1 \mid P_2 \equiv P_2 \mid P_1}\text{S-Commut} \qquad \frac{P \equiv Q}{\nu a.P \equiv \nu a.Q}\text{S-Create} \qquad \frac{P \equiv Q}{a(X).P \equiv a(X).Q}\text{S-In}$$

$$\frac{P_1 \equiv Q_1 \quad P_2 \equiv Q_2}{\overline{a}\langle `P_1\rangle.P_2 \equiv \overline{a}\langle `Q_1\rangle.Q_2}\text{S-Out} \qquad \frac{}{!P \equiv !P \mid P}\text{S-Rep} \qquad \frac{P \equiv Q}{!P \equiv !Q}\text{S-Bang}$$

$$\frac{P_1 \equiv Q_1 \quad P_2 \equiv Q_2}{P_1 \mid P_2 \equiv Q_1 \mid Q_2}\text{S-Comp} \qquad \frac{P \equiv Q}{a[P] \equiv a[Q]}\text{S-Loc} \qquad \frac{P \equiv Q}{run(`P) \equiv run(`Q)}\text{S-Run}$$

Definition A.3. *Structural congruence on labels $\equiv$ is defined by:*

$$\frac{}{\tau \equiv \tau}\text{L-Tau} \qquad \frac{M \equiv N}{a(M) \equiv a(N)}\text{L-In} \qquad \frac{M \equiv N}{\overline{a}\langle M\rangle \equiv \overline{a}\langle N\rangle}\text{L-Out}$$

Lemma A.4. [Reduction preserves structural congruence]
If $P \equiv Q$ then

(a) for all α, P', s, s', if $s \vdash P \xrightarrow{\alpha} s' \vdash P'$ then either

i. there are a, M such that if $\alpha \equiv \overline{a}\langle M \rangle$ or $\alpha \equiv \tau$, then there are β, Q' such that $s \vdash Q \xrightarrow{\beta} s' \vdash Q'$, $\alpha \equiv \beta$ and $P' \equiv Q'$, or

ii. there are a, M such that if $\alpha \equiv a(M)$, then for all β such that $\alpha \equiv \beta$, there is Q' such that $s \vdash Q \xrightarrow{\beta} s' \vdash Q'$ and $P' \equiv Q'$, and

(b) for all α, Q', s, s', if $s \vdash Q \xrightarrow{\alpha} s' \vdash Q'$ then either

i. there are a, M such that if $\alpha \equiv \overline{a}\langle M \rangle$ or $\alpha \equiv \tau$, then there are β, P' such that $s \vdash P \xrightarrow{\beta} s \vdash P'$, $\alpha \equiv \beta$ and $P' \equiv Q'$, or

ii. there are a, M such that if $\alpha \equiv a(M)$, then for all β such that $\alpha \equiv \beta$, there is P' such that $s \vdash P \xrightarrow{\beta} s' \vdash P'$ and $P' \equiv Q'$.

Proof. By induction on the derivations of $P \equiv Q$.

B Environmental bisimulations of HOπPn

1 Generalities

Definition B.1. [Contexts]
We define multi-hole contexts for terms C (contexts that have holes for terms) and multi-hole contexts for processes C_p (contexts that have holes for processes) as:

$$\begin{array}{l} D_p ::= X \mid {\scriptstyle\lq}C_p \\ C_p ::= [\cdot]_i \mid 0 \mid a(X).C_p \mid \overline{a}\langle D_p\rangle.C_p \mid (C_p \mid C_p) \mid a[C_p] \mid \nu a.C_p \mid !C_p \mid run(D_p) \end{array}$$

$$\begin{array}{l} D ::= [\cdot]_i \mid X \mid {\scriptstyle\lq}C \\ C ::= 0 \mid a(X).C \mid \overline{a}\langle D\rangle.C \mid (C \mid C) \mid a[C] \mid \nu a.C \mid !C \mid run(D) \end{array}$$

Unless explicitly specified otherwise, the word "context" will denote a context for terms.

Definition B.2. [Context closures]
We write

$$\begin{array}{l} (\mathcal{E};r)^\circ = \{(C[\widetilde{M}], C[\widetilde{N}]) \mid bn(C) \cap fn(\widetilde{M},\widetilde{N}) = \emptyset,\ fn(C) \subseteq r,\ (\widetilde{M},\widetilde{N}) \in \mathcal{E}\} \\ (\mathcal{E};r)^\star = \{(D[\widetilde{M}], D[\widetilde{N}]) \mid bn(D) \cap fn(\widetilde{M},\widetilde{N}) = \emptyset,\ fn(D) \subseteq r,\ (\widetilde{M},\widetilde{N}) \in \mathcal{E}\} \end{array}$$

Definition B.3. [Reduction-closed barbed equivalence]
Reduction-closed barbed equivalence $\approx$ is the largest binary relation on configurations indexed with a set of names $r \subseteq s \cap t$ such that when $s \vdash P \approx_r t \vdash Q$,

- *$s \vdash P \xrightarrow{\tau} s' \vdash P'$ implies there are Q' and t' such that $t \vdash Q \Rightarrow t' \vdash Q'$ and $s' \vdash P' \approx_r t' \vdash Q'$,*
- *$s \vdash P \downarrow_\mu$ implies $t \vdash Q \Downarrow_\mu$ if μ or $\overline{\mu}$ is in r,*
- *the converse of the above two, on Q, and*
- *for all R such that $fn(R) \cap ((s \cup t) \setminus r) = \emptyset$, we have $s \cup fn(R) \vdash P \mid R \approx_{r \cup fn(R)} t \cup fn(R) \vdash Q \mid R$.*

Definition B.4. [Non-capturing reduction-closed barbed congruence]
Reduction-closed barbed congruence $\dot{\approx}$ is the largest binary relation on variable-closed configurations indexed with a set of names $r \subseteq s \cap t$ such that when $s \vdash P \dot{\approx}_r t \vdash Q$,

- *$s \vdash P \xrightarrow{\tau} s' \vdash P'$ implies there are Q' and t' such that $t \vdash Q \Rightarrow t' \vdash Q'$ and $s' \vdash P' \dot{\approx}_r t' \vdash Q'$,*
- *$s \vdash P \downarrow_\mu$ implies $t \vdash Q \Downarrow_\mu$ if μ or $\overline{\mu}$ is in r,*
- *the converse of the above two, on Q, and*
- *for all C context with holes for processes such that $fn(C) \cap ((s \cup t) \setminus r) = bn(C) \cap fn(P,Q) = \emptyset$, we have $s \cup fn(C) \vdash C[P] \dot{\approx}_{r \cup fn(C)} t \cup fn(C) \vdash C[Q]$.*

Note 1. Notice that this definition does not allow capturing names in r (nor s and t) and is therefore not that of a real congruence.

Definition B.5. [Reduction-closed barbed approximation]
Reduction-closed barbed approximation $\lessapprox$ is the largest binary relation on configurations indexed with a set of names $r \subseteq s \cap t$ such that when $s \vdash P \lessapprox_r t \vdash Q$,

- *$s \vdash P \xrightarrow{\tau} s' \vdash P'$ implies there are Q' and t' such that $t \vdash Q \Rightarrow t' \vdash Q'$ and $s' \vdash P' \lessapprox_r t' \vdash Q'$,*
- *$s \vdash P \downarrow_\mu$ implies $t \vdash Q \Downarrow_\mu$ if μ or $\overline{\mu}$ is in r,*
- *for all R such that $fn(R) \cap ((s \cup t) \setminus r) = \emptyset$, we have $s \cup fn(R) \vdash P \mid R \lessapprox_{r \cup fn(R)} t \cup fn(R) \vdash Q \mid R$.*

Definition B.6. [Non-capturing reduction-closed barbed pre-congruence]
Reduction-closed barbed pre-congruence $\dot{\lessapprox}$ is the largest binary relation on variable-closed configurations indexed with a set of names $r \subseteq s \cap t$ such that when $s \vdash P \dot{\lessapprox}_r t \vdash Q$,

- *$s \vdash P \xrightarrow{\tau} s' \vdash P'$ implies there are Q' and t' such that $t \vdash Q \Rightarrow t' \vdash Q'$ and $s' \vdash P' \dot{\lessapprox}_r t' \vdash Q'$,*
- *$s \vdash P \downarrow_\mu$ implies $t \vdash Q \Downarrow_\mu$ if μ or $\overline{\mu}$ is in r,*
- *for all C context with holes for processes such that $fn(C) \cap ((s \cup t) \setminus r) = bn(C) \cap fn(P, Q) = \emptyset$, we have $s \cup fn(C) \vdash C[P] \dot{\lessapprox}_{r \cup fn(C)} t \cup fn(C) \vdash C[Q]$.*

Note 2. Notice that this definition does not allow capturing names in r (nor s and t) and is therefore not that of a real pre-congruence.

Definition B.7. [Environmental simulation]
$\mathcal{X}$ is an environmental simulation if for all $(s \vdash P)\ \mathcal{X}_{\mathcal{E};r}\ (t \vdash Q)$,

1. *if $s \vdash P \xrightarrow{\tau} s' \vdash P'$ then there is $t' \vdash Q'$ such that $t \vdash Q \stackrel{\tau}{\Rightarrow} t' \vdash Q'$ and $(s' \vdash P')\ \mathcal{X}_{\mathcal{E};r}\ (t' \vdash Q')$,*
2. *if $s \vdash P \xrightarrow{a(M)} s \vdash P'$ with $a \in r$, then for all $(M, N) \in (\mathcal{E}; r)^\star$ there is $t' \vdash Q'$ such that $t \vdash Q \stackrel{a(N)}{\Longrightarrow} t' \vdash Q'$ and $(s \vdash P')\ \mathcal{X}_{\mathcal{E};r}\ (t' \vdash Q')$,*
3. *if $s \vdash P \xrightarrow{\overline{a}\langle M \rangle} s \vdash P'$ with $a \in r$, then there are $t' \vdash Q'$ and N such that $t \vdash Q \stackrel{\overline{a}\langle N \rangle}{\Longrightarrow} t' \vdash Q'$ and $(s \vdash P')\ \mathcal{X}_{(M,N) \oplus \mathcal{E};r}\ (t' \vdash Q')$,*
4. *for all $l \in r$ and $(`P_1, `Q_1) \in \mathcal{E}$, we have $(s \vdash P \mid l[P_1])\ \mathcal{X}_{\mathcal{E};r}\ (t \vdash Q \mid l[Q_1])$, and*
5. *for all $n \notin s \cup t$, we have $(s, n \vdash P)\ \mathcal{X}_{\mathcal{E};r,n}\ (t, n \vdash Q)$.*

Definition B.8. [Environmental bisimulation]
$\mathcal{X}$ is an environmental bisimulation if for all $(s \vdash P)\ \mathcal{X}_{\mathcal{E};r}\ (t \vdash Q)$,

1. *if $s \vdash P \xrightarrow{\tau} s' \vdash P'$ then there is $t' \vdash Q'$ such that $t \vdash Q \stackrel{\tau}{\Rightarrow} t' \vdash Q'$ and $(s' \vdash P')\ \mathcal{X}_{\mathcal{E};r}\ (t' \vdash Q')$,*

2. *if* $s \vdash P \xrightarrow{a(M)} s \vdash P'$ *with* $a \in r$, *then for all* $(M,N) \in (\mathcal{E};r)^\star$ *there is* $t' \vdash Q'$ *such that* $t \vdash Q \xRightarrow{a(N)} t' \vdash Q'$ *and* $(s \vdash P')\ \mathcal{X}_{\mathcal{E};r}\ (t' \vdash Q')$,
3. *if* $s \vdash P \xrightarrow{\overline{a}\langle M\rangle} s \vdash P'$ *with* $a \in r$, *then there are* $t' \vdash Q'$ *and* N *such that* $t \vdash Q \xRightarrow{\overline{a}\langle N\rangle} t' \vdash Q'$ *and* $(s \vdash P')\ \mathcal{X}_{(M,N)\oplus\mathcal{E};r}\ (t' \vdash Q')$,
4. *for all* $l \in r$ *and* $(`P_1, `Q_1) \in \mathcal{E}$, *we have* $(s \vdash P \mid l[P_1])\ \mathcal{X}_{\mathcal{E};r}\ (t \vdash Q \mid l[Q_1])$,
5. *for all* $n \notin s \cup t$, *we have* $(s, n \vdash P)\ \mathcal{X}_{\mathcal{E};r,n}\ (t, n \vdash Q)$, *and*
6. *the converse of the three first clauses, on* Q*'s transitions.*

Definition B.9. [Context closure of an environmental relation]
We define

$$\begin{aligned}\mathcal{X}^\star = \{(\mathcal{E}, r, s, P, t, Q) \mid\ & P \equiv P_0|P_1,\\ & Q \equiv Q_0|Q_1,\\ & (s, r' \vdash P_0)\ \mathcal{X}_{\mathcal{E}';r,r'}\ (t, r' \vdash Q_0),\\ & (P_1, Q_1) \in (\mathcal{E}'; rr')^\circ,\\ & \mathcal{E} \subseteq (\mathcal{E}'; rr')^\star,\\ & r' \cap (s \cup t) = \emptyset\}\end{aligned}$$

Definition B.10. [Environmental simulation up-to context]
$\mathcal{X}$ *is an environmental simulation up-to context if for all* $(s \vdash P)\ \mathcal{X}_{\mathcal{E};r}\ (t \vdash Q)$,

1. *if* $s \vdash P \xrightarrow{\tau} s' \vdash P'$ *then there is* $t' \vdash Q'$ *such that* $t \vdash Q \xRightarrow{\tau} t' \vdash Q'$ *and* $(s' \vdash P')\ \mathcal{X}^\star_{\mathcal{E};r}\ (t' \vdash Q')$,
2. *if* $s \vdash P \xrightarrow{a(M)} s \vdash P'$ *with* $a \in r$, *then for all* $(M,N) \in (\mathcal{E};r)^\star$ *there is* $t' \vdash Q'$ *such that* $t \vdash Q \xRightarrow{a(N)} t' \vdash Q'$ *and* $(s \vdash P')\ \mathcal{X}^\star_{\mathcal{E};r}\ (t' \vdash Q')$,
3. *if* $s \vdash P \xrightarrow{\overline{a}\langle M\rangle} s \vdash P'$ *with* $a \in r$, *then there are* $t' \vdash Q'$ *and* N *such that* $t \vdash Q \xRightarrow{\overline{a}\langle N\rangle} t' \vdash Q'$ *and* $(s \vdash P')\ \mathcal{X}^\star_{(M,N)\oplus\mathcal{E};r}\ (t' \vdash Q')$,
4. *for all* $l \in r$ *and* $(`P_1, `Q_1) \in \mathcal{E}$, *we have* $(s \vdash P \mid l[P_1])\ \mathcal{X}^\star_{\mathcal{E};r}\ (t \vdash Q \mid l[Q_1])$, *and*
5. *for all* $n \notin s \cup t$, *we have* $(s, n \vdash P)\ \mathcal{X}_{\mathcal{E};r,n}\ (t, n \vdash Q)$.

Definition B.11. [Environmental bisimulation up-to context]
$\mathcal{X}$ *is an environmental bisimulation up-to context if for all* $(s \vdash P)\ \mathcal{X}_{\mathcal{E};r}\ (t \vdash Q)$,

1. *if* $s \vdash P \xrightarrow{\tau} s' \vdash P'$ *then there is* $t' \vdash Q'$ *such that* $t \vdash Q \xRightarrow{\tau} t' \vdash Q'$ *and* $(s' \vdash P')\ \mathcal{X}^\star_{\mathcal{E};r}\ (t' \vdash Q')$,
2. *if* $s \vdash P \xrightarrow{a(M)} s \vdash P'$ *with* $a \in r$, *then for all* $(M,N) \in (\mathcal{E};r)^\star$ *there is* $t' \vdash Q'$ *such that* $t \vdash Q \xRightarrow{a(N)} t' \vdash Q'$ *and* $(s \vdash P')\ \mathcal{X}^\star_{\mathcal{E};r}\ (t' \vdash Q')$,
3. *if* $s \vdash P \xrightarrow{\overline{a}\langle M\rangle} s \vdash P'$ *with* $a \in r$, *then there are* $t' \vdash Q'$ *and* N *such that* $t \vdash Q \xRightarrow{\overline{a}\langle N\rangle} t' \vdash Q'$ *and* $(s \vdash P')\ \mathcal{X}^\star_{(M,N)\oplus\mathcal{E};r}\ (t' \vdash Q')$,
4. *for all* $l \in r$ *and* $(`P_1, `Q_1) \in \mathcal{E}$, *we have* $(s \vdash P \mid l[P_1])\ \mathcal{X}^\star_{\mathcal{E};r}\ (t \vdash Q \mid l[Q_1])$,
5. *for all* $n \notin s \cup t$, *we have* $(s, n \vdash P)\ \mathcal{X}_{\mathcal{E};r,n}\ (t, n \vdash Q)$, *and*
6. *the converse of the three first clauses, on* Q*'s transitions.*

2 Soundness of environmental bisimulations

Lemma B.12. *If* $(P,Q) \in (\mathcal{E};r)^\circ$ *and* $s \vdash P \xrightarrow{a(M)} s \vdash P'$ *then for all* N *there is* Q' *such that* $t \vdash Q \xrightarrow{a(N)} t \vdash Q'$ *and* $(P',Q') \in ((M,N)\oplus\mathcal{E};r)^\circ$.

Proof. By induction on the transition derivation $s \vdash P \xrightarrow{a(M)} s \vdash P'$. There are five cases to check.

1. **Case** In: $C = a(X).C_1$
   We have that $s \vdash P = s \vdash a(X).C_1[\widetilde{M}] \xrightarrow{a(M)} s \vdash C_1[\widetilde{M}]\{M/X\}$ and that $t \vdash Q = t \vdash a(X).C_1[\widetilde{N}] \xrightarrow{a(N)} t \vdash C_1[\widetilde{N}]\{N/X\}$. We are done since we replace term X by terms M and N, hence $(C_1[\widetilde{M}]\{M/X\}, C_1[\widetilde{N}]\{N/X\}) \in ((M,N)\oplus\mathcal{E};r)^\circ$.
2. **Case** Par-L: $C = C_1 \mid C_2$
   We have that $s \vdash P = s \vdash C_1[\widetilde{M}] \mid C_2[\widetilde{M}] \xrightarrow{a(M)} P_1' \mid C_2[\widetilde{M}]$, i.e. $s \vdash C_1[\widetilde{M}] \xrightarrow{a(M)} s \vdash P_1'$. By the induction hypothesis $t \vdash C_1[\widetilde{N}] \xrightarrow{a(N)} t \vdash Q_1'$ and $(P_1', Q_1') \in ((M,N)\oplus\mathcal{E};r)^\circ$, from which we derive $((P_1' \mid C_2[\widetilde{M}]), (Q_1' \mid C_2[\widetilde{N}])) \in ((M,N)\oplus\mathcal{E};r)^\circ$ as well as $t \vdash C_1[\widetilde{N}] \mid C_2[\widetilde{N}] \xrightarrow{a(N)} t \vdash Q_1' \mid C_2[\widetilde{N}]$.
3. **Case** Par-R: $C = C_1 \mid C_2$
   Similar.
4. **Case** Transp: $C = l[C_1]$
   We have that $s \vdash P = s \vdash l[C_1[\widetilde{M}]] \xrightarrow{a(M)} s \vdash l[P_1']$, that is $s \vdash C_1[\widetilde{M}] \xrightarrow{a(M)} s \vdash P_1'$. By the induction hypothesis, we have that $t \vdash C_1[\widetilde{N}] \xrightarrow{a(N)} t \vdash Q_1'$ and $(P_1', Q_1') \in ((M,N)\oplus\mathcal{E};r)^\circ$, from which we derive $(l[P_1'], l[Q_1']) \in ((M,N)\oplus\mathcal{E};r)^\circ$ as well as $t \vdash l[C_1[\widetilde{N}]] \xrightarrow{a(N)} t \vdash l[Q_1']$.
5. **Case** Rep: $C = !C_1$
   We have that $s \vdash P = s \vdash !C_1[\widetilde{M}] \xrightarrow{a(M)} s \vdash P'$, i.e. $s \vdash !C_1[\widetilde{M}] \mid C_1[\widetilde{M}] \xrightarrow{a(M)} s \vdash P'$. By the induction hypothesis, we have that $t \vdash !C_1[\widetilde{N}] \mid C_1[\widetilde{N}] \xrightarrow{a(N)} t \vdash Q'$ and $(P',Q') \in ((M,N)\oplus\mathcal{E};r)^\circ$. Thus $t \vdash !C_1[\widetilde{N}] \xrightarrow{a(N)} t \vdash Q'$ and still $(P',Q') \in ((M,N)\oplus\mathcal{E};r)^\circ$.

Lemma B.13. *If* $(P,Q) \in (\mathcal{E};r)^\circ$ *and* $s \vdash P \xrightarrow{\overline{a}\langle M\rangle} s \vdash P'$ *then there are* Q' *and* N *such that* $t \vdash Q \xrightarrow{\overline{a}\langle N\rangle} t \vdash Q'$, $(P',Q') \in (\mathcal{E};r)^\circ$ *and* $(M,N) \in (\mathcal{E};r)^\star$.

Proof. By induction on the transition derivation $P \xrightarrow{\overline{a}\langle M\rangle} P'$. There are six cases to check.

1. **Case** Output: $C = \overline{a}\langle \text{`}C_1\rangle.C_2$
   We have that $s \vdash P = \overline{a}\langle \text{`}C_1[\widetilde{M}]\rangle.C_2[\widetilde{M}] \xrightarrow{\overline{a}\langle \text{`}C_1[\widetilde{M}]\rangle} s \vdash C_2[\widetilde{M}]$ and that $t \vdash Q_1 = \overline{a}\langle \text{`}C_1[\widetilde{N}]\rangle.C_2[\widetilde{N}] \xrightarrow{\overline{a}\langle \text{`}C_1[\widetilde{N}]\rangle} t \vdash C_2[\widetilde{N}]$. It is immediate to confirm that $(\text{`}C_1[\widetilde{M}], \text{`}C_1[\widetilde{N}]) \in (\mathcal{E};r)^\star$ and $(C_2[\widetilde{M}], C_2[\widetilde{N}]) \in (\mathcal{E};r)^\circ$ hold.

2. **Case** PAR-L: $C = C_1 \mid C_2$
   We have that $s \vdash P = C_1[\widetilde{M}] | C_2[\widetilde{M}] \xrightarrow{\overline{a}\langle M\rangle} s \vdash P_1' | C_2[\widetilde{M}]$, i.e. $s \vdash C_1[\widetilde{M}] \xrightarrow{\overline{a}\langle M\rangle} s \vdash P_1'$. By the induction hypothesis, we have that $t \vdash C_1[\widetilde{N}] \xrightarrow{\overline{a}\langle N\rangle} t \vdash Q_1'$ and $(P_1', Q_1') \in (\mathcal{E}; r)^\circ$ and $(M, N) \in (\mathcal{E}; r)^\star$. Hence $t \vdash C_1[\widetilde{N}] \mid C_2[\widetilde{N}] \xrightarrow{\overline{a}\langle N\rangle} t \vdash Q_1' \mid C_2[\widetilde{N}]$, and $((P_1' \mid C_2[\widetilde{M}]), (Q_1' \mid C_2[\widetilde{N}])) \in (\mathcal{E}; r)^\circ$.
3. **Case** PAR-R: $C = C_1 \mid C_2$
   Similar.
4. **Case** TRANSP: $C = l[C_1]$
   We have that $s \vdash P = l[C_1[\widetilde{M}]] \xrightarrow{\overline{a}\langle M\rangle} s \vdash l[P_1']$, i.e. $s \vdash C_1[\widetilde{M}] \xrightarrow{\overline{a}\langle M\rangle} s \vdash P_1'$. By the induction hypothesis, we have $t \vdash C_1[\widetilde{N}] \xrightarrow{a\langle N\rangle} t \vdash Q_1'$, $(P_1', Q_1') \in (\mathcal{E}; r)^\circ$ and $(M, N) \in (\mathcal{E}; r)^\star$. From this we derive $t \vdash l[C_1[\widetilde{N}]] \xrightarrow{\overline{a}\langle N\rangle} l[Q_1']$ and $(l[P_1'], l[Q_1']) \in (\mathcal{E}; r)^\circ$ and we are done.
5. **Case** PASSIV: $C = l[C_1]$
   We have that $s \vdash P = s \vdash l[C_1[\widetilde{M}]] \xrightarrow{\overline{l}\langle \text{`}C_1[\widetilde{M}]\rangle} 0$. Immediately, we have $t \vdash Q = t \vdash l[C_1[\widetilde{N}]] \xrightarrow{\overline{l}\langle \text{`}C_1[\widetilde{N}]\rangle} 0$ with $(\text{`}C_1[\widetilde{M}], \text{`}C_1[\widetilde{N}]) \in (\mathcal{E}; r)^\star$ and $(0, 0) \in (\mathcal{E}; r)^\circ$.
6. **Case** REP: $C = !C_1$
   We have that $s \vdash P = s \vdash !C_1[\widetilde{M}] \xrightarrow{\overline{a}\langle M\rangle} s \vdash P'$, i.e. $s \vdash !C_1[\widetilde{M}] \mid C_1[\widetilde{M}] \xrightarrow{\overline{a}\langle M\rangle} s \vdash P'$. By the induction hypothesis, we have $t \vdash !C_1[\widetilde{N}] \mid C_1[\widetilde{N}] \xrightarrow{a\langle N\rangle} t \vdash Q'$, $(M, N) \in (\mathcal{E}; r)^\star$ and $(P', Q') \in (\mathcal{E}; r)^\circ$, hence $!C_1[\widetilde{N}] \xrightarrow{\overline{a}\langle N\rangle} Q'$ and we are done.

Proposition B.14. [Non-interference of names]

1. *If* $s \vdash P_0 \mid P_1 \xrightarrow{\alpha} s' \vdash P_0' \mid P_1$ *then for any* x*, we can assume that* $s, x \vdash P_0 \mid P_1 \xrightarrow{\alpha} s', x \vdash P_0' \mid P_1$ *using implicit* α*-conversion in* $s \vdash P_0 \mid P_1$*.*
2. *If* $s \vdash P_0 \stackrel{\alpha}{\Rightarrow} s' \vdash P_0'$ *then for any* P_1 *such that* $fn(P_1) \cap (s' \setminus s)$*, we have* $s \vdash P_0 \mid P_1 \stackrel{\alpha}{\Rightarrow} s' \vdash P_0' \mid P_1$*.*

Lemma B.15. [Input and output preserve environmental bisimulation up-to context]
Let $\mathcal{Y}$ *be an environmental bisimulation up-to context and* $\mathcal{X} = \{(\mathcal{E}, r, s, P, t, Q) \mid (s \vdash P)\ \mathcal{Y}^\star_{\mathcal{E};r}\ (t \vdash Q)\}$*. Then, for all* $(s \vdash P)\ \mathcal{X}_{\mathcal{E};r}\ (t \vdash Q)$*,*

1. *if* $s \vdash P \xrightarrow{a(M)} s \vdash P'$ *with* a *in* r*, then for all* $(M, N) \in (\mathcal{E}; r)^\star$ *there are* Q'*,* t' *such that* $t \vdash Q \stackrel{a(N)}{\Longrightarrow} t' \vdash Q'$ *and* $(s \vdash P')\ \mathcal{X}_{\mathcal{E};r}\ (t' \vdash Q')$*,*
2. *if* $s \vdash P \xrightarrow{\overline{a}\langle M\rangle} s \vdash P'$ *with* a *in* r*, then there are* Q'*,* N*,* t' *such that* $t \vdash Q \stackrel{\overline{a}\langle N\rangle}{\Longrightarrow} t' \vdash Q'$ *and* $(s \vdash P')\ \mathcal{X}_{(M,N)\oplus\mathcal{E};r}\ (t' \vdash Q')$*, and*
3. *the converse of the above two hold for* Q*'s transitions too.*

Proof. Suppose $(s \vdash P)\ \mathcal{Y}^\star_{\mathcal{E};r}\ (t \vdash Q)$, therefore for some P_0, P_1, Q_0, Q_1, $\mathcal{E}'$, r', we have $P \equiv P_0 \mid P_1$, $Q \equiv Q_0 \mid Q_1$, $r' \cap (s \cup t) = \emptyset$, $\mathcal{E} \subseteq (\mathcal{E}'; rr')^\star$, $(s, r' \vdash$

$P_0)\ \mathcal{Y}_{\mathcal{E}';rr'}\ (t, r' \vdash Q_0)$ and $(P_1, Q_1) \in (\mathcal{E}'; rr')^\circ$. We are going to analyse all the possible input/output transitions.

1. **Case**: Input
   There are two subcases for this transition:
   (a) **Subcase**: $s \vdash P_0 \mid P_1 \xrightarrow{a(M)} s \vdash P_0' \mid P_1$
   By $s \vdash P_0 \mid P_1 \xrightarrow{a(M)} s \vdash P_0' \mid P_1$, we know that we have $s \vdash P_0 \xrightarrow{a(M)} s \vdash P_0'$, and therefore $s, r' \vdash P_0 \xrightarrow{a(M)} s, r' \vdash P_0'$. By $\mathcal{E} \subseteq (\mathcal{E}'; rr')^\star$, we have $(\mathcal{E}; r)^\star \subseteq (\mathcal{E}'; rr')^\star$, and thus, by $(s, r' \vdash P_0)\ \mathcal{Y}_{\mathcal{E}';rr'}\ (t, r' \vdash Q_0)$ and $s, r' \vdash P_0 \xrightarrow{a(M)} s, r' \vdash P_0'$, we have (i) $t, r' \vdash Q_0 \stackrel{a(N)}{\Longrightarrow} t', r' \vdash Q_0'$ and (ii) $(s, r' \vdash P_0')\ \mathcal{Y}^\star_{\mathcal{E}';rr'}\ (t', r' \vdash Q_0')$. (i) tells us that $t \vdash Q_0 \mid Q_1 \stackrel{a(N)}{\Longrightarrow} t' \vdash Q_0' \mid Q_1 \equiv Q'$, and (ii) tells us, using the up-to techniques, that $(s \vdash P')\ \mathcal{Y}^\star_{\mathcal{E};r}\ (t' \vdash Q')$, hence $(s \vdash P')\ \mathcal{X}_{\mathcal{E};r}\ (t' \vdash Q')$.
   (b) **Subcase**: $s \vdash P_0 \mid P_1 \xrightarrow{a(M)} P_0 \mid P_1'$
   By $s \vdash P_0 \mid P_1 \xrightarrow{a(M)} s \vdash P_0 \mid P_1'$, we know that we have $s \vdash P_1 \xrightarrow{a(M)} s \vdash P_1'$, and therefore that we have $s, r' \vdash P_1 \xrightarrow{a(M)} s, r' \vdash P_1'$. By Lemma B.12, we have that (i) $t, r' \vdash Q_1 \xrightarrow{a(N)} t, r' \vdash Q_1'$ and (ii) $(P_1', Q_1') \in ((M, N) \oplus \mathcal{E}'; rr')^\circ$. Since $(M, N) \in (\mathcal{E}; r)^\star \subseteq (\mathcal{E}'; rr')^\star$, (ii) actually implies $(P_1', Q_1') \in (\mathcal{E}'; rr')^\circ$, and thus we have, for $Q' \equiv Q_0 \mid Q_1'$, $(s \vdash P')\ \mathcal{Y}^\star_{\mathcal{E};r}\ (t \vdash Q')$, that is, $(s \vdash P')\ \mathcal{X}_{\mathcal{E};r}\ (t \vdash Q')$. (i) implies $t \vdash Q_0 \mid Q_1 \xrightarrow{a(N)} t \vdash Q_0 \mid Q_1'$, and we are done.
2. **Case**: Output
   There are two cases for this transition:
   (a) **Subcase**: $s \vdash P_0 \mid P_1 \xrightarrow{\overline{a}\langle M\rangle} s \vdash P_0' \mid P_1$
   By $s \vdash P_0 \mid P_1 \xrightarrow{\overline{a}\langle M\rangle} s \vdash P_0' \mid P_1$, we know that we have $s \vdash P_0 \xrightarrow{\overline{a}\langle M\rangle} s \vdash P_0'$, and therefore $s, r' \vdash P_0 \xrightarrow{\overline{a}\langle M\rangle} s, r' \vdash P_0'$. By $(s, r' \vdash P_0)\, \mathcal{Y}_{\mathcal{E}';rr'}\, (t, r' \vdash Q_0)$ and $s, r' \vdash P_0 \xrightarrow{\overline{a}\langle M\rangle} s, r' \vdash P_0'$, we have (i) $t, r' \vdash Q_0 \stackrel{\overline{a}\langle N\rangle}{\Longrightarrow} t', r' \vdash Q_0'$ and (ii) $(s, r' \vdash P_0')\, \mathcal{Y}^\star_{(M,N)\oplus\mathcal{E}';rr'}\, (t', r' \vdash Q_0')$. (i) tells us that $t \vdash Q_0 \mid Q_1 \stackrel{\overline{a}\langle N\rangle}{\Longrightarrow} t' \vdash Q_0' \mid Q_1 \equiv Q'$, and (ii) tells us, using the up-to techniques and the fact that $(M, N) \oplus \mathcal{E} \subseteq ((M, N) \oplus \mathcal{E}'; rr')^\star$, that $(s \vdash P')\ \mathcal{Y}^\star_{(M,N)\oplus\mathcal{E};r}\ (t' \vdash Q')$, hence $(s \vdash P')\ \mathcal{X}_{(M,N)\oplus\mathcal{E};r}\ (t' \vdash Q')$.
   (b) **Subcase**: $s \vdash P_0 \mid P_1 \xrightarrow{\overline{a}\langle M\rangle} P_0 \mid P_1'$
   By $s \vdash P_0 \mid P_1 \xrightarrow{\overline{a}\langle M\rangle} s \vdash P_0 \mid P_1'$, we know that we have $s \vdash P_1 \xrightarrow{\overline{a}\langle M\rangle} s \vdash P_1'$, and therefore that we have $s, r' \vdash P_1 \xrightarrow{\overline{a}\langle M\rangle} s, r' \vdash P_1'$. By Lemma B.12, we have that (i) $t, r' \vdash Q_1 \xrightarrow{\overline{a}\langle N\rangle} t, r' \vdash Q_1'$ and (ii) $(P_1', Q_1') \in (\mathcal{E}'; rr')^\circ$ and $(M, N) \in (\mathcal{E}'; rr')^\star$. (ii) means that $(s \vdash P')\ \mathcal{Y}^\star_{(M,N)\oplus\mathcal{E};r}\ (t \vdash Q')$ for $Q' = Q_0 \mid Q_1'$, that is, $(s \vdash P')\ \mathcal{X}_{(M,N)\oplus\mathcal{E};r}\ (t \vdash Q')$, and (i) implies $t \vdash Q_0 \mid Q_1 \xrightarrow{\overline{a}\langle N\rangle} t \vdash Q_0 \mid Q_1'$, and we are done.

3. **Case**: The converse of the above two cases on Q's transitions.
   Similar to clauses 1 and 2.

Definition B.16. *For all processes A, B, we write $A < B$ and $B > A$ if there are C_p and R such that $A = C_p[R]$ and $B = C_p[run\text{`}R]$. We write $P_0 \leq P_n$ if $P_0 < \cdots < P_n$ for some $n \geq 0$, and $A \leq_n B$ if $A = P_0 < \cdots < P_m = B$ for some $m \leq n$. We naturally write $A \geq B$ whenever $B \leq A$, and extend $\leq$ and $\geq$'s definitions to terms and labels.*

We use the metavariables P^+ and P^- along with P when we mean that $P \leq P^+$ and that $P^- \leq P$. (The notations $(\cdot)^+$ and $(\cdot)^-$ therefore do not represent operators.) Similarly, we use the metavariables M^+ and M^- to represent run-expansions and run-erasures of term M.

Definition B.17. [*run*-transition]
We write $s \vdash P \xrightarrow{run} s \vdash P'$ when $s \vdash P \xrightarrow{\tau} s \vdash P'$ is derived using the rule Run. *Then, we write $s \vdash P_0 \xrightarrow{run^n} s \vdash P_n$ to mean that $s \vdash P_0 \xrightarrow{run} \cdots \xrightarrow{run} s \vdash P_n$, and $s \vdash P \xrightarrow{run}\!\!\!\!\!\Rightarrow \vdash Q$ when $s \vdash P \xrightarrow{run^n}\!\!\!\!\!\Rightarrow \vdash Q$ for some $n \geq 0$.*

Lemma B.18. *Let $\mathcal{X}_S = \{(\mathcal{E}, P, Q) \mid (P, Q) \in S,\ \mathcal{E} \subseteq S\}$. If $(\mathcal{E}, P, Q) \in \mathcal{X}_<$, then for any r, s, t*

- *if $s \vdash P \xrightarrow{run} s \vdash P'$ then*
  - *there is Q' such that $t \vdash Q \xrightarrow{run} t \vdash Q'$ and $(\mathcal{E}, P', Q') \in \mathcal{X}_<$, or*
  - *$t \vdash Q \xrightarrow{run} t \vdash P \xrightarrow{run} t \vdash P'$ and $(\mathcal{E}, P', P') \in \mathcal{X}_{\leq_1}$ with $P' = Q'$,*
- *if $s \vdash P \xrightarrow{\tau} s, x \vdash P'$ (not with the* Run *rule) then there is Q' such that $t \vdash Q \xrightarrow{run}\!\!\!\!\!\Rightarrow \xrightarrow{\tau} t, x \vdash Q'$, and $(\mathcal{E}, P', Q') \in \mathcal{X}_\leq$,*
- *if $s \vdash P \xrightarrow{\overline{a}\langle M\rangle} s \vdash P'$ then there are Q', $M \leq_1 N$ such that $t \vdash Q \xrightarrow{run}\!\!\!\!\!\Rightarrow \xrightarrow{\overline{a}\langle N\rangle} t \vdash Q'$, and $((M, N) \oplus \mathcal{E}, P', Q') \in \mathcal{X}_\leq$,*
- *if $s \vdash P \xrightarrow{a(M)} s \vdash P'$ then for all $(M, N) \in (\mathcal{E}; r)^\star$ there is Q' such that $t \vdash Q \xrightarrow{run}\!\!\!\!\!\Rightarrow \xrightarrow{a(N)} t \vdash Q'$, and $(\mathcal{E}, P', Q') \in \mathcal{X}_\leq$,*
- *the converse on Q's transitions (without run pre-steps).*

Similarly for $>$.

Proof. By induction on the derivation transition of $s \vdash P \xrightarrow{\alpha} s' \vdash P'$ (or $t \vdash Q \xrightarrow{\beta} t' \vdash Q$).

- Output
  - **Case** P's output
    There are two subcases: the context outputs, or some R_i.
    - **Subcase** $s \vdash P = C_p[R] \xrightarrow{\overline{a}\langle M\rangle} s \vdash P'$ by an output by the context.
      We have $s \vdash Q = C_p[run\text{`}R] \xrightarrow{\overline{a}\langle N\rangle} s \vdash Q'$. We are done as either $P' = C'_p[R] < C'_p[run\text{`}R] = Q'$ and $M = N$, or $P' = Q'$ and $M < N$, hence $((M, N) \oplus \mathcal{E}, P', Q') \in \mathcal{X}_\leq$.

* **Subcase** $s \vdash P = C_p[R_i] \xrightarrow{\overline{a}\langle M\rangle} s \vdash C_p[R_i'] = P'$.
  We have $s \vdash Q = C_p[run\text{`}R_i] \xrightarrow{run} s \vdash C_p[R_i] \xrightarrow{\overline{a}\langle N\rangle} s \vdash C_p[R_i']$. We have that $M = N$ and that $P' = Q'$ hence $((M,N)\oplus\mathcal{E}, P', Q') \in \mathcal{X}_{\leq}$.

- **Case** Q's output:
  There is only one subcase, as R cannot output for it is guarded.
  * **Subcase** $s \vdash Q = C_p[run\text{`}R] \xrightarrow{\overline{a}\langle N\rangle} s \vdash P'$ by an output from the context.
    Then $s \vdash P = s \vdash C_p[\text{`}R] \xrightarrow{\overline{a}\langle M\rangle} s \vdash Q'$. We are done as either $P' = Q'$ and $M < N$ or $P' < Q'$ and $M = n$, hence $((M,N)\oplus\mathcal{E}, P', Q') \in \mathcal{X}_{\leq}$.

– Reduction

- **Case** P's reduction:
  There are four subcases: the context reduces, sends R, receives from R, or R reduces.
  * **Subcase** $s \vdash P = C_p[R] \xrightarrow{\tau} s' \vdash P'$ by a reduction by the context.
    We can do a case analysis on how the transition is done.
    1. **Subsubcase** *run*-transition.
       We have in fact $s \vdash P = C_p[R] \xrightarrow{run} s \vdash C_p'[R]$ and also $s \vdash Q = C_p[run\text{`}R] \xrightarrow{run} s \vdash C_p'[run\text{`}R]$. Therefore, $(\mathcal{E}, P', Q') \in \mathcal{X}_{<} \subseteq \mathcal{X}_{\leq_1}$.
    2. **Subsubcase** alloc-transition.
       We have in fact $s \vdash P = C_p[R] \xrightarrow{\tau} s' \vdash C_p'[R]$ and also $s \vdash Q = C_p[run\text{`}R] \xrightarrow{\tau} s' \vdash C_p'[run\text{`}R]$. Therefore, $(\mathcal{E}, P', Q') \in \mathcal{X}_{<} \subseteq \mathcal{X}_{\leq}$.
    3. **Subsubcase** other τ-transition.
       We have in fact $s \vdash P = C_p[R] \xrightarrow{\tau} s \vdash C_p'[\widetilde{R'}]$ because it may send R and duplicate or discard it, or substitute some variable in it for an other process. Therefore, we have $s \vdash Q = C_p[run\text{`}R] \xrightarrow{\tau} s \vdash C_p'[\widetilde{run\text{`}R'}]$ with the same number n of copies since the same reaction can be done. We then have $P' < \cdots < Q'$ with n "$<$", hence $(\mathcal{E}, P', Q') \in \mathcal{X}_{\leq}$.
  * **Subcase** $s \vdash P = C_p[R] \xrightarrow{\tau} s \vdash C_p'[R'\{M/X\}]$ by a communication between the context and R.
    Then $s \vdash Q = C_p[run\text{`}R] \xrightarrow{run} s \vdash C_p[R] \xrightarrow{\tau} s \vdash C_p'[R'\{N/X\}]$. Since $M = N$ has to hold, we have $C_p'[R'\{M/X\}] = C_p'[R'\{N/X\}]$ hence $(\mathcal{E}, P', Q') \in \mathcal{X}_{\leq}$.
  * **Subcase** $s \vdash P = C_p[R] \xrightarrow{\tau} s \vdash C_p'[R', A]$ by a communication between the context and R.
    Then $s \vdash Q = C_p[run\text{`}R] \xrightarrow{run} s \vdash C_p[R] \xrightarrow{\tau} s \vdash C_p'[R', A]$. We have $C_p'[R', A] = C_p'[R', A]$, hence $(\mathcal{E}, P', Q') \in \mathcal{X}_{\leq}$.
  * **Subcase** $s \vdash P = C_p[R] \xrightarrow{\tau} s' \vdash C_p[R']$ by a reduction of some R.
    Then $s \vdash Q = C_p[run\text{`}R] \xrightarrow{run} s \vdash C_p[R] \xrightarrow{\tau} s' \vdash C_p[R']$. We have $C_p[R'] = C_p[R']$ hence $(\mathcal{E}, P', Q') \in \mathcal{X}_{\leq}$.

- **Case** Q's reduction:
  There are only two subcases, either the context reduces, or some *run* around R is consumed. All the other subcases would imply R, but it is not in a redex position.

* **Subcase** $s \vdash Q \xrightarrow{\tau} s' \vdash Q'$ by a reduction from the context. We can do a case analysis on how the transition was done.
  1. **Subsubcase** *run*-transition.
     We have in fact $s \vdash Q = C_p[run`R] \xrightarrow{run} s \vdash C'_p[run`R]$. Therefore $s \vdash P = C_p[R] \xrightarrow{run} s \vdash C'_p[R]$ and $(\mathcal{E}, C'_p[R], s, C'_p[run`R]) \in \mathcal{X}_<$.
  2. **Subsubcase** alloc-transition.
     Similarly.
  3. **Subsubcase** other τ-transition.
     We have in fact $s \vdash Q = C_p[run`R] \xrightarrow{\tau} s \vdash C'_p[\widetilde{run`R'}]$ since the transition may substitute a variable in R for a process. Therefore we have $s \vdash P = C_p[R'] \xrightarrow{\tau} s \vdash C'_p[\widetilde{R'}]$, and we then have $P' < \cdots < Q'$, hence $(\mathcal{E}, P', Q') \in \mathcal{X}_\leq$.
* **Subcase** $s \vdash Q = C_p[run`R] \xrightarrow{run} s \vdash C_p[R]$. We are done since $Q' = P$, hence $(\mathcal{E}, P, Q') \in \mathcal{X}_{\leq_1}$.

– Input:

- **Case** P's input:
  There are two subcases: the context inputs, or R does.
  * **Subcase** $s \vdash P = C_p[R] \xrightarrow{a(M)} s \vdash C'_p[R'\{M/X\}, \widetilde{M}]$ by an input by the context.
    Then, for all $(M, N) \in (<; s)^\star \subseteq \ \leq_1$, we have $s \vdash Q = C_p[run`R] \xrightarrow{a(N)} s \vdash C'_p[run`R'\{N/X\}, \widetilde{N}]$. We are done as $P' = s \vdash C'_p[R, R'\{M/X\}, \widetilde{M}] < \cdots < C'_p[run`R'\{N/X\}, \widetilde{N}] = Q'$ hence $(\mathcal{E}, P', Q') \in \mathcal{X}_\leq$.
  * **Subcase** $s \vdash P = C_p[R] \xrightarrow{a(M)} s \vdash C_p[R'\{M/X\}] = P'$.
    Then, for all $(M, N) \in (<; s)^\star \subseteq \ \leq_1$, we have $s \vdash Q = C_p[run`R] \xrightarrow{run} s \vdash C_p[R] \xrightarrow{a(N)} s \vdash C_p[R'\{N/X\}] = Q'$. Since $M \leq_1 N$, we have $C_p[R'\{M/X\}] < \cdots < C_p[R'\{N/X\}]$ hence $(\mathcal{E}, P', Q') \in \mathcal{X}_\leq$.
- **Case** Q's input:
  There is only one subcase, as R cannot input for it is guarded.
  * $s \vdash Q = C_p[run`R] \xrightarrow{a(N)} s \vdash C'_p[run`R'\{N/X\}, N]$ by an input from the context, and thus for all $(M, N) \in (<; s)^\star \subseteq \ \leq_1$, we have $s \vdash P = s \vdash C_p[R] \xrightarrow{a(M)} s \vdash C'_p[R'\{M/X\}, M]$. We are done as $C'_p[R'\{M/X\}, M] < \cdots < C'_p[run`R'\{N/X\}, N]$, hence $(\mathcal{E}, P', Q') \in \mathcal{X}_\leq$.

Corollary B.19. *If $P < Q$ and $s \vdash P \xRightarrow{run} s \vdash P'$ then $s \vdash Q \xRightarrow{run} s \vdash Q'$ and $P' = Q'$ or $P' < Q'$, and conversely.*

Proof. By induction on the number of *run*'s in $s \vdash P \xRightarrow{run^n} s \vdash P'$, using Lemma B.18.

Corollary B.20. *If $P \leq_m Q$ and $s \vdash P \xRightarrow{run} s \vdash P'$ then $s \vdash Q \xRightarrow{run} s \vdash Q'$ and $P' \leq_n Q'$, with $n \leq m$ and conversely.*

Proof. By induction on the number of $<$'s in $P \leq Q$, using Corollary B.19.

Lemma B.21. *Suppose* $P_0 \leq P_m$ *and* $s \vdash P_0 \stackrel{run}{\Longrightarrow} \stackrel{\alpha}{\rightarrow} s' \vdash P_0'$ *without using the* Run *or* React-{L,R} *rules for transition* $\stackrel{\alpha}{\rightarrow}$. *Then,* $s \vdash P_m \stackrel{run}{\Longrightarrow} \stackrel{\beta}{\rightarrow} s' \vdash P_m'$, $\alpha \leq \beta$ *and* $P_0' \leq P_m'$. *And conversely.*

Proof. By induction on the number of "$<$" in $\leq$.

Direct

- **Case** 0
  Trivial.
- **Case** $\neq 0$
  We have $s \vdash P_0 \stackrel{run}{\Longrightarrow} s \vdash P_0'' \stackrel{\alpha}{\rightarrow} s' \vdash P_0'$ and $P_0 \leq P_m$, so we can apply Corollary B.20 to have $s \vdash P_m \stackrel{run}{\Longrightarrow} s \vdash P_m''$ and $P_0'' \leq P_m''$, that is $P_0'' < P_1'' < \cdots < P_m''$ with at most as many "$<$". Then, $s \vdash P_0'' \stackrel{\alpha}{\rightarrow} s' \vdash P_0'$, so, by Lemma B.18, we have $s \vdash P_1'' \stackrel{run}{\Longrightarrow} \stackrel{\gamma}{\rightarrow} s' \vdash P_1'$ with $\alpha \leq \gamma$ and $P_0' \leq P_1'$. We can apply the induction hypothesis to $s \vdash P_1'' \stackrel{run}{\Longrightarrow} \stackrel{\gamma}{\rightarrow} s' \vdash P_1'$ and $P_1'' \leq P_m''$ to obtain that $s \vdash P_m'' \stackrel{run}{\Longrightarrow} \stackrel{\beta}{\rightarrow} s' \vdash P_m'$, $\alpha \leq \beta$ and $P_1' \leq P_m'$. Therefore, $P_0' \leq P_1' \leq P_m'$, $\alpha \leq \gamma \leq \beta$, and $s \vdash P_m \stackrel{run}{\Longrightarrow} s \vdash P_m'' \stackrel{run}{\Longrightarrow} \stackrel{\beta}{\rightarrow} s' \vdash P_m'$ as desired.

– Converse
  By Corollary B.20, we can get rid of the initial *run*'s and just consider the simpler hypothesis $P_0 \leq P_m$ and $s \vdash P_m \stackrel{\beta}{\rightarrow} s' \vdash P_m'$.
  - **Case** 0
    Trivial
  - **Case** $\neq 0$
    By Lemma B.18 and $P_0 < \cdots < P_{m-1} < P_m$ we have that $s \vdash P_{m-1} \stackrel{\gamma}{\rightarrow} s' \vdash P_{m-1}'$ with $P_{m-1}' \leq P_m'$ and $\gamma \leq \beta$. We just call the induction hypothesis on $P_0 \leq P_{m-1}$ and $s \vdash P_{m-1} \stackrel{\gamma}{\rightarrow} s' \vdash P_{m-1}'$ and we obtain as desired $s \vdash P_0 \stackrel{\alpha}{\rightarrow} s' \vdash P_0'$, with $\alpha \leq \gamma \leq \beta$. and $P_0' \leq P_{m-1}' \leq P_m'$.

Lemma B.22. *Suppose* $P_0 \leq P_m$ *and* $s \vdash P_0 \stackrel{run}{\Longrightarrow} \stackrel{\tau}{\rightarrow} s \vdash P_0'$ *using rule* React-{L,R} *for the* $\stackrel{\tau}{\rightarrow}$ *transition. Then,* $s \vdash P_m \stackrel{run}{\Longrightarrow} \stackrel{\tau}{\rightarrow} s \vdash P_m'$, *and* $P_0' \leq P_m'$. *And conversely*

Proof. Using Lemma B.21, considering that P_0 communicates a term N.

– We have $s \vdash P_0 \stackrel{run}{\Longrightarrow} \stackrel{\tau}{\rightarrow} s \vdash P_0'$, doing a reaction, that is, $s \vdash P_0 \stackrel{run}{\Longrightarrow} \xrightarrow{a(N)} s \vdash P_{0l}$ for some process N and channel a. So, by Lemma B.21, we have $s \vdash P_m \stackrel{run}{\Longrightarrow} s \vdash P_{ml} \xrightarrow{a(N^+)} \cdot$ with $P_0 \leq P_{ml}$. Now, as we also have $s \vdash P_0 \xrightarrow{\overline{a}\langle N \rangle} s \vdash P_{0r}$, by Lemma B.21, we have $s \vdash P_{ml} \stackrel{run}{\Longrightarrow} s \vdash P_{mr} \xrightarrow{\overline{a}\langle N^{+'} \rangle} \cdot$ with $P_0 \leq P_{mr}$. As it happens that P_{mr} can reduce through the communication of an expansion of N on channel a, we have $s \vdash P_{mr} \stackrel{\tau}{\rightarrow} s \vdash P_m'$ with $P_0' \leq P_m'$ and we are done.
– Converse
  Similarly.

Corollary B.23. *The sets $\mathcal{X}_{\leq}$ and $\mathcal{X}_{\geq}$ are both preserved by input, output and reduction.*

Proof. Consequence of Corollary B.20 and Lemmas B.21 and B.22.

Corollary B.24. *For any contexts C and its erasures C_1^- and C_2^-, for any processes $\widetilde{P}$ and $\widetilde{Q}$, if $s \vdash C_1^-[\widetilde{P}] \xrightarrow{\alpha} s, x \vdash C_1'^-[\widetilde{P}]$ then $t \vdash C_2^-[\widetilde{Q}] \stackrel{\beta}{\Rightarrow} t, x \vdash C_2'^-[\widetilde{Q}]$ with $C_1'^-$ and $C_2'^-$ erasures of C', α and β erasures of some γ, and possibly $x = \emptyset$.*

Proof. Consequence of Corollary B.23, more precisely of Lemma B.18 focusing on the cases where the context only does a transition.

Definition B.25. [Minimal transition of *run*-expanded processes]
Suppose that $A \leq B$, $s \vdash A \xrightarrow{\alpha} s' \vdash A'$, $s \vdash B \xrightarrow{run^n} \xrightarrow{\beta} s' \vdash B'$ with $\alpha \leq \beta$ and that $A' \leq B'$. We say that $s \vdash B \xrightarrow{run^n} \xrightarrow{\beta} s' \vdash B'$ is minimal with respect to $s \vdash A \xrightarrow{\alpha} s' \vdash A'$ if and only if for all $s \vdash B \xrightarrow{run^m} \xrightarrow{\gamma} s' \vdash B''$ with $A' \leq B''$ and $\alpha \leq \gamma$, we have $n \leq m$.

Lemma B.26. [Minimality and *run*-transition]
Suppose that $s \vdash B \xrightarrow{run} s \vdash B'' \xrightarrow{run^{n-1}} \xrightarrow{\beta} s' \vdash B'$ with $n > 0$ is minimal with respect to $s \vdash A \xrightarrow{\alpha} s' \vdash A'$. We have that $s \vdash B'' \xrightarrow{run^{n-1}} \xrightarrow{\beta} s' \vdash B'$ too is minimal with respect to $s \vdash A \xrightarrow{\alpha} s' \vdash A'$.

Proof. By *reductio ad absurdum*. Suppose that $s \vdash B'' \xrightarrow{run^{n-1}} \xrightarrow{\beta} s' \vdash B'$ is not minimal with respect to $s \vdash A \xrightarrow{\alpha} s' \vdash A'$. There must be a minimal transition $s \vdash B'' \xrightarrow{run^m} \xrightarrow{\gamma} s' \vdash B'''$ with $s \vdash A' \leq s \vdash B'''$, $\alpha \leq \gamma$, and $m < n - 1$. Then we have a derivation $s \vdash B \xrightarrow{run} s \vdash B'' \xrightarrow{run^m} \xrightarrow{\gamma} s' \vdash B'''$ of length $m + 1 < n$ with $s \vdash A' \leq s \vdash B'''$, which contradicts the assumption that $s \vdash B \xrightarrow{run} s \vdash B'' \xrightarrow{run^{n-1}} \xrightarrow{\beta} s' \vdash B'$ is minimal.

Lemma B.27. [Minimality and contexts]
For all $s \vdash Q \xrightarrow{run^n} \xrightarrow{\beta} s' \vdash Q'$ minimal with respect to $s \vdash P \xrightarrow{\alpha} s' \vdash P'$,

- *for all evaluation context C and its erasure C^-, $s \vdash C[Q] \xrightarrow{run^n} \xrightarrow{\beta} s' \vdash C[Q']$ is minimal with respect to $s \vdash C^-[P] \xrightarrow{\alpha} s' \vdash C^-[P']$,*
- *if $Q = Q_0 \mid Q_1$, $Q' = Q_0' \mid Q_1'$, and $P = P_0 \mid P_1$ with $P_0 \leq Q_0$, $P_1 \leq Q_1$, then for all l and m, $s \vdash l[Q_0] \mid m[Q_1] \xrightarrow{run^n} \xrightarrow{\beta} s' \vdash l[Q_0'] \mid m[Q_1']$ is minimal with respect to $s \vdash l[P_0] \mid m[P_1] \xrightarrow{\alpha} s' \vdash l[P_0'] \mid m[P_1']$.*

Proof. Immediate, as none of the above operations can reduce the number of *run*'s that have to be deleted, and as they all preserve membership to $\leq$.

Definition B.28. [*run*-erased context closure]
We define the run-erased context closure $(\mathcal{E};r)^-$ *of environment* $\mathcal{E}$ *with names* r *as* $\leq (\mathcal{E};r)^\star \geq$ *, that is* $\{(M,N) \mid M \leq A,\ N \leq B,\ (A,B) \in (\mathcal{E};r)^\star\}$*. Notice that* $(\mathcal{E};r)^-$ *may erase run's inside elements related by* $\mathcal{E}$ *too.*

We also write $(s \vdash P)\, \mathcal{Y}^-_{\mathcal{E};r}\, (t \vdash Q)$ *if* $(s \vdash P \leq)\, \mathcal{Y}^\star_{\leq\mathcal{E}\geq;r}\, (\geq t \vdash Q)$ *(which implies* $\mathcal{Y}^\star \subseteq \mathcal{Y}^-$*). In other words* $(s \vdash P)\mathcal{Y}^-_{\mathcal{E};r}(t \vdash Q)$ *if* $P \equiv P_0|P_1$*,* $Q \equiv Q_0|Q_1$*,* $(s, r' \vdash P_0 \leq)\, \mathcal{Y}_{\mathcal{E}';rr'}\, (\geq t, r' \vdash Q_0)$*,* $(\text{`}P_1, \text{`}Q_1) \in (\mathcal{E}';rr')^-$*,* $\mathcal{E} \subseteq (\mathcal{E}';rr')^-$*, and* $r' \cap (s \cup t) = \emptyset$*.*

Corollary B.29. [*run*-erasure preserves *run*-erased context closure of environmental bisimulation up-to context]
If $(s \vdash P)\, \mathcal{Y}^-_{\mathcal{E};r}\, (t \vdash Q)$*,* $P^- \leq P$*,* $Q^- \leq Q$ *and* $\mathcal{E}^- \leq \mathcal{E}$ *then* $(s \vdash P^-)\, \mathcal{Y}^-_{\mathcal{E}^-;r}\, (t \vdash Q^-)$*.*

Proof. From transitivity of $\leq$ and $\geq$ given by Definition B.16.

Lemma B.30. [Addition of fresh names preserves environmental bisimulation up-to context and its *run*-erased context closure]
Let $\mathcal{Y}$ *be an environmental bisimulation up-to context. If* $(s \vdash P)\, \mathcal{Y}^\star_{\mathcal{E};r}\, (t \vdash Q)$ *and* $l \notin s \cup t$*, then* $(s, l \vdash P)\, \mathcal{Y}^\star_{\mathcal{E};l\oplus r}\, (t, l \vdash Q)$*. Similarly, if* $(s \vdash P)\, \mathcal{Y}^-_{\mathcal{E};r}\, (t \vdash Q)$ *and* $l \notin s \cup t$*, then* $(s, l \vdash P)\, \mathcal{Y}^-_{\mathcal{E};l\oplus r}\, (t, l \vdash Q)$*.*

Proof. By simple set arithmetic and use of definitions.

- **Case** $\mathcal{Y}^\star$
  Given $P = P_0 \mid P_1$, $Q = Q_0 \mid Q_1$ such that $(s, x \vdash P_0)\, \mathcal{Y}_{\mathcal{E}';r}\, (t, x \vdash Q_0)$, $(P_1, Q_1) \in (\mathcal{E}';rx)^\circ$, $x \cap (s \cup t) = \emptyset$, and $\mathcal{E} \subseteq (\mathcal{E}';rx)^\star$, and assuming $l \neq x$ (otherwise it is immediate), it holds that
  - $(s, x \vdash P_0)\, \mathcal{Y}_{\mathcal{E}';l\oplus rx}\, (t, x \vdash Q_0)$ by clause 5 of environmental bisimulation up-to context,
  - $(P_1, Q_1) \in (\mathcal{E}';l\oplus r)^\circ \subseteq (\mathcal{E}';l\oplus rx)^\circ$,
  - $\mathcal{E} \subseteq (\mathcal{E}';rx)^\star \subseteq (\mathcal{E}';l\oplus rx)^\circ$,
  - $x \notin s \cup t$

  Therefore, $(s \vdash P)\, \mathcal{Y}^\star_{\mathcal{E};l\oplus r}\, (t \vdash Q)$ holds.
- **Case** $\mathcal{Y}^-$
  We have some $P^+ \geq P$, $Q^+ \geq Q$, $\mathcal{E}^+ \geq \mathcal{E}$ such that $(s \vdash P^+)\, \mathcal{Y}^\star_{\mathcal{E}^+;r}\, (t \vdash Q^+)$. Therefore, according to the above case, we have $(s \vdash P^+)\, \mathcal{Y}^\star_{\mathcal{E}^+;l\oplus r}\, (t \vdash Q^+)$, hence $(s \vdash P)\, \mathcal{Y}^-_{\mathcal{E};l\oplus r}\, (t \vdash Q)$ by Definition B.28.

Lemma B.31. [Spawning preserves context closure of environmental bisimulation up-to context]
Let $\mathcal{Y}$ *be an environmental bisimulation up-to context. For all* $(s \vdash P)\, \mathcal{Y}^\star_{\mathcal{E};r}\, (t \vdash Q)$*,* $l \in r$ *and* $(\text{`}P_2, \text{`}Q_2) \in \mathcal{E}$*, we have* $(s \vdash P \mid l[P_2])\, \mathcal{Y}^\star_{\mathcal{E};r}\, (t \vdash Q \mid l[Q_2])$*.*

Proof. We have $P \equiv P_0 \mid P_1$ and $Q \equiv Q_0 \mid Q_1$, with $(s, r' \vdash P_0)\, \mathcal{Y}_{\mathcal{E}';rr'}\, (t, r' \vdash Q_0)$, $(P_1, Q_1) \in (\mathcal{E}';rr')^\circ$, $\mathcal{E} \subseteq (\mathcal{E}';rr')^\star$, and $r' \notin s \cup t$. By $(\text{`}P_2, \text{`}Q_2) \in \mathcal{E}$, we have either $(\text{`}P_2, \text{`}Q_2) \in \mathcal{E}'$ or $(P_2, Q_2) \in (\mathcal{E}';rr')^\circ$. In the former case, it holds

that $(s, r' \vdash P_0 \mid l[P_2])\ \mathcal{Y}^{\star}_{\mathcal{E}';rr'}\ (t, r' \vdash Q_0 \mid l[Q_2])$ by clause 4 of environmental bisimulation up-to context, hence $(s \vdash P_0 \mid l[P_2] \mid P_1)\ \mathcal{Y}^{\star}_{\mathcal{E};r}\ (t \vdash Q_0 \mid l[Q_2] \mid Q_1)$ up-to environment, context and name creation. In the latter case, we immediately have $(P_1 \mid l[P_2], Q_1 \mid l[Q_2]) \in (\mathcal{E}'; rr')^{\circ}$, hence $(s \vdash P \mid l[P_2])\ \mathcal{Y}^{\star}_{\mathcal{E};r}\ (t \vdash Q \mid l[Q_2])$.

Lemma B.32. [*run*-transitions of $(\mathcal{E}; r)^{\circ}$]
Suppose that $(P_1, Q_1) = (C[\widetilde{M}], C[\widetilde{N}]) \in (\mathcal{E}; r)^{\circ}$ *and that* $s \vdash P_1 \xrightarrow{run} s \vdash P_1'$. *Then there is a* Q_1' *such that* $t \vdash Q_1 \xrightarrow{run} t \vdash Q_1'$ *and either* $(P_1', Q_1') = (C'[\widetilde{M}], C'[\widetilde{N}]) \in (\mathcal{E}; r)^{\circ}$ *or* $(`P_1', `Q_1') = (`C_p[run(\widetilde{M}'), A], `C_p[run(\widetilde{N}'), B]) \in (\mathcal{E}; r)^{-}$ *with* (A, B) *in redex position (i.e. not under a run, a* ν*, an* $a(\cdot)$ *or an* $\overline{a}\langle\cdot\rangle$*) and* $(\widetilde{M}', `A) = \widetilde{M}$, $(\widetilde{N}', `B) = \widetilde{N}$.

Proof. By induction on the transition derivation of $s \vdash P_1 \xrightarrow{run} s \vdash P1'$. The only case of interest is the RUN one, developed below. The others (PAR-L, PAR-R,REP and TRANSP) are straightforward.

1. **Case** RUN: $C = run(C_1)$
   There are two subcases
   (a) $C_1 = `C_1$
   We have $s \vdash P_1 = s \vdash run(`C_1[\widetilde{M}]) \xrightarrow{run} s \vdash C_1[\widetilde{M}]$ and $t \vdash Q_1 = t \vdash run(`C_1[\widetilde{N}]) \xrightarrow{run} t \vdash C_1[\widetilde{N}]$ with $(C_1[\widetilde{M}], C_2[\widetilde{N}]) \in (\mathcal{E}; r)^{\circ}$.
   (b) $C_1 = [\cdot]$
   We have $s \vdash P_1 = run(`A) \xrightarrow{run} s \vdash A$, $t \vdash Q_1 \xrightarrow{run} t \vdash B$ with $(`A, `B) \in \mathcal{E} \subseteq (\mathcal{E}; r)^{-}$ (we can assume that $(A, B) \notin (\mathcal{E}; r)^{\circ}$, otherwise we could have handled this situation in the above subcase) and obviously (A, B) in redex position.

Lemma B.33. [Create-transitions of $(\mathcal{E}; r)^{\circ}$]
Suppose that $(P_1, Q_1) = (C[\widetilde{M}], C[\widetilde{N}]) \in (\mathcal{E}; r)^{\circ}$ *and that* $s \vdash P_1 \xrightarrow{\tau} s, a \vdash P_1'$ *by the* CREATE *rule. Then there is a* Q_1' *such that* $t \vdash Q_1 \xrightarrow{\tau} t, a \vdash Q_1'$ *and* $(P_1', Q_1') = (C'[\widetilde{M}], C'[\widetilde{N}]) \in (\mathcal{E}; ra)^{\circ}$.

Proof. By induction on the transition derivation of $s \vdash P_1 \xrightarrow{\tau} s, a \vdash P_1'$. The only case of interest is the CREATE one, developed below. The others (PAR-L, PAR-R, REP and TRANSP) are straightforward.

1. **Case** CREATE: $C = \nu a.C_1$
   We have $s \vdash P_1 = s \vdash \nu a.C_1[\widetilde{M}] \xrightarrow{\tau} s, a \vdash C_1[\widetilde{M}]$ and $t \vdash Q_1 = t \vdash \nu a.C_1[\widetilde{N}] \xrightarrow{\tau} t, a \vdash C_1[\widetilde{N}]$ with $(C_1[\widetilde{M}], C_1[\widetilde{N}]) \in (\mathcal{E}; ra)^{\circ}$.

Lemma B.34. [Non-*run* non-alloc τ-transitions of $(\mathcal{E}; r)^{\circ}$]
Suppose that $(P_1, Q_1) \in (\mathcal{E}; r)^{\circ}$ *and that* $s \vdash P_1 \xrightarrow{\tau} s \vdash P_1'$ *is not derived with* RUN *nor* CREATE. *Then there is a* Q_1' *such that* $t \vdash Q_1 \xrightarrow{\tau} t \vdash Q_1'$ *and* $(P_1', Q_1') = (\mathcal{E}; r)^{\circ}$.

Proof. By induction on the transition derivation of $s \vdash P_1 \xrightarrow{\tau} s \vdash P1'$. One case of interest is the REACT-L one, developed below. The others (REACT-R, PAR-L, PAR-R, REP and TRANSP) are similar or straightforward.

– **Case** REACT-L: $C = C1 \mid C2$

We have $s \vdash C_1[\widetilde{M}] \mid C_2[\widetilde{M}] \xrightarrow{\tau} s \vdash P_1' \mid P_2'$ with $s \vdash C_1[\widetilde{M}] \xrightarrow{\overline{a}\langle M\rangle} s \vdash P_1'$ and $s \vdash C_2[\widetilde{M}] \xrightarrow{a(M)} s \vdash P_2'$ So, by Lemmas B.12 and B.13 we have $t \vdash C_1[\widetilde{N}] \xrightarrow{\overline{a}\langle N\rangle} t \vdash Q_1'$ with $(P_1', Q_1') \in (\mathcal{E}; r)^\circ$ and $(M, N) \in (\mathcal{E}; r)^\star$, and $t \vdash C_2[\widetilde{N}] \xrightarrow{a(N)} t \vdash Q_2'$ with $(P_2', Q_2') \in ((M, N) \oplus \mathcal{E}; r)^\circ = (\mathcal{E}; r)^\circ$. Therefore, $t \vdash C_1[\widetilde{N}] \mid C_2[\widetilde{N}] \xrightarrow{\tau} t \vdash Q_1' \mid Q_2'$ and $(P_1' \mid P_2', Q_1' \mid Q_2') \in (\mathcal{E}; r)^\circ$.

Lemma B.35. [Reduction and environmental bisimulation up-to context] *Let $\mathcal{Y}$ be an environmental bisimulation up-to context. If $(s \vdash P)\, \mathcal{Y}^\star_{\mathcal{E};r}\, (t \vdash Q)$ and $s \vdash P \to s' \vdash P'$ then there is a Q' such that $t \vdash Q \Rightarrow t' \vdash Q'$ and $(s' \vdash P')\, \mathcal{Y}^-_{\mathcal{E};r}\, (t' \vdash Q')$.*

Proof. Suppose $(s \vdash P)\, \mathcal{Y}^\star_{\mathcal{E};r}\, (t \vdash Q)$, therefore for some P_0, P_1, Q_0, Q_1, $\mathcal{E}'$, r', we have $P \equiv P_0 \mid P_1$, $Q \equiv Q_0 \mid Q_1$, $r' \cap (s \cup t) = \emptyset$, $\mathcal{E} \subseteq (\mathcal{E}'; rr')^\star$, $(s, r' \vdash P_0)\, \mathcal{Y}_{\mathcal{E}';rr'}\, (t, r' \vdash Q_0)$ and $(P_1, Q_1) \in (\mathcal{E}'; rr')^\circ$.

We are going to analyse all the possible reduction transitions. We recall that $\mathcal{Y}^\star \subseteq \mathcal{Y}^-$.

1. **Case**: $s \vdash P \xrightarrow{\tau} s' \vdash P'$. We have four cases for the transitions of $P_0 \mid P_1$:
   (a) **Subcase** $s, r' \vdash P_0 \xrightarrow{\tau} s', r' \vdash P_0'$
   By $(s, r' \vdash P_0)\, \mathcal{Y}_{\mathcal{E}';rr'}\, (t, r' \vdash Q_0)$, we have that $t, r' \vdash Q_0 \Rightarrow t', r' \vdash Q_0'$ and $(s', r' \vdash P_0')\, \mathcal{Y}^\star_{\mathcal{E}';rr'}\, (t', r' \vdash Q_0')$. Therefore, by $t, r' \vdash Q_0 \mid Q_1 \Rightarrow t', r' \vdash Q_0' \mid Q_1$ we have $t \vdash Q_0 \mid Q_1 \Rightarrow t' \vdash Q_0' \mid Q_1$ since the created names can be guaranteed not in $fn(Q_1)$, and by up-to context and environment and name creation, we have $(s' \vdash P_0' \mid P_1)\, \mathcal{Y}^\star_{\mathcal{E};r}\, (t' \vdash Q_0' \mid Q_1)$
   (b) **Subcase** $s, r' \vdash P_1 \xrightarrow{\tau} s', r' \vdash P_1'$
   There are several cases, depending on the last derivation rule used.
      i. Non-run non-alloc transition, $s' = s$
      By Lemma B.34, we have $t, r' \vdash Q_1 \xrightarrow{\tau} t, r' \vdash Q_1'$ and $(P_1', Q_1') \in (\mathcal{E}'; rr')^\circ$. Therefore, $t \vdash Q_1 \xrightarrow{\tau} t \vdash Q_1'$, hence $t \vdash Q_0 \mid Q_1 \xrightarrow{\tau} t \vdash Q_0 \mid Q_1'$. Finally, $(s' \vdash P')\, \mathcal{Y}^\star_{\mathcal{E};r}\, (t' \vdash Q')$ with $t' = t$ and we are done.
      ii. Create transition, $s' = s, a$
      By Lemma B.33, we have $t, r' \vdash Q_1 \xrightarrow{\tau} t, r', a \vdash Q_1'$, hence $t \vdash Q_0 \mid Q_1 \xrightarrow{\tau} t, a \vdash Q_0 \mid Q_1'$, and $(P_1', Q_1') \in (\mathcal{E}; rr'a)^\circ$. By freshness of a, we can use clause 5 of environmental bisimulation up-to and have $(s, r', a \vdash P_0)\, \mathcal{Y}_{\mathcal{E}';rr'a}\, (t, r', a \vdash Q_0)$ as well as $(r', a) \cap (s \cup t) = \emptyset$. Finally, $\mathcal{E} \subseteq (\mathcal{E}'; rr')^\star \subseteq (\mathcal{E}'; rr'a)^\star$, giving $(s' \vdash P')\, \mathcal{Y}^\star_{\mathcal{E};r}\, (t' \vdash Q')$ and we are done.
      iii. *run* transition, $s' = s$
      By Lemma B.32, we have $t, r' \vdash Q_1 \xrightarrow{run} t, r' \vdash Q_1'$ (hence $t \vdash Q_0 \mid Q_1 \xrightarrow{run} t \vdash Q_0 \mid Q_1'$) and either $(P_1', Q_1') = (C'[\widetilde{M}], C'[\widetilde{N}]) \in (\mathcal{E}'; rr')^\circ$ or $(`P_1', `Q_1') = (`C_p[run(\widetilde{M'}), A], `C_p[run(\widetilde{N'}), B]) \in (\mathcal{E}'; rr')^-$ with $(`A, `B) \in \mathcal{E}'$. Therefore, $(s \vdash P')\, \mathcal{Y}^-_{\mathcal{E};r}\, (t \vdash Q')$ and we are done.

(c) **Subcase** $s, r' \vdash P_0 \xrightarrow{\overline{a}\langle M\rangle} s, r' \vdash P_0' \quad s, r' \vdash P_1 \xrightarrow{a(M)} s, r' \vdash P_1'$
By $(s, r' \vdash P_0)\ \mathcal{Y}_{\mathcal{E}';rr'}\ (t, r' \vdash Q_0)$ and clause 4 of environmental bisimulation up-to context, we have $t, r' \vdash Q_0 \stackrel{\overline{a}\langle N\rangle}{\Longrightarrow} t', r' \vdash Q_0'$ and also $(s, r' \vdash P_0')\mathcal{Y}^{\star}_{(M,N)\oplus\mathcal{E}';rr'}(t', r' \vdash Q_0')$. Also, since $s, r' \vdash P_1 \xrightarrow{a(M)} s, r' \vdash P_1'$ we have by Lemma B.12 that $t, r' \vdash Q_1 \xrightarrow{a(N)} t, r' \vdash Q_1'$ and $(P_1', Q_1') \in (\mathcal{E}' \cup \{(M, N)\}; rr')^{\circ}$.
Decomposing the transitions we know that, for some possibly empty set y of names, $t, r' \vdash Q_0 \stackrel{\tau}{\Rightarrow} t, r', y \vdash Q_0'' \xrightarrow{a(N)} t, r', y \vdash Q_0''' \stackrel{\tau}{\Rightarrow} t', r' \vdash Q_0'$. Also, by $t, r' \vdash Q_1 \xrightarrow{a(N)} t, r' \vdash Q_1'$ we have by $t, r', y \vdash Q_1 \xrightarrow{a(N)} t, r', y \vdash Q_1'$. Thus, we have $t, r' \vdash Q_0 \mid Q_1 \stackrel{\tau}{\Rightarrow} t, r', y \vdash Q_0'' \mid Q_1$ by Par-L, $t, r', y \vdash Q_0'' \mid Q_1 \xrightarrow{\tau} t, r', yQ_0''' \mid Q_1'$ by React-R, and finally $t, r', y \vdash Q_0''' \mid Q_1' \stackrel{\tau}{\Rightarrow} t', r' \vdash Q_0' \mid Q_1'$ by Par-L. We can then therefore derive $t \vdash Q_0 \mid Q_1 \Rightarrow t' \vdash Q_0' \mid Q_1'$.
By $(P_1', Q_1') \in (\mathcal{E}' \cup \{(M, N)\}; rr')^{\circ}$, we also easily have $(P_1', Q_1') \in (\mathcal{E}' \cup \{(M, N)\}; rr')^{\circ}$, and we can derive up-to context from $(s, r' \vdash P_0')\ \mathcal{Y}^{\star}_{(M,N)\oplus\mathcal{E}';rr'}\ (t', r' \vdash Q_0')$ that $(s \vdash P_0' \mid P_1')\ \mathcal{Y}^{\star}_{\mathcal{E};r}\ (t' \vdash Q_0' \mid Q_1')$.

(d) **Subcase** $s \vdash P_0 \xrightarrow{a(M)} s \vdash P_0' \quad s \vdash P_1 \xrightarrow{\overline{a}\langle M\rangle} s \vdash P_1'$
By $s \vdash P_1 \xrightarrow{\overline{a}\langle M\rangle} s \vdash P_1'$ we have $s, r' \vdash P_1 \xrightarrow{\overline{a}\langle M\rangle} s, r' \vdash P_1'$, and then by Lemma B.13, we have that $t, r' \vdash Q_1 \xrightarrow{\overline{a}\langle N\rangle} t, r' \vdash Q_1'$ and $(M, N) \in (\mathcal{E}'; rr')^{\star}$ as well as $(P_1', Q_1') \in (\mathcal{E}'; rr')^{\circ}$. By clause 2 of environmental bisimulation up-to context and the input of P_0, we have $t, r' \vdash Q_0 \stackrel{a(N)}{\Longrightarrow} t', r' \vdash Q_0'$ and $(P_0')\ \mathcal{Y}^{\star}_{\mathcal{E}';rr'}\ (Q_0')$.
Again, we can compose the transitions and obtain $t, r' \vdash Q_0 \mid Q_1 \stackrel{\tau}{\Rightarrow} t', r' \vdash Q_0' \mid Q_1'$ as expected.
By $(P_1', Q_1') \in (\mathcal{E}'; rr')^{\circ}$, we also have $(P_1', Q_1') \in (\mathcal{E}'; rr')^{\circ}$, and we can then derive up-to context from $(s, r' \vdash P_0')\ \mathcal{Y}^{\star}_{\mathcal{E}';rr'}\ (t', r' \vdash Q_0')$ that $(s \vdash P_0' \mid P_1')\ \mathcal{Y}^{\star}_{\mathcal{E};r}\ (t' \vdash Q_0' \mid Q_1')$.

2. **Case**: $t \vdash Q$ reduces.
Conversely.

Lemma B.36. [*run*-expanded output with spawning]
Suppose that $(s \vdash P_0 | l[P_1])\mathcal{Y}^{\star}_{\mathcal{E};r}(t \vdash Q_0 | l[Q_1])$ *for an environmental bisimulation up-to context* $\mathcal{Y}$ *with* $l \in r$ *and that* $s \vdash P_1 \stackrel{run^n}{\Longrightarrow} \xrightarrow{\overline{a}\langle M\rangle} s \vdash P_1'$ *is minimal with respect to* $s \vdash P_1^{-} \xrightarrow{\overline{a}\langle M^{-}\rangle} s \vdash P_1'^{-}$. *Then* $t \vdash Q_0 \mid l[Q_1] \stackrel{\overline{a}\langle N\rangle}{\Longrightarrow} t' \vdash Q_0' \mid l[Q_1']$, *and* $(s \vdash P_0)\ \mathcal{Y}^{-}_{(M,N)\oplus(`P_1', `Q_1')\oplus\mathcal{E};r}\ (t' \vdash Q_0')$

Proof. By induction on n.

– **Case** $n = 0$
Immediate by Lemma B.15 used twice (once for the output of M and N, and then once more for the passivation of P_1' and Q_1') and by the fact that $\mathcal{Y}^{\star} \subseteq \mathcal{Y}^{-}$.

– **Case** $n > 0$
By Lemma B.35 and Lemma B.26, we have two possible subcases preserving minimality after the first *run*-transition of $s \vdash P_0 \mid l[P_1] \xRightarrow{run^n} \xrightarrow{\overline{a}\langle M\rangle} s \vdash P_0 \mid l[P_1]$, namely $s \vdash P_0 \mid l[P_1] \xrightarrow{run} s \vdash P_0 \mid l[P_1'']$.

- **Subcase** "still in $\mathcal{Y}^\star$"
We have $t \vdash Q_0 \mid l[Q_1] \stackrel{\tau}{\Rightarrow} t'' \vdash Q_0'' \mid l[Q_1'']$ and $(s \vdash P_0 \mid l[P_1'']) \, \mathcal{Y}^\star_{\mathcal{E};r} \, (t'' \vdash Q_0'' \mid l[Q_1''])$. As $s \vdash P_0 \mid l[P_1''] \xRightarrow{run^{n-1}} \xrightarrow{\overline{a}\langle M\rangle} s \vdash P_0 \mid l[P_1']$ is still minimal with respect to $s \vdash P_0^- \mid l[P_1^-] \xrightarrow{\overline{a}\langle M^-\rangle} s \vdash P_0^- \mid l[P_1'^-]$, we can apply the induction hypothesis and get the desired results.
- **Subcase** "in $\mathcal{Y}^- \setminus \mathcal{Y}^\star$"
We have $t \vdash Q_0 \mid l[Q_1] \xrightarrow{run} t \vdash Q_0 \mid l[Q_1'']$ and $(s \vdash P_0 \mid l[P_1'']) \, \mathcal{Y}^-_{\mathcal{E};r} \, (t \vdash Q_0'' \mid l[Q_1''])$, with $(P_1'', Q_1'') = (C_p[run(\widetilde{M}), A], C_p[run(\widetilde{N}), B])$ with (A, B) in redex position such that $(`A, `B) \in \mathcal{E}'$ and $(A, B) \notin (\mathcal{E}'; rr')^\circ$ for some $\mathcal{E}'$, r' such that $\mathcal{E} \subseteq (\mathcal{E}'; rr')^\star$, $P_0 \mid l[P_1] \equiv P_A \mid P_B$, $Q_0 \mid l[Q_1] \equiv Q_A \mid Q_B$, $(s, r' \vdash P_A) \, \mathcal{Y}_{\mathcal{E}';rr'} \, (t, r' \vdash Q_A)$, $(P_B, Q_B) \in (\mathcal{E}'; rr')^\circ$, $r' \cap (s \cup t) = \emptyset$.

  By $s \vdash P_0^- \mid l[P_1^-] \xrightarrow{\overline{a}\langle M^-\rangle} s \vdash P_0^- \mid l[P_1'^-]$, $P_1 = C_p[run(\widetilde{M}), run `A]$, $P_1'' = C_p[run(\widetilde{M}), A]$ and $P_1^- \leq P_1'$, we know that there is a *run*-erasure $A^- \leq A$ such that A^- is in redex position in P_1^- and that $s, r' \vdash A \xrightarrow{\overline{a}\langle M\rangle} s, r' \vdash A'$ is minimal with respect to $s, r' \vdash A^- \xrightarrow{\overline{a}\langle M^-\rangle} s, r' \vdash A'^-$. Using Lemma B.30 (to add a fresh name m), and clause 4 of environmental bisimulation up-to context, as well as derived $s, r', m \vdash A^- \xrightarrow{\overline{a}\langle M^-\rangle} s, r', m \vdash A'^-$, we can apply the induction hypothesis to $(s, r', m \vdash P_A \mid m[A]) \, \mathcal{Y}^\star_{\mathcal{E}';rr'm} \, (t, r', m \vdash Q_B \mid m[B])$. We obtain that $t, r', m \vdash Q_B \mid m[B] \xRightarrow{\overline{a}\langle N\rangle} t', r', m \vdash Q_B' \mid m[B']$ and that also $(s, r', m \vdash P_A) \mathcal{Y}^-_{(M,N)\oplus(`A',`B')\oplus\mathcal{E}';rr'm} (t', r', m \vdash Q_B')$. From the former, we can derive that $t'' \vdash Q_0 | l[Q_1''] \xRightarrow{\overline{a}\langle N\rangle} t' \vdash Q_0' | l[Q_1']$, (with $(P_1', Q_1') = (C_p[run(\widetilde{N}), A'], C_p[run(\widetilde{N}), B'])$) and from the latter that $(s, r', m \vdash P_0) \mathcal{Y}^-_{(M,N)\oplus(`A',`B')\oplus\mathcal{E}';rr'm} (t', r', m \vdash Q_0')$ up-to context, $(s, r', m \vdash P_0) \, \mathcal{Y}^-_{(M,N)\oplus(`P_1',`Q_1')\oplus\mathcal{E};rr'm} (t', r', m \vdash Q_0')$ up to environment, and $(s \vdash P_0) \mathcal{Y}^-_{(M,N)\oplus(`P_1',`Q_1')\oplus\mathcal{E};r} (t' \vdash Q_0')$ up-to name creation.

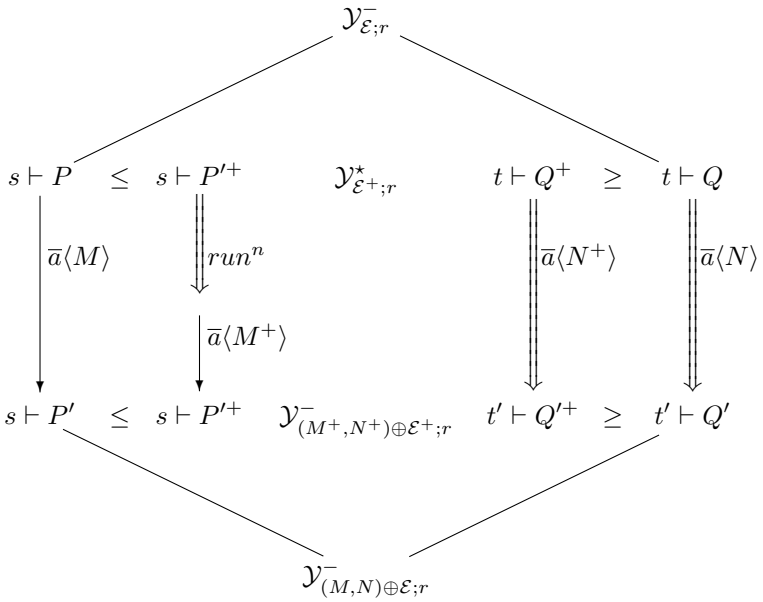
Corollary B.37. [*run*-expanded output]
Suppose that $(s \vdash P_0 \mid P_1) \, \mathcal{Y}^\star_{\mathcal{E};r} \, (t \vdash Q_0 \mid Q_1)$ *for an environmental bisimulation up-to context* $\mathcal{Y}$ *with* $(s, x \vdash P_0) \, \mathcal{Y}_{\mathcal{E}';rx} \, (t, x \vdash Q_0)$, $(P_1, Q_1) \in (\mathcal{E}'; rx)^\circ$, $x \cap (s \cup t) = \emptyset$, *and that* $s \vdash P_0 \mid P_1 \xRightarrow{run^n} \xrightarrow{\overline{a}\langle M\rangle} s \vdash P_0' \mid P_1'$ *is minimal with respect to* $s \vdash P_0^- \mid P_1^- \xrightarrow{\overline{a}\langle M^-\rangle} s \vdash P_0'^- \mid P_1'^-$. *Then* $t \vdash Q_0 \mid Q_1 \xRightarrow{\overline{a}\langle N\rangle} t' \vdash Q_0' \mid Q_1'$, *and* $(s \vdash P_0' \mid P_1') \, \mathcal{Y}^-_{(M,N)\oplus\mathcal{E};r} \, (t' \vdash Q_0' \mid Q_1')$.

Proof. By induction on n.

– **Case** $n = 0$
As in the above Lemma B.36, immediate by Lemma B.15 and the fact that $\mathcal{Y}^\star \subseteq \mathcal{Y}^-$.
– **Case** $n > 0$
By Lemma B.35 and Lemma B.26, we have two possible subcases preserving minimality after the first *run*-transition of $s \vdash P_0 \mid P_1 \xRightarrow{run^n} \xrightarrow{\overline{a}\langle M\rangle} s \vdash P_0' \mid P_1'$.
- **Subcase** "still in $\mathcal{Y}^\star$"
We have $s, r' \vdash P_0 \xrightarrow{run} s, r' \vdash P_0''$, hence $t \vdash Q_0 \stackrel{\tau}{\Rightarrow} t'' \vdash Q_0''$ and $(s \vdash P_0'')\ \mathcal{Y}^\star_{\mathcal{E}';rr'}\ (t'' \vdash Q_0'')$, and also that $s, r' \vdash P_0'' \xRightarrow{run^{n-1}} \xrightarrow{\overline{a}\langle M\rangle} s \vdash P_0'$ is minimal with respect to $s \vdash P_0^- \xrightarrow{\overline{a}\langle M^-\rangle} s \vdash P_0'^-$. Thus, we can apply the induction hypothesis and get (i) $t'' \vdash Q_0'' \xRightarrow{\overline{a}\langle N\rangle} t' \vdash Q_0'$ as well as (ii) $(s, r' \vdash P_0')\ \mathcal{Y}^-_{(M,N)\oplus\mathcal{E}';rr'}\ (t', r' \vdash Q_0')$. Therefore, by (i) we have $t \vdash Q_0 \mid Q_1 \xRightarrow{\overline{a}\langle N\rangle} t' \vdash Q_0' \mid Q_1$. and by (ii) we have $(s \vdash P_0' \mid P_1)\ \mathcal{Y}^-_{(M,N)\oplus\mathcal{E};r}$ $(t' \vdash Q_0' \mid Q_1)$ up-to environment and name creation for using $\mathcal{E}$ and removing r', and context for spawning P_1 and Q_1.
- **Subcase** "in $\mathcal{Y}^- \setminus \mathcal{Y}^\star$"
We have $s, r' \vdash P_1 \xrightarrow{run} s, r' \vdash P_1'$. Using Lemma B.30 to add a fresh name l and the fact that $(P_1, Q_1) \in (\mathcal{E}'; rr')^\circ$, we have $(s, r', l \vdash P_0 \mid l[P_1])\ \mathcal{Y}^\star_{\mathcal{E};r}\ (t, r', l \vdash Q_0 \mid l[Q_1])$. As $s, r', l \vdash P_0 \mid l[P_1] \xRightarrow{run^n} \xrightarrow{\overline{a}\langle M\rangle} s, r', l \vdash P_0 \mid l[P_1']$ is minimal with respect to $s, r', l \vdash P_0^- \mid l[P_1^-] \xrightarrow{\overline{a}\langle M^-\rangle} s, r', l \vdash P_0^- \mid l[P_1'^-]$, we can use Lemma B.36 and have $t, r', l \vdash Q_0 \mid l[Q_1] \xRightarrow{\overline{a}\langle N\rangle} t', r', l \vdash Q_0' \mid l[Q_1']$, hence $t \vdash Q_0 \mid Q_1 \xRightarrow{\overline{a}\langle N\rangle} t' \vdash Q_0' \mid Q_1'$ and also $(s, r', l \vdash P_0)\mathcal{Y}^-_{(M,N)\oplus(`P_1',`Q_1')\oplus\mathcal{E};rl}(t', r', l \vdash Q_0')$, hence $(s \vdash P_0 | P_1')\mathcal{Y}^-_{(M,N)\oplus\mathcal{E};r}(t' \vdash Q_0' \mid Q_1')$ up-to context, environment and name creation.

Corollary B.38. [Output preserves *run*-erased environmental bisimulation up-to context]
For any environmental bisimulation up-to context $\mathcal{Y}$, if $(s \vdash P)\ \mathcal{Y}^-_{\mathcal{E};r}\ (t \vdash Q)$ and $s \vdash P \xrightarrow{\overline{a}\langle M\rangle} s \vdash P'$ with $a \in r$, then there are Q', t' such that $t \vdash Q \xRightarrow{\overline{a}\langle N\rangle} t' \vdash Q'$ and $(s \vdash P')\ \mathcal{Y}^-_{(M,N)\oplus\mathcal{E};r}\ (t' \vdash Q')$. The converse on Q's transition holds too.

Proof. By $\mathcal{Y}^-$'s definition, we know there are P^+, Q^+ and $\mathcal{E}^+$ such that $(s \vdash P^+)\mathcal{Y}^\star_{\mathcal{E}^+;r}(t \vdash Q^+)$. Since $s \vdash P \xrightarrow{\overline{a}\langle M\rangle} s \vdash P'$, there is a minimal output transition $s \vdash P^+ \xRightarrow{run^n} \xrightarrow{\overline{a}\langle M^+\rangle} s \vdash P'^+$. By Lemma B.37, we have $t \vdash Q^+ \xRightarrow{\overline{a}\langle N^+\rangle} t' \vdash Q'^+$ and $(s \vdash P'^+)\ \mathcal{Y}^-_{(M^+,N^+)\oplus\mathcal{E}^+;r}\ (t' \vdash Q'^+)$ which implies by Corollary B.23 that $t \vdash Q$ can also weakly do an output transition $t \vdash Q \xRightarrow{\overline{a}\langle N\rangle} t' \vdash Q'$, such that $Q' \leq Q'^+$ and $N \leq N^+$. By Corollary B.29, as $P' \leq P'^+$, $Q' \leq Q'^+$ and $(M, N)\oplus\mathcal{E} \leq (M^+, N^+)\oplus\mathcal{E}^+$, we have $(s \vdash P')\ \mathcal{Y}^-_{(M,N)\oplus\mathcal{E};r}\ (t' \vdash Q')$ as desired. Visually, the following diagram holds.



The converse on Q's transitions is shown similarly.

Lemma B.39. [*run*-expanded input]
Suppose that $(s \vdash P)\mathcal{Y}^{\star}_{\mathcal{E};r}(t \vdash Q)$ *for an environmental bisimulation up-to context* $\mathcal{Y}$ *and that* $s \vdash P \stackrel{run^{n}}{\Longrightarrow} \xrightarrow{a(M)} s \vdash P'$ *is minimal with respect to* $s \vdash P^{-} \xrightarrow{a(M^{-})} s \vdash P'^{-}$. *Then for all* N *such that* $(M,N) \in (\mathcal{E};r)^{\star}$, $t \vdash Q \stackrel{a(N)}{\Longrightarrow} t' \vdash Q'$, *and* $(s \vdash P')\,\mathcal{Y}^{-}_{\mathcal{E};r}\,(t' \vdash Q')$.

Proof. By induction on n.

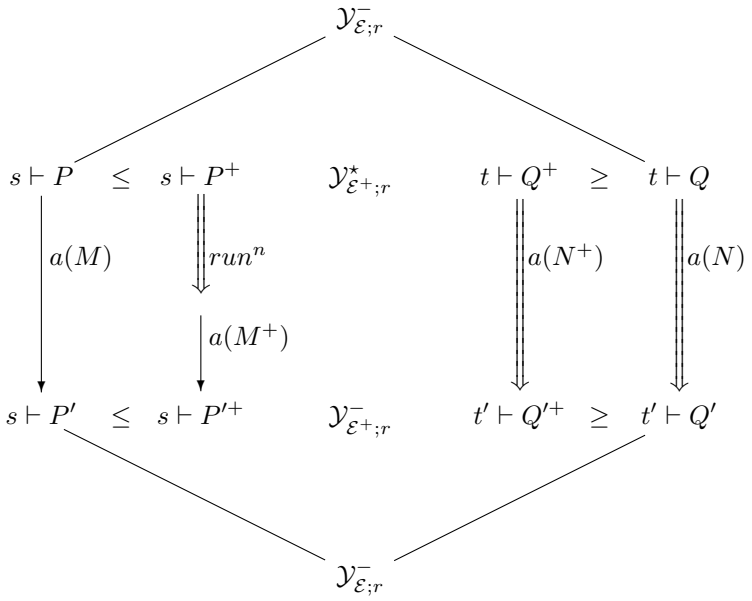
- **Case** $n = 0$
  Immediate by Lemma B.15 and the fact that $\mathcal{Y}^{\star} \subseteq \mathcal{Y}^{-}$.
- **Case** $n > 0$
  By Lemma B.35 and Lemma B.26, we have two possible subcases preserving minimality after the first *run*-transition of $s \vdash P \stackrel{run^{n}}{\Longrightarrow} \xrightarrow{a(M)} s \vdash P'$.
  - **Subcase** $s \vdash P \xrightarrow{run} s \vdash P''$, $t \vdash Q \stackrel{\tau}{\Rightarrow} t'' \vdash Q''$, $(s \vdash P'')\,\mathcal{Y}^{\star}_{\mathcal{E};r}\,(t'' \vdash Q'')$
    We have that $s \vdash P'' \stackrel{run^{n-1}}{\Longrightarrow} \xrightarrow{a(M)} s \vdash P'$ is still minimal, so we can apply the induction hypothesis, and we are done.
  - **Subcase** $s \vdash P \xrightarrow{run} s \vdash P''$, $t \vdash Q \xrightarrow{run} t \vdash Q''$, $(s \vdash P'')\mathcal{Y}^{-}_{\mathcal{E};r}(t \vdash Q'')$ with $P'' = P_0|P_1$ and $Q'' = Q_0|Q_1$, some $\mathcal{E}'$ such that $(s, r' \vdash P_0)\mathcal{Y}_{\mathcal{E}';rr'}(t, r' \vdash Q_0)$, $(P_1, Q_1) = (C_p[run(\widetilde{M}), A], C_p[run(\widetilde{N}), B])$ with $(`A, `B) \in \mathcal{E}$ and $(A,B) \notin (\mathcal{E}';rr')^{\circ}$, $(\widetilde{M}, \widetilde{N}) \in \mathcal{E}'$, $(s, r' \vdash P_0)\,\mathcal{Y}_{\mathcal{E}';rr'}\,(t, r' \vdash Q_0)$ and $r' \cap (s \cup t) = \emptyset$.
    Using Lemma B.30 (to add a fresh l to the know names) and clause 4 of environmental bisimulations up-to context, we have $(s, r', l \vdash P_0 \mid$

$l[A])\,\mathcal{Y}^{\star}_{\mathcal{E}';l\oplus rr'}\,(t, r', l \vdash Q_0 \,|\, l[B])$. Using an argument similar to the one in Lemma B.36, case 2, subcase 2, we know that we can apply the induction hypothesis to minimal transition $s, r', l \vdash P_0 \mid l[A] \xRightarrow{run^{n-1}} \xrightarrow{a(M)} s, r', l \vdash P_0 \mid l[A']$. We obtain $t, r', l \vdash Q_0 \mid l[B] \xRightarrow{a(N)} t'', r', l \vdash Q_0'' \mid l[B'']$ and $(s, r', l \vdash P_0 \,|\, l[A'])\,\mathcal{Y}^{-}_{\mathcal{E}';l\oplus rr'}\,(t'', r', l \vdash Q_0'' \,|\, l[B''])$. By Corollary B.38, after an output to channel l, we have $(s, r', l \vdash P_0)\,\mathcal{Y}^{-}_{(A',B')\oplus\mathcal{E}';l\oplus rr'}\,(t', r', l \vdash Q_0')$, hence $(s \vdash P_0 \mid C_p[run(\widetilde{M}), A'])\,\mathcal{Y}^{-}_{\mathcal{E};r}\,(t' \vdash Q_0' \mid C_p[run(\widetilde{N}), B'])$ up-to environment, context and name creation. And of course, we do have $t \vdash Q \xRightarrow{a(N)} t' \vdash Q_0' \mid C_p[run(\widetilde{N}), B']$.

Corollary B.40. [Input preserves *run*-erased environmental bisimulation up-to context]

For any environmental bisimulation up-to context $\mathcal{Y}$, if $(s \vdash P)\,\mathcal{Y}^{-}_{\mathcal{E};r}\,(t \vdash Q)$ and $s \vdash P \xrightarrow{a(M)} s \vdash P'$ with $a \in r$, then there are t', Q' such that for all $(M, N) \in (\mathcal{E}; r)^{-}$, $t \vdash Q \xRightarrow{a(N)} t' \vdash Q'$ and $(s \vdash P')\,\mathcal{Y}^{-}_{\mathcal{E};r}\,(t' \vdash Q')$. The converse on $t \vdash Q$'s transitions holds too.

Proof. By $\mathcal{Y}^{-}$'s definition, we know there are $s \vdash P^{+}$, $t \vdash Q^{+}$ and $\mathcal{E}^{+}$ such that $(s \vdash P^{+})\,\mathcal{Y}^{\star}_{\mathcal{E}^{+};r}\,(t \vdash Q^{+})$. Since $s \vdash P \xrightarrow{a(M)} s \vdash P'$, there is a minimal input transition $s \vdash P^{+} \xRightarrow{run^{n}} \xrightarrow{a(M^{+})} s \vdash P'^{+}$. By Lemma B.39, we have $t \vdash Q^{+} \xRightarrow{a(N^{+})} t \vdash Q'^{+}$ for any $(M^{+}, N^{+}) \in (\mathcal{E}^{+}; r)^{\star}$ and $(s \vdash P'^{+})\,\mathcal{Y}^{-}_{\mathcal{E}^{+};r}\,(t \vdash Q'^{+})$ which implies by Corollary B.23 that $t \vdash Q$ can also weakly do an input transition $t \vdash Q \xRightarrow{a(N)} t' \vdash Q'$ such that $Q' \leq Q'^{+}$ for any $N \leq N^{+}$, i.e. for any $(M, N) \in (\mathcal{E}; r)^{-}$. By Corollary B.29, as $P' \leq P'^{+}$, $Q' \leq Q'^{+}$ and $\mathcal{E} \leq \mathcal{E}^{+}$, we have $(s \vdash P')\,\mathcal{Y}^{-}_{\mathcal{E};r}\,(t' \vdash Q')$ as desired. Visually, the following diagram holds.



The converse on $t \vdash Q$'s transitions is shown similarly.

Lemma B.41. [*run*-expanded reduction for environmental bisimulation up-to context]
Suppose that $(s \vdash P)\mathcal{Y}^{\star}_{\mathcal{E};r}(t \vdash Q)$ *for an environmental bisimulation up-to context* $\mathcal{Y}$*, and that* $s \vdash P \stackrel{run^{n}}{\Longrightarrow} \xrightarrow{\tau} s' \vdash P'$ *is minimal with respect to* $s \vdash P^{-} \xrightarrow{\tau} s' \vdash P'^{-}$*, then* $t \vdash Q \stackrel{\tau}{\Rightarrow} t' \vdash Q'$*, and* $(s' \vdash P')\,\mathcal{Y}^{-}_{\mathcal{E};r}\,(t' \vdash Q')$*.*

Proof. By induction on n.

- **Case** $n = 0$
  Immediate by Lemma B.35.
- **Case** $n > 0$
  By Lemma B.35 and Lemma B.26, we have two possible subcases preserving minimality after the first *run*-transition of $s \vdash P \stackrel{run^{n}}{\Longrightarrow} \xrightarrow{\tau} s \vdash P'$.
  - **Subcase** $s \vdash P \xrightarrow{run} s \vdash P''$, $t \vdash Q \stackrel{\tau}{\Rightarrow} t'' \vdash Q''$, $(s \vdash P'')\,\mathcal{Y}^{\star}_{\mathcal{E};r}\,(t'' \vdash Q'')$
    We have that $s \vdash P'' \stackrel{run^{n-1}}{\Longrightarrow} \xrightarrow{\tau} s \vdash P'$ is still minimal with respect to $s \vdash P^{-} \xrightarrow{\tau} s \vdash P'^{-}$, so we can apply the induction hypothesis, and we are done.
  - **Subcase** $s \vdash P \xrightarrow{run} s \vdash P''$, $t \vdash Q \xrightarrow{run} t \vdash Q''$, $(s \vdash P'')\,\mathcal{Y}^{-}_{\mathcal{E};r}\,(t \vdash Q'')$ hold, and we have $P'' = P_0 \,|\, P_1$ and $Q'' = Q_0 \,|\, Q_1$ with $(s, r' \vdash P_0)\,\mathcal{Y}_{\mathcal{E}';rr'}$ $(t, r' \vdash Q_0)$, $(`P_1, `Q_1) = (`C_p[run(\widetilde{M}), A], `C_p[run(\widetilde{N}), B]) \in (\mathcal{E}'; rr')^{-}$ with $(A, B) \notin (\mathcal{E}'; rr')^{\circ}$, $((\widetilde{M}; `A), (\widetilde{N}, `B)) \in \mathcal{E}'$, $r' \cap (s \cup t) = \emptyset$. Since we know $s, r' \vdash P^{-} \xrightarrow{\tau} s', r' \vdash P'^{-}$ and that $s, r' \vdash P \stackrel{run^{n}}{\Longrightarrow} \xrightarrow{\tau} s', r' \vdash P'$ is minimal with respect to it, we can infer how P weakly reduces to P'. Let us analyse each possibility.

* **Subsubcase** A reacts with P_0
  Using clause 5 of environmental bisimulation up-to context to add a new name l to known names, and clause 4 to spawn A, we have $(s, l, r' \vdash P_0 \mid l[A])\ \mathcal{Y}^{\star}_{\mathcal{E}';rr'l}\ (t, l, r' \vdash Q_0 \mid l[B])$. Using an argument similar to the one in Lemma B.36, case 2, subcase 2, we know that we can apply the induction hypothesis to minimal transition $s, r', l \vdash P_0 \mid l[A] \xrightarrow{run^{n-1}} \xrightarrow{\tau} s', r', l \vdash P_0' \mid l[A']$. We obtain $t, r', l \vdash Q_0 \mid l[B] \stackrel{\tau}{\Rightarrow} t'', r', l \vdash Q_0'' \mid l[B'']$ and $(s', r', l \vdash P_0' \mid l[A'])\ \mathcal{Y}^{-}_{\mathcal{E}';rr'l}\ (t'', r', l \vdash Q_0'' \mid l[B''])$. Therefore, by $(s', r', l \vdash P_0' \mid l[A'])\, \mathcal{Y}^{-}_{\mathcal{E}';rr'l}\, (t'', r', l \vdash Q_0'' \mid l[B''])$ and by Corollary B.38, we have $(s', r', l \vdash P_0')\ \mathcal{Y}^{-}_{(\text{`}A', \text{`}B')\oplus\mathcal{E}';rr'l}\ (t', r', l \vdash Q_0')$ hence $(s', r', l \vdash P_0' \mid C_p[run(\widetilde{M}), A'])\ \mathcal{Y}^{-}_{(\text{`}A', \text{`}B')\oplus\mathcal{E}';rr'l}\ (t', r', l \vdash Q_0' \mid C_p[run(\widetilde{N}), B'])$ up-to context, $(s' \vdash P_0' \mid C_p[run(\widetilde{M}), A'])\ \mathcal{Y}^{-}_{\mathcal{E};r}\ (t' \vdash Q_0' \mid C_p[run(\widetilde{N}), B'])$ up-to environment and name creation. Also, $t, r', l \vdash Q'' \stackrel{\tau}{\Rightarrow} t', r', l \vdash Q_0' \mid C_p[\widetilde{N}, B']$, holds as well from the above, hence $t \vdash Q \stackrel{\tau}{\Rightarrow} t' \vdash Q_0' \mid C_p[run(\widetilde{N}), B']$.
* **Subsubcase** A reduces alone
  Similarly, but with $P_0' = P_0$.
* **Subsubcase** A reacts with an A_i from $\text{`}\widetilde{A} = \widetilde{M}$ (or a *run*-erasure of it)
  Let $G = C[\text{`}A_i, \widetilde{M'}]$ (resp. $H = C[\text{`}B_i, \widetilde{N'}]$) be the process of P_1 (resp. Q_1) in redex position that contains A_i (resp. B_i) and reacts with the process containing A according to rule React-R or React-L. Then there is a process context C_p' such that $C_p[A, run(\widetilde{M})] = C_p'[A', G, run(\widetilde{M''})]$.
  By clause 5 of environmental bisimulation up-to context to add a new name l and clause 4, we have $(s, r', l \vdash P_0 \mid l[A])\, \mathcal{Y}^{\star}_{\mathcal{E}';rr'l}\, (t, r', l \vdash Q_0 \mid l[B])$. By Lemma B.30 to add a new name m and by up-to context, we have $(s, r', l, m \vdash P_0 \mid l[A] \mid m[G])\mathcal{Y}^{\star}_{\mathcal{E}';m\oplus rr'l}(t, r', l, m \vdash Q_0 \mid l[B] \mid m[H])$. Then, $s, r', l, m \vdash P_0 \mid l[A] \mid m[G] \stackrel{run}{\Longrightarrow} \xrightarrow{\tau} s, r', l, m \vdash P_0 \mid l[A'] \mid m[G']$. Applying the induction hypothesis, we obtain $t, r', l, m \vdash Q_0 \mid l[B] \mid m[H] \stackrel{\tau}{\Rightarrow} t'', r', l, m \vdash Q_0'' \mid l[B''] \mid m[H'']$, as well as $(s, r', l, m \vdash P_0 \mid l[A'] \mid m[G'])\ \mathcal{Y}^{-}_{\mathcal{E}';m\oplus rr'l}\ (t'', r', l, m \vdash Q_0' \mid l[B''] \mid m[H''])$.
  We can now passivate the contents of $l[\ ]$ and $m[\ ]$ and use up-to context, environment and name creation to get $(s \vdash P_0 \mid C_p'[A', G', run(\widetilde{M''})])\mathcal{Y}^{-}_{\mathcal{E};r}$ $(t' \vdash Q_0' \mid C_p'[B', H', run(\widetilde{N''})])$. It also follows that $t \vdash Q \stackrel{\tau}{\Rightarrow} t' \vdash Q_0' \mid C_p'[B', H', run(\widetilde{N''})]$ as expected.
* **Subsubcase** A outputs and reacts with the context
  Let $C[\widetilde{M'}]$ (resp. $C[\widetilde{N'}]$) be the process of P_1 (resp. Q_1) in redex position that reacts with the process containing A according to rule React-R or React-L. Then there is another context C_p' such that $C_p[A, run(\widetilde{M})] = C_p'[A, C[\widetilde{M'}], run(\widetilde{M''})]$. Using clause 5 of environmental bisimulation up-to context to add a new name l and clause 4

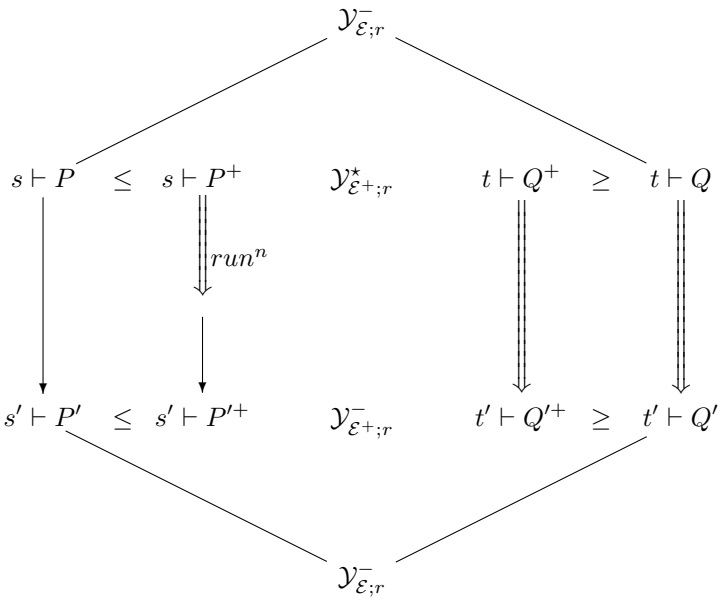
to spawn A, we have $(s, r', l \vdash P_0 \mid l[A])\ \mathcal{Y}^{\star}_{\mathcal{E}';rr'l}\ (t, r', l \vdash Q_0 \mid l[B])$. We can now apply Lemma B.37 to simulate A's output after *run*-transitions: $s, r', l \vdash P_0 \mid l[A] \stackrel{run^{n-1}}{\Longrightarrow} \xrightarrow{\overline{a}\langle M\rangle} s, r', l \vdash P_0 \mid l[A']$, and we obtain $t, r', l \vdash Q_0 \mid l[B] \stackrel{\overline{a}\langle N\rangle}{\Longrightarrow} t'', r', l \vdash Q_0'' \mid l[B'']$, and $(s, r', l \vdash P_0 \mid l[A'])\ \mathcal{Y}^{-}_{(M,N)\oplus\mathcal{E}';rr'l}\ (t'', r', l \vdash Q_0'' \mid l[B''])$. We passivate the contents of $l[\ ]$ and have $(s, r', l \vdash P_0)\ \mathcal{Y}^{-}_{(`A', `B')\oplus(M,N)\oplus\mathcal{E}';rr'l}\ (t', r', l \vdash Q_0')$. By Lemma B.12, we have that $s, r', l \vdash C[\widetilde{M}] \xrightarrow{a(M)} s, r', l \vdash G$ and $t''', r', l \vdash C[\widetilde{N}] \xrightarrow{a(N)} t''', r', l \vdash H$ with $(G, H) \in ((M, N)\oplus\mathcal{E}'; rr'l)^{\circ}$ (for some "intermediate" t''') such that we have $t'', r', l \vdash Q_0'' \mid l[B''] \mid C[\widetilde{N}] \stackrel{\tau}{\Rightarrow} t''', r', l \vdash Q_0''' \mid l[B'''] \mid C[\widetilde{N}] \xrightarrow{\tau} t''', r', l \vdash Q_0'''' \mid l[B''''] \mid H \stackrel{\tau}{\Rightarrow} t', r', l \vdash Q_0' \mid l[B'] \mid H$ hence $t \vdash Q_0 \mid Q_1 \stackrel{\tau}{\Rightarrow} t' \vdash Q_0' \mid Q_1'$. From $(s, r', l \vdash P_0)\ \mathcal{Y}^{-}_{(`A', `B')\oplus(M,N)\oplus\mathcal{E}';rr'l}\ (t', r', l \vdash Q_0')$. we build up-to context, environment and name creation $(s \vdash P_0 \mid C_p'[A', G, run(\widetilde{M})])\ \mathcal{Y}^{-}_{\mathcal{E};r}\ (t' \vdash Q_0' \mid C_p[B', H, run(\widetilde{N})])$.

∗ **Subsubcase** A inputs and reacts with the context
Suppose the context outputs a process M (resp. N by Lemma B.13) by means of a process $C_o[\widetilde{M'}]$ (resp. $C_o[\widetilde{N'}]$). Using clause 5 of environmental bisimulation up-to context several times to add a new name l and clause 4 to spawn A, we have $(s, r', l \vdash P_0 \mid l[A])\ \mathcal{Y}^{\star}_{\mathcal{E}';rr'l}$ $(t, r', l \vdash Q_0 \mid l[B])$. We can now apply Lemma B.39 to trigger A's input of M and $Q_0 \mid l[B]$'s simulation of that input, and we obtain $(s, r', l \vdash P_0 \mid l[A'])\, \mathcal{Y}^{-}_{\mathcal{E}';rr'l}\, (t'', r', l \vdash Q_0'' \mid l[B''])$. We passivate the content of $l[\ ]$, obtaining $(s, r', l \vdash P_0)\ \mathcal{Y}^{-}_{(`A', `B')\oplus\mathcal{E}';rr'l}\ (t', r', l \vdash Q_0')$. We remove r' and l up-to name creation from the known names and then replace A' and B' up-to context, giving $(s \vdash P_0 | C_p'[A', run(\widetilde{M})])\mathcal{Y}^{-}_{(`A', `B')\oplus\mathcal{E}';r}$ $(t' \vdash Q_0' \mid C_p'[B', run(\widetilde{N})])$. Finally, up-to environment, we have $(s \vdash P_0 \mid C_p'[A', run(\widetilde{M})])\ \mathcal{Y}^{-}_{\mathcal{E};r}\ (t' \vdash Q_0' \mid C_p[B', run(\widetilde{N})])$. The transition $t \vdash Q_0 \mid Q_1 \stackrel{\tau}{\Rightarrow} t' \vdash Q_0' \mid C_p[B', run(\widetilde{N})]$ is derived from the above, composing input $t \vdash Q_0 \mid Q_1 \stackrel{\tau}{\Rightarrow} \xrightarrow{a(N)} \stackrel{\tau}{\Rightarrow} t' \vdash Q_0' \mid C_p[B', run(\widetilde{N})]$ with the output from the context.

Corollary B.42. [Reduction preserves *run*-erased environmental bisimulation up-to context]
For any environmental bisimulation up-to context $\mathcal{Y}$, if $(s \vdash P)\ \mathcal{Y}^{-}_{\mathcal{E};r}\ (t \vdash Q)$ and $s \vdash P \xrightarrow{\tau} s' \vdash P'$, then there are t' and Q' such that $t \vdash Q \stackrel{\tau}{\Rightarrow} t' \vdash Q'$ and $(s' \vdash P')\ \mathcal{Y}^{-}_{\mathcal{E};r}\ (t' \vdash Q')$. The converse on $t \vdash Q$'s transitions holds too.

Proof. By $\mathcal{Y}^{-}$'s definition, we know there are $s \vdash P^{+}$, $t \vdash Q^{+}$ and $\mathcal{E}^{+}$ such that $(s \vdash P^{+})\ \mathcal{Y}^{\star}_{\mathcal{E}^{+};r}\ (t \vdash Q^{+})$. Since $s \vdash P \xrightarrow{\tau} s' \vdash P'$, there is a minimal reduction transition $s \vdash P^{+} \stackrel{run^{n}}{\Longrightarrow} \xrightarrow{\tau} s' \vdash P'^{+}$. By Lemma B.41, we have $t \vdash Q^{+} \stackrel{\tau}{\Rightarrow} t' \vdash Q'^{+}$ and $(s' \vdash P'^{+})\ \mathcal{Y}^{-}_{\mathcal{E}^{+};r}\ (t' \vdash Q'^{+})$ which implies by Corollary B.23 that $t \vdash Q$ can

also weakly reduce to some $t' \vdash Q'$ such that $Q' \leq Q'^{+}$. By Corollary B.29, as $P' \leq P'^{+}$, $Q' \leq Q'^{+}$ and $\mathcal{E} \leq \mathcal{E}^{+}$, we have $(s' \vdash P')\, \mathcal{Y}^{-}_{\mathcal{E};r}\, (t' \vdash Q')$ as desired. Visually, the following diagram holds.



The converse on $t \vdash Q$'s transitions is shown similarly.

Theorem B.43. [Soundness of environmental bisimulation up-to context] *If $\mathcal{Y}$ is a environmental bisimulation up-to context, then $\mathcal{Y}^{-}$ is included in bisimilarity.*

Proof. Let $\mathcal{X} = \{(\mathcal{E}, r, s, P, t, Q) \mid (s \vdash P)\, \mathcal{Y}^{-}_{\mathcal{E};r}\, (t \vdash Q)\}$ and let us prove that $\mathcal{X}$ verifies each clause of environmental bisimulation.

1. By Corollary B.42, whenever $s \vdash P \xrightarrow{\tau} s' \vdash P'$, we have a $t' \vdash Q'$ such that $t \vdash Q \stackrel{\tau}{\Rightarrow} t' \vdash Q'$ and $(s \vdash P')\, \mathcal{Y}^{-}_{\mathcal{E};r}\, (t' \vdash Q')$, i.e. $(s \vdash P')\, \mathcal{X}_{\mathcal{E};r}\, (t' \vdash Q')$.
2. By Corollary B.38, whenever $s \vdash P \xrightarrow{\overline{a}\langle M \rangle} s \vdash P'$ with $a \in r$, we have a $t' \vdash Q'$ such that $t \vdash Q \stackrel{\overline{a}\langle N \rangle}{\Longrightarrow} t' \vdash Q'$ and $(s \vdash P')\, \mathcal{Y}^{-}_{(M,N)\oplus\mathcal{E};r}\, (t' \vdash Q')$, i.e. $(s \vdash P')\, \mathcal{X}_{(M,N)\oplus\mathcal{E};r}\, (t' \vdash Q')$.
3. By Corollary B.40, whenever $s \vdash P \xrightarrow{a(M)} s \vdash P'$ with $a \in r$, we have for all $(M, N) \in (\mathcal{E}; r)^{\star}$ a $t' \vdash Q'$ such that $t \vdash Q \stackrel{a(N)}{\Longrightarrow} t' \vdash Q'$ with $(s \vdash P')\, \mathcal{Y}^{-}_{\mathcal{E};r}\, (t' \vdash Q')$, i.e. $(s \vdash P')\, \mathcal{X}_{\mathcal{E};r}\, (t' \vdash Q')$.
4. By Lemma B.31, we have $(s \vdash P^{+} \,|\, l[P_1^{+}])\, \mathcal{Y}^{\star}_{\mathcal{E}^{+};r}\, (t \vdash Q^{+} \,|\, l[Q_1^{+}])$ for some $(s \vdash P^{+})\, \mathcal{Y}^{\star}_{\mathcal{E}^{+};r}\, (t \vdash Q^{+})$ with $P \leq P^{+}$, $Q \leq Q^{+}$, $\mathcal{E} \subseteq \leq \mathcal{E}^{+} \geq$, and $(`P_1, `Q_1) \leq (`P_1^{+}, `Q_1^{+}) \in \mathcal{E}^{+}$, whose existence is guaranteed by definition of $\mathcal{Y}^{-}$. Then, by *run*-erasure, we have $(s \vdash P \,|\, l[P_1])\, \mathcal{Y}^{-}_{\mathcal{E};r}\, (t \vdash Q \,|\, l[Q_1])$.

5. By Lemma B.30, we have for any n not in $s \cup t$, $(s, n \vdash P)\, \mathcal{Y}^{-}_{\mathcal{E};n\oplus r}\, (t, n \vdash Q)$, i.e. $(s, n \vdash P)\, \mathcal{X}_{\mathcal{E};n\oplus r}\, (t, n \vdash Q)$.
6. Similarly, the converse of the first three clauses holds too.

Theorem B.44. [Reduction-closed Barbed equivalence from environmental bisimulation]
Let $r \subseteq f = fn(P,Q)$, then if $(f \vdash P)\, \mathcal{Y}^{-}_{\emptyset;r}\, (f \vdash Q)$ for an environmental bisimulation up-to context $\mathcal{Y}$, then $f \vdash P \approx_r f \vdash Q$.

Proof. We know by Theorem B.43 that $\mathcal{Y}^{-}$ is an environmental bisimulation. We let $\mathcal{Z} = \{(r, s, P, t, Q) \mid fn(P,Q) \subseteq s \cap t, \quad r \subseteq s \cap t, \quad (s \vdash P)\, \mathcal{Y}^{-}_{\emptyset;r}\, (t \vdash Q)\}$ and prove that $\mathcal{Z}$ is included in $\approx$.

1. **Clause** $s \vdash P \xrightarrow{\tau} s' \vdash P$
   As $\mathcal{Y}^{-}$ is an environmental bisimulation, by clause 1 of the bisimulation, there is $t' \vdash Q'$ such that $t \vdash Q \Rightarrow t' \vdash Q'$ and $(s' \vdash P')\, \mathcal{Y}^{-}_{\emptyset;r}\, (t' \vdash Q')$. We have $fn(P', Q') \subseteq s' \cap t'$ and $r \subseteq s' \cap t'$, henc $(r, s', P', t', Q') \in \mathcal{Z}$.
2. **Clause** $P \downarrow_{\mu} \quad \mu$ *or* $\overline{\mu} \in r$
   There are two cases depending on μ:
   - **Case** $s \vdash P \downarrow_a$
     We have that $s \vdash P \xrightarrow{a(M)} s \vdash P'$ for some $(M, N) \in (\emptyset; r)^{\star}$ and P'. Since $a \in r$, by $\mathcal{Y}^{-}$ being an environmental bisimulation and clause 2 of the bisimulation, there is also $t' \vdash Q'$ such that $t \vdash Q \xRightarrow{a(N)} t' \vdash Q'$, that is, $t \vdash Q \Downarrow_a$.
   - **Case** $s \vdash P \downarrow_{\overline{a}}$
     We have that $s \vdash P \xrightarrow{\overline{a}\langle M\rangle} s \vdash P'$ for some M and P'. Since $a \in r$, by $\mathcal{Y}^{-}$ being an environmental bisimulation and clause 3 of the bisimulation, there are also N, $t' \vdash Q'$ and such that $t \vdash Q \xRightarrow{\overline{a}\langle N\rangle} t' \vdash Q'$, that is, $t \vdash Q \Downarrow_a$.
3. **Clause** Converse of 1, 2 on Q
   Similar to 1, 2.
4. **Clause** R a process, $fn(R) \cap ((s \cup t) \setminus r) = \emptyset$
   Let $r' = fn(R)$; by appealing to the clause 5 of the bisimulation, since the names in r' are either fresh or already in r, we have that $(s, r' \vdash P)\mathcal{Y}^{-}_{\emptyset;rr'}(t, r' \vdash Q)$. Also, $(R, R) \in (\emptyset; rr')^{\circ}$ and thus, using the up-to context technique, $(s, r' \vdash P \mid R)\, \mathcal{Y}^{-}_{\emptyset;rr'}\, (t, r' \vdash Q \mid R)$ since $(s, r' \vdash \cdot)\, \mathcal{Y}^{-}_{\mathcal{E};rr'}\, (t, r' \vdash \cdot)$ is preserved by parallel composition of processes from $(\emptyset; rr')^{\circ}$. Therefore, $(rr', sr', P \mid R, tr', Q \mid R) \in \mathcal{Z}$.

Conclusion: we showed that $\mathcal{Z} \subseteq \approx$ and as $(f \vdash P)\, \mathcal{Y}^{-}_{\emptyset;r}\, (f \vdash Q)$ for $r \subseteq f = fn(P,Q)$ implies $(r, f, P, f, Q) \in \mathcal{Z}$, we have that $f \vdash P \approx_r f \vdash Q$.

3 Reduction-closed barbed congruence from environmental bisimulations

Definition B.44. *Capturing reduction-closed barbed congruence $\cong$ is the largest binary relation on variables configurations indexed by a set of names $r \subseteq s \cap t$ such that when $s \vdash P \cong_r t \vdash Q$,*

- *$s \vdash P \xrightarrow{\tau} s' \vdash P'$ implies there are Q' and t' such that $t \vdash Q \Rightarrow t' \vdash Q'$ and $s' \vdash P' \cong_r t' \vdash Q'$,*
- *$s \vdash P \downarrow_\mu$ implies $t \vdash Q \Downarrow_\mu$, if $\mu \in r$ or $\overline{\mu} \in r$,*
- *the converse of the above two on Q, and*
- *for all context C_p with holes for process and which can capture names such that $fn(C_p) \cap ((s \cup t) \setminus r) = \emptyset$, $bn(C_p) \cap ((s \cup t) \setminus r) = \emptyset$, we have $s \cup fn(C_p) \vdash C_p[P] \cong_{r,fn(C_p)} t \cup fn(C_p) \vdash C_p[Q]$.*

Theorem B.45. *There exist P, Q, r and a such that $\{r, a\} = fn(P, Q)$ and $(r, a \vdash \overline{a}\langle P\rangle)\, \mathcal{Y}_{\emptyset;ra}\, (r, a \vdash \overline{a}\langle Q\rangle)$ but not $r, a \vdash P \cong_{ra} r, a \vdash Q$.*

Proof. We provide such P, Q, r

We let $P_{1x} = a(X).(X \mid i(Y).\overline{x}) \mid x.\overline{f}$, let $Q_{1x} = a(X).(X \mid i(Y).Y) \mid x.\overline{f}$, let $P_{2x} = \overline{a}\langle\overline{i}\langle\overline{x}\rangle\rangle$ and $Q_{2x} = \overline{a}\langle\overline{i}\langle\overline{x}\rangle\rangle$.

We then consider the two processes

$$P = \nu i.(\overline{b}\langle P_{1m}\rangle \mid \overline{c}\langle P_{2m}\rangle) = \nu i.(\overline{b}\langle a(X).(X \mid i(Y).\overline{m}) \mid m.\overline{f}\rangle \mid \overline{a}\langle\overline{i}\langle\overline{m}\rangle\rangle)$$

and

$$Q = \nu i.(\overline{b}\langle Q_{1m}\rangle \mid \overline{c}\langle Q_{2m}\rangle) = \nu i.(\overline{b}\langle a(X).(X \mid i(Y).Y) \mid m.\overline{f}\rangle \mid \overline{a}\langle\overline{i}\langle\overline{m}\rangle\rangle)$$

which are such that

$$fn(a, P, Q) \vdash \overline{a}\langle P\rangle \mathcal{Y}_{\emptyset;fn(a,P,Q)} fn(a, P, Q) \vdash \overline{a}\langle Q\rangle$$

We let $n = \{a, b, c, f, m\}$ the set of there free names, and compare P and Q under this knowledge:

$$n \vdash \nu i.(\overline{b}\langle P_{1m}\rangle \mid \overline{c}\langle P_{2m}\rangle) \cong_n n \vdash \nu i.(\overline{b}\langle Q_{1m}\rangle \mid \overline{c}\langle Q_{2m}\rangle)$$

We first consider a reduction on the left hand-side (lhs) to create the name i, and let the right hand-side (rhs) follow weakly. We then do a reduction on the rhs to force the creation of the (other) name i if it had not be done, and let the lhs as is. We then are guaranteed to have:

$$n, i \vdash \overline{b}\langle P_{1m}\rangle \mid \overline{c}\langle P_{2m}\rangle \cong_n n, i \vdash \overline{b}\langle Q_{1m}\rangle \mid \overline{c}\langle Q_{2m}\rangle$$

We can now create a context $C_p = \overline{a}\langle\nu m.[\,]_1\rangle | a(X).(X|X) | g.b.b(X).c.c(Y).(X|Y)$ that will allow us to:

- capture the name m,
- duplicate the processes with captured names,

– discard subprocesses and keep others
– have a side of the equivalence always show a barb while the other can never.

$$n, i, g \vdash C_p[\overline{b}\langle P_{1m}\rangle \mid \overline{c}\langle P_{2m}\rangle] \cong_{n,g} n, i, g \vdash C_p[\overline{b}\langle Q_{1m}\rangle \mid \overline{c}\langle Q_{2m}\rangle]$$

Then, we force the reaction between $\overline{a}$ and $a(X)$ and immediately after we create the names x and y. The conditions on the fresh barb g will ensure that we do not have (weak) uncontrolled reactions. We write C'_p for $g.b.b(X).c.c(Y).(X \mid Y)$ since we lack space.

$$n, i, g, x, y \vdash \overline{b}\langle P_{1x}\rangle|\overline{c}\langle P_{2x}\rangle|\overline{b}\langle P_{1y}\rangle|\overline{c}\langle P_{2y}\rangle|C'_p \cong_{n,g} n, i, g, x, y \vdash \overline{b}\langle Q_{1x}\rangle|\overline{c}\langle Q_{2x}\rangle|\overline{b}\langle Q_{1y}\rangle|\overline{c}\langle Q_{2y}\rangle|C'_p$$

We can now spawn in parallel a process $\overline{g}$ to remove the guard of C'_p, and then *in the rhs*, consider the reactions that select Q_{1x} and Q_{2y}, discarding the other subprocesses. The left hand-side will follow, choosing to discard two subprocesses, keeping two others which we will call P_1 (whose "free private name" we will call z, whichever of x or y it is) and P_2. The fact that P_1 and P_2 may weakly continue to react after being sent is not an issue, and we will consider they did not.

$$n, i, g, x, y \vdash P_1 \mid P_2 \cong_{n,g} n, i, g, x, y \vdash Q_{1x} \mid Q_{2y}$$

Then, P_1 and Q_1 may react on channel a, and rhs will follow somehow.

$$n, i, g, x, y \vdash \overline{i}\langle x \text{ or } y\rangle \mid i(Y).\overline{z} \mid z.\overline{f} \cong_{n,g} n, i, g, x, y \vdash \ldots$$

Then *lhs* can react on i, and both $\overline{z}$ and $z.\overline{f}$ will react, exhibiting barb $\overline{f}$.

$$n, i, g, x, y \vdash \overline{i}\langle x \text{ or } y\rangle \mid i(Y).\overline{z} \mid z.\overline{f} \xrightarrow{\tau} n, i, g, x, y \vdash \overline{x} \mid x.\overline{f} \xrightarrow{\tau} n, i, g, x, y \vdash \overline{f} \downarrow_{\overline{f}}$$

However, rhs will never be able to exhibit barb $\overline{f}$ since it will be stuck at

$$n, i, g, x, y \vdash \overline{i}\langle x\rangle \mid i(Y).Y \mid y.\overline{f} \xrightarrow{\tau} n, i, g, x, y \vdash \overline{x} \mid y.\overline{f}$$

where x and y cannot react.

Lemma B.46. $\{(r, s, P, t, Q) \mid r \subseteq r',\ (r', s, P, t, Q) \in\ \dot{\approx}\} \in\ \dot{\approx}$.

Proof. By verifying the clauses of $\dot{\approx}$.

Lemma B.47. *Let $\mathcal{Y}$ be the environmental bisimilarity up-to context, and*

$$\begin{aligned}\mathcal{S} = \{(r, s, P, t, Q) \mid\ & P \leq C[\widetilde{P}^+],\ Q \leq C[\widetilde{Q}^+],\\ & fn(P^+) \subseteq s_0,\ fn(Q^+) \subseteq t_0,\\ & (\widetilde{P}^+, \widetilde{Q}^+) \subseteq \mathcal{E},\\ & r \subseteq r_0 \cup fn(C),\\ & s = s_0 \cup fn(C),\ t = t_0 \cup fn(C)\\ & fn(C) \cap ((s \cup t) \setminus r_0) = \emptyset,\\ & bn(C) \cap (s \cup t) = \emptyset,\\ & (s_0 \vdash 0)\ \mathcal{Y}_{\mathcal{E};r_0}\ (t_0 \vdash 0)\}\end{aligned}$$

for contexts C for processes. We show that for all closed $(r, s, P, t, Q) \in \mathcal{S}$, if $s \vdash P \downarrow_\mu$ and μ or $\overline{\mu}$ is in r, then $t \vdash Q \Downarrow_\mu$, and that if $s \vdash P \to s' \vdash P'$ then $t \vdash Q \Rightarrow t' \vdash Q'$ for some $t' \vdash Q'$ with $(r, s', P', t', Q') \in \mathcal{S}$, and conversely.

Proof. By induction on the transition derivation of $s \vdash P \xrightarrow{\alpha} s' \vdash P'$ with $(r, s, P, t, Q) \in \mathcal{S}$. We prove the two properties separately. In both situations, there is a case analysis on who does the transition: the context's erasure or some P_i. By symmetricity, we do not show the converse proofs on Q's transition; they are similar. We write $run^*`(P)$ to mean $run`(\dots(run`(P))\dots)$, and C_p^-, C_q^- for two possibly different erasures of the same C.

Barbs: the cases necessary to check for barbs are Ho-In, Ho-Out, Par-R, Par-L, Rep, Transp, and Passiv.

- Ho-In
  - **Subcase** $C_p^-[\widetilde{P}] = a(X).C_{1p}^-[\widetilde{P}]$

    If $s \vdash C_p^-[\widetilde{P}] \downarrow_a$, we have $s \vdash a(X).C_{1p}^-[\widetilde{P}] \xrightarrow{a(M)} \cdot$. Thus $t \vdash C_q^-[\widetilde{Q}] = t \vdash run^*`(a(X).C_{1q}^-[\widetilde{Q}]) \overset{run}{\Longrightarrow} t \vdash a(X).C_{1q}^-[\widetilde{Q}] \xrightarrow{a(N)} \cdot$, i.e. $C_q^-[\widetilde{Q}] \Downarrow_a$
  - **Subcase** $[\,]_i$

    By $(\mathcal{E}, r, s, 0, t, 0) \in \mathcal{Y}$, we have $(s \vdash P_i)\, \mathcal{Y}_{\emptyset;r}^-\, (t \vdash Q_i)$ up-to environment, hence by Theorem B.44 $t \vdash Q_i \Downarrow_a$ if $s \vdash P_i \downarrow_a$, hence $t \vdash run^*`Q_i \Downarrow_a$, that is $t \vdash C_p^-[\widetilde{Q}] \Downarrow_a$.
- Ho-Out
  - **Subcase** $C_p^-[\widetilde{P}] = \overline{a}\langle `D_p^-[\widetilde{P}]\rangle.C_{1p}^-[\widetilde{P}]$

    If $s \vdash C_p^-[\widetilde{P}] \downarrow_{\overline{a}}$, we have $s \vdash \overline{a}\langle `D_p^-[\widetilde{P}]\rangle.C_{1p}^-[\widetilde{P}] \xrightarrow{\overline{a}\langle `D_p^-[\widetilde{P}]\rangle} \cdot$. Thus, $t \vdash C_q^-[\widetilde{Q}] = t \vdash run^*`(\overline{a}\langle `D_p^-[\widetilde{Q}]\rangle.C_{1q}^-[\widetilde{Q}]) \overset{run}{\Longrightarrow} t \vdash \overline{a}\langle `D_p^-[\widetilde{Q}]\rangle.C_{1q}^-[\widetilde{Q}] \xrightarrow{\overline{a}\langle `D_p^-[\widetilde{Q}]\rangle} \cdot$ i.e. $t \vdash C_q^-[\widetilde{Q}] \Downarrow_{\overline{a}}$
  - **Subcase** $[\,]_i$

    By $(\mathcal{E}, r, s, 0, t, 0) \in \mathcal{Y}$, we have $(s \vdash P_i)\, \mathcal{Y}_{\emptyset;r}^-\, (t \vdash Q_i)$, hence $t \vdash Q_i \Downarrow_{\overline{a}}$ if $s \vdash P_i \downarrow_{\overline{a}}$, hence $t \vdash run^*`Q_i \Downarrow_{\overline{a}}$, that is $t \vdash C_p^-[\widetilde{Q}] \Downarrow_{\overline{a}}$.

    In all the other cases, this subcase is similar, and we will thus not write it anymore.
- Passiv

  $s \vdash C_p^-[\widetilde{P}] = s \vdash a[C_{1p}^-[\widetilde{P}]] \downarrow_{\overline{a}}$. Trivially $t \vdash a[C_{1q}^-[\widetilde{Q}]] \downarrow_{\overline{a}}$, hence $t \vdash C_q^-[\widetilde{Q}] = t \vdash run^*`(a[C_{1q}^-[\widetilde{Q}]]) \Downarrow_{\overline{a}}$.
- Par-L

  $s \vdash C_p^-[\widetilde{P}] = s \vdash C_{1p}^-[\widetilde{P}] \mid C_{2p}^-[\widetilde{P}] \downarrow_\mu$ hence $s \vdash C_{1p}^-[\widetilde{P}] \downarrow_\mu$. By the induction hypothesis, $t \vdash C_{1q}^-[\widetilde{Q}] \Downarrow_\mu$ hence $t \vdash C_q^-[\widetilde{Q}] = t \vdash run^*`(C_{1q}^-[\widetilde{Q}] \mid C_{2q}^-[\widetilde{Q}]) \Downarrow_\mu$.
- Par-R

  Similarly.
- Rep

  $s \vdash C_p^-[\widetilde{P}] = s \vdash !C_{1p}^-[\widetilde{P}] \downarrow_\mu$, hence $s \vdash !C_{1p}^-[\widetilde{P}] \mid C_{1p}^-[\widetilde{P}] \downarrow_\mu$. By the induction hypothesis, $t \vdash !C_{1q}^-[\widetilde{Q}] \mid C_{1q}^-[\widetilde{Q}] \Downarrow_\mu$, hence $t \vdash C_q^-[\widetilde{Q}] = t \vdash run^*`(!C_{1q}^-[\widetilde{Q}]) \Downarrow_\mu$.

– Transp
$s \vdash C_p^-[\widetilde{P}] = s \vdash a[C_{1p}^-[\widetilde{P}]] \downarrow_\mu$, hence $s \vdash C_{1p}^-[\widetilde{P}] \downarrow_\mu$. By the induction hypothesis, $t \vdash C_{1q}^-[\widetilde{Q}] \Downarrow_\mu$, hence $t \vdash a[C_{1q}^-[\widetilde{Q}]] \Downarrow_\mu$, hence $t \vdash C_q^-[\widetilde{Q}] = t \vdash run^*{\scriptstyle `}(a[C_{1q}^-[\widetilde{Q}]]) \Downarrow_\mu$.

Reductions: the cases necessary to check for reduction closure are Run, Transp, Par-L, Par-R, Rep, Create, React-L, and React-R.

The Create case is particularly different from the others.

– Create
- $s \vdash C_p^-[\widetilde{P}] = s \vdash \nu x.C_{1p}^-[\widetilde{P}] \to s, x \vdash C_{1p}^-[\widetilde{P}]$ and $t \vdash C_q^-[\widetilde{Q}] = t \vdash run^*{\scriptstyle `}(\nu x.C_{1q}^-[\widetilde{Q}])$
  A similar transition can be taken by Q: $t \vdash C_q^-[\widetilde{Q}] = run^*{\scriptstyle `}(\nu x.\, C_{1q}^-[\widetilde{Q}]) \stackrel{run}{\Longrightarrow} t \vdash \nu x.C_{1q}^-[\widetilde{Q}] \to t, x \vdash C_{1q}^-[\widetilde{Q}]$ (we can choose the same name x for both creations).
  By $(s \vdash 0)\, \mathcal{Y}_{\mathcal{E};r}^-\, (t \vdash 0)$ and freshness of x, we have $(s, x \vdash 0)\, \mathcal{Y}_{\mathcal{E};rx}^-\, (t, x \vdash 0)$ which implies $(r, sx, C_{1p}^-[\widetilde{P}], tx, C_{1q}^-[\widetilde{Q}]) \in \mathcal{S}$.
- $[\ ]_i$
  We have $(s \vdash a[P_i])\ \mathcal{Y}_{\mathcal{E};r}^-\ (t \vdash a[Q_i])$ and $s \vdash a[P_i] \to s, x \vdash a[P_i'] \xrightarrow{\overline{a}\langle {\scriptstyle `}P_i' \rangle} s, x \vdash 0$, so $t \vdash a[Q_i] \Rightarrow t'' \vdash a[Q_i''] \stackrel{\overline{a}\langle {\scriptstyle `}Q_i' \rangle}{\Longrightarrow} t' \vdash 0$, and $(s, x \vdash 0)\, \mathcal{Y}_{({\scriptstyle `}P_i', {\scriptstyle `}Q_i') \oplus \mathcal{E};r}^-\, (t' \vdash 0)$, hence $(r, sx, P_i', t', Q_i') \in \mathcal{S}$. Also, $t \vdash run^*({\scriptstyle `}Q_i) \Rightarrow t \vdash Q_i \Rightarrow t' \vdash Q_i'$ and we are done.

– Run
- $s \vdash run({\scriptstyle `}C_{1p}^-[\widetilde{P}]) \to s \vdash C_{1p}^-[\widetilde{P}]$
  We have $s \vdash run({\scriptstyle `}C_{1p}^-[\widetilde{P}]) \to s \vdash C_{1p}^-[\widetilde{P}]$ and $C_{1p}^-[\widetilde{P}] \leq C[\widetilde{P}^+]$. We still have $C_q^-[\widetilde{Q}] \leq C[\widetilde{Q}^+]$, so $(r, s, P', t, Q) \in \mathcal{S}$ and we are done.
- $[\ ]_i$
  We have $(s \vdash a[P_i])\, \mathcal{Y}_{\mathcal{E};r}^-\, (t \vdash a[Q_i])$ and $s \vdash a[P_i] \to s \vdash a[P_i'] \xrightarrow{\overline{a}\langle {\scriptstyle `}P_i' \rangle} s \vdash 0$, so $t \vdash a[Q_i] \Rightarrow t'' \vdash a[Q_i''] \stackrel{\overline{a}\langle {\scriptstyle `}Q_i' \rangle}{\Longrightarrow} t' \vdash 0$, and $(s \vdash 0)\, \mathcal{Y}_{({\scriptstyle `}P_i', {\scriptstyle `}Q_i') \oplus \mathcal{E};r}^-\, (t' \vdash 0)$, hence $(r, s, P_i', t', Q_i') \in \mathcal{S}$. Also, $t \vdash run^*({\scriptstyle `}Q_i) \Rightarrow t \vdash Q_i \Rightarrow t' \vdash Q_i'$ and we are done.

– Transp
- $s \vdash a[C_{1p}^-[\widetilde{P}]] \to s' \vdash a[R]$
  We have $s \vdash C_{1p}^-[\widetilde{P}] \to s' \vdash R$, so, by the induction hypothesis $t \vdash C_{1q}^-[\widetilde{Q}] \Rightarrow t' \vdash S$ and $(r, s', t', R, S) \in \mathcal{S}$. Therefore $t \vdash a[C_{1q}^-[\widetilde{Q}]] \Rightarrow t' \vdash a[S]$, and $(r, s', a[R], t', a[S]) \in \mathcal{S}$ since the fresh names $(t' \setminus t)$ can be guaranteed different from a.
- $[\ ]_i$
  We have $(s \vdash a[P_i])\ \mathcal{Y}_{\mathcal{E};r}^-\ (t \vdash a[Q_i])$, $s \vdash a[P_i] \to s' \vdash a[P_i'] \xrightarrow{\overline{a}\langle {\scriptstyle `}P_i' \rangle} s' \vdash 0$. So, $t \vdash a[Q_i] \Rightarrow t'' \vdash a[Q_i''] \stackrel{\overline{a}\langle {\scriptstyle `}Q_i' \rangle}{\Longrightarrow} t' \vdash 0$, and $(s' \vdash 0)\, \mathcal{Y}_{({\scriptstyle `}P_i', {\scriptstyle `}Q_i') \oplus \mathcal{E};r}^-\, (t' \vdash 0)$,

hence $(r, s', a[P_i'], t', a[Q_i']) \in \mathcal{S}$ since a can be guaranteed different from $(s' \setminus s)$ and $t' \setminus t$. Also, $t \vdash run^*(\text{`}Q_i) \Rightarrow t \vdash Q_i \Rightarrow t' \vdash Q_i'$ and we are done. Again, this subcase always holds similarly, and thus it is not repeated below.

- Par-L
  $s \vdash C_p^-[\widetilde{P}] = s \vdash C_{1p}^-[\widetilde{P}] \mid C_{2p}^-[\widetilde{P}] \to s' \vdash R \mid C_{2p}^-[\widetilde{P}]$, i.e. $s \vdash C_{1p}^-[\widetilde{P}] \to s' \vdash R$. So, by the induction hypothesis, $t \vdash C_{1q}^-[\widetilde{Q}] \Rightarrow t' \vdash S$ and $(r, s', R, t', S) \in \mathcal{S}$. Therefore, $t \vdash run^*\text{`}(C_q^-[\widetilde{P}]) \Rightarrow t' \vdash S \mid C_{2q}^-[\widetilde{Q}]$, and also $(r, s', R \mid C_{2p}^-[\widetilde{P}], t', S \mid C_{2q}^-[\widetilde{Q}]) \in \mathcal{S}$.
- Par-R
  Similarly
- Rep
  $s \vdash C_p^-[\widetilde{P}] = s \vdash !C_{1p}^-[\widetilde{P}] \to s' \vdash R$, hence $s \vdash !C_{1p}^-[\widetilde{P}] \mid C_{1p}^-[\widetilde{P}] \to s' \vdash R$. By the induction hypothesis, $t \vdash !C_{1q}^-[\widetilde{Q}] \mid C_{1q}^-[\widetilde{Q}] \Rightarrow t' \vdash S$ with $(r, s', R, t', S) \in \mathcal{S}$, hence, $t \vdash !C_{1q}^-[\widetilde{Q}] \Rightarrow t' \vdash S$, $t \vdash C_q^-[\widetilde{Q}] = t \vdash run^*\text{`}(!C_{1q}^-[\widetilde{Q}]) \Rightarrow t' \vdash S$, and still $(r, s', R, t', S) \in \mathcal{S}$.
- React-L
  There are several subcases.
  - The two contexts react
    $s \vdash C_p^-[\widetilde{P}] = s \vdash C_{1p}^-[\widetilde{P}_0, \widetilde{P}_1] \mid C_{2p}^-[\widetilde{P}_2] \to s \vdash C_{1p}'^-[\widetilde{P}_0] \mid C_{2p}'^-[\widetilde{P}_2, \widetilde{P}_1]$. Of course, C_{1q}^-can (weakly) do the same reaction, giving $t \vdash C_q^-[\widetilde{Q}] \Rightarrow t \vdash C_{1q}'^-[\widetilde{Q}_0] \mid C_{2q}'^-[\widetilde{Q}_2, \widetilde{Q}_1]$ and as expected $(r, s, C_{1p}'^-[\widetilde{P}_0] \mid C_{2p}'^-[\widetilde{P}_2, \widetilde{P}_1], t, C_{1q}'^-[\widetilde{Q}_0] \mid C_{2q}'^-[\widetilde{Q}_2, \widetilde{Q}_1]) \in \mathcal{S}$.
  - $C_{1p}^-[\widetilde{P}_1]$ sends, P_i in $C_{2p}^-[\widetilde{P}_2, P_i]$ receives.
    $s \vdash C_p^-[\widetilde{P}] = s \vdash C_{1p}^-[\widetilde{P}_1] \mid C_{2p}^-[\widetilde{P}_2, P_i] \to s \vdash C_{1p}'^-[\widetilde{P}_1] \mid C_{2p}'^-[\widetilde{P}_2, P_i']$. We know that $C_{1q}^-[\widetilde{Q}]$ can weakly do the same output transition on some channel o. Also, we have $(s \vdash a[P_i]) \, \mathcal{Y}_{\mathcal{E};r}^- \, (t \vdash a[Q_i])$ and $s \vdash a[P_i] \xrightarrow{o} s \vdash a[P_i'] \xrightarrow{\overline{a}\langle \text{`}P_i' \rangle}$ $s \vdash 0$ for the same channel $o \in r$. So, $t \vdash a[Q_i] \stackrel{o}{\Rightarrow} t'' \vdash a[Q_i''] \stackrel{\overline{a}\langle \text{`}Q_i' \rangle}{\Longrightarrow} t' \vdash 0$ with $(s \vdash 0) \, \mathcal{Y}_{(\text{`}P_i', \text{`}Q_i') \oplus \mathcal{E};r}^- \, (t' \vdash 0)$, hence $t \vdash C_q^-[\widetilde{Q}] \Rightarrow t' \vdash C_{1q}'^-[\widetilde{Q}_1] \mid C_{2q}'^-[\widetilde{Q}_2, Q_i']$ since the input can be done by Q_i and since Q_i is (weakly) in redex position, and $(r, s, C_{1p}'^-[\widetilde{P}_1] \mid C_{2p}^-[\widetilde{P}_2, P_i'], t', C_{1q}'^-[\widetilde{Q}_1] \mid C_{2q}'^-[\widetilde{Q}_2, Q_i']) \in \mathcal{S}$.
  - $C_{2p}^-[\widetilde{P}_2]$ receives, P_i in $C_{1p}^-[\widetilde{P}_1, P_i]$ sends
    $s \vdash C_p^-[\widetilde{P}] = s \vdash C_{1p}^-[\widetilde{P}_1, P_i] \mid C_{2p}^-[\widetilde{P}_2] \to s \vdash C_{1p}^-[\widetilde{P}_1, P_i'] \mid C_{2p}'^-[\widetilde{P}_2, P_j]$. We know that $C_{2q}^-[\widetilde{Q}_2]$ can weakly do the same input transition on some channel o. Also, we have $s \vdash (a[P_i]) \mathcal{Y}_{\mathcal{E};r}^- (t \vdash a[Q_i])$ and $s \vdash a[P_i] \xrightarrow{\overline{o}\langle \text{`}P_j \rangle} s \vdash a[P_i'] \xrightarrow{\overline{a}\langle \text{`}P_i' \rangle} s \vdash 0$, so, $t \vdash a[Q_i] \stackrel{\overline{o}\langle \text{`}Q_j \rangle}{\Longrightarrow} t'' \vdash a[Q_i''] \stackrel{\overline{a}\langle \text{`}Q_i' \rangle}{\Longrightarrow} t' \vdash 0$ with $(s \vdash 0) \mathcal{Y}_{(\text{`}P_j, \text{`}Q_j) \oplus \mathcal{E};r}^- (t' \vdash 0)$, so $t \vdash C_q^-[\widetilde{Q}] \Rightarrow t' \vdash C_{1q}'^-[\widetilde{Q}_1, Q_i'] \mid C_{2q}'^-[\widetilde{Q}_2, Q_j]$, and $(r, s, C_{1p}^-[\widetilde{P}_1, P_i'] \mid C_{2p}'^-[\widetilde{P}_2, P_j], t', C_{1q}'^-[\widetilde{Q}_1, Q_i'] \mid C_{2q}'^-[\widetilde{Q}_2, Q_j]) \in \mathcal{S}$.

- P_i in $C^-_{1p}[\widetilde{P}_1, P_i]$ and P_j in $C^-_{2p}[\widetilde{P}_2, P_j]$ react
  $s \vdash C^-_p[\widetilde{P}] = s \vdash C^-_{1p}[\widetilde{P}_1, P_i] \mid C^-_{2p}[\widetilde{P}_2, P_j] \to s \vdash C^-_{1p}[\widetilde{P}_1, P'_i] \mid C^-_{2p}[\widetilde{P}_2, P'_j]$.
  We have $(s, a, b \vdash a[P_i] \mid b[P_j])\, \mathcal{Y}^-_{\mathcal{E};rab}\, (t, a, b \vdash a[Q_i] \mid b[Q_j])$ and $s, a, b \vdash a[P_i] \mid b[P_j] \to s, a, b \vdash a[P'_i] \mid b[P'_j] \xrightarrow{\overline{a}\langle \text{`}P'_i\rangle} \xrightarrow{\overline{b}\langle \text{`}P'_j\rangle} s, a, b \vdash 0$, so $t, a, b \vdash a[Q_i] \mid b[Q_j] \Rightarrow t'', a, b \vdash a[Q''_i] \mid b[Q''_j] \xRightarrow{\overline{a}\langle \text{`}Q'_i\rangle} \xRightarrow{\overline{b}\langle \text{`}Q'_j\rangle} t', a, b \vdash 0$ with $(s, a, b \vdash 0) \mathcal{Y}^-_{(\text{`}P'_j, \text{`}Q'_j) \oplus (\text{`}P'_i, \text{`}Q'_i) \oplus \mathcal{E};rab} (t', a, b \vdash 0)$, hence $(s \vdash 0) \mathcal{Y}^-_{(\text{`}P'_j, \text{`}Q'_j) \oplus (\text{`}P'_i, \text{`}Q'_i) \oplus \mathcal{E};r}$ $(t' \vdash 0)$ up-to name creation. Therefore, $t \vdash C^-_q[\widetilde{Q}] \Rightarrow t' \vdash C'^-_{1q}[\widetilde{Q}_1, Q'_i] \mid C'^-_{2q}[\widetilde{Q}_2, Q'_j]$, and $(C^-_{1p}[\widetilde{P}_1, P'_i] | C^-_{2p}[\widetilde{P}_2, P'_j], t', C'^-_{1q}[\widetilde{Q}_1, Q'_i] | C'^-_{2q}[\widetilde{Q}_2, Q'_j]) \in \mathcal{S}$

– React-R
  Similarly

Corollary B.48. *For all P, Q, r and a such that $r \subseteq f = fn(P, Q)$, $a \notin f \setminus r$, and $(r, a \vdash \overline{a}\langle P\rangle)\, \mathcal{Y}_{\emptyset;ra}\, (r, a \vdash \overline{a}\langle Q\rangle)$, we have $f, a \vdash P \mathrel{\dot{\approx}}_{ra} r, a \vdash Q$.*

Proof. By $(f, a \vdash \overline{a}\langle P\rangle)\, \mathcal{Y}_{\emptyset;ra}\, (f, a \vdash \overline{a}\langle Q\rangle)$, we have $(f, a \vdash 0)\, \mathcal{Y}^\star_{\{(\text{`}P, \text{`}Q)\};ra}\, (f, a \vdash 0)$, hence $(r, s, P, t, Q) \in \mathcal{S}$ hence $f, a \vdash P \mathrel{\dot{\approx}}_{r,a} f, a \vdash Q$ by Lemma B.47.

4 Completeness of environmental bisimulation

Lemma B.49. *Run erasure of* $\approx$.

Proof. *We show that* $\{(r, s, P, t, Q) \mid P \leq P^+, Q \leq Q^+, s \vdash P^+ \approx_r t \vdash Q^+\} \subseteq \approx$.

- *Derived from Corollary B.23.*
- *Derived from Corollary B.23.*
- *Conversely.*
- *Let* $u = fn(R) \setminus r$. *We have* $s, u \vdash P^+ \mid R \approx_{r,u} t, u \vdash Q^+ \mid R$ *by definition of* $\approx$ *and* $P \mid R \leq P^+ \mid R$ *as well as* $Q \mid R \leq Q^+ \mid R$ *by definition of* $\leq$.

Lemma B.50. [Completeness and environmental bisimulation]
Consider the set

$$\begin{aligned}\mathcal{X} = \{(\mathcal{E}_x, r_x, s_x, P \mid a_1[P_1] \mid \ldots \mid a_n[P_n], t_x, Q \mid a_1[Q_1] \mid \ldots \mid a_n[Q_n]) \mid \\ ((\text{`}P_1, \ldots, \text{`}P_n), (\text{`}Q_1, \ldots, \text{`}Q_n)) \subseteq \mathcal{E}_x, \\ (\widetilde{\text{`}P_i}, \widetilde{\text{`}Q_i}) = \mathcal{E}, \\ \mathcal{E}_x \subseteq \mathcal{E}, \\ s \vdash P \mid \textstyle\prod_{i=1}^n !\overline{f_i}\langle \text{`}P_i\rangle \approx_r t \vdash Q \mid \prod_{i=1}^n !\overline{f_i}\langle \text{`}Q_i\rangle, \\ \widetilde{a}_i \subseteq r_x, \\ r = r_x \oplus \widetilde{f_i} \oplus e_x, (\oplus \textit{ meaning no overlap}) \\ s = s_x \oplus \widetilde{f_i} \oplus e_x, \\ t = t_x \oplus \widetilde{f_i} \oplus e_x, \\ r_x \subseteq (s_x \cap t_x)\}.\end{aligned}$$

We show that $\mathcal{X}$ *is an environmental bisimulation (up-to* $\equiv$*).*

Proof. We check each clause of environmental bisimulation (up-to context) against $\mathcal{X}$. We may just write $\prod$ for both $\prod_{i=1}^n !\overline{f_i}\langle \text{`}P_i\rangle$ and $\prod_{i=1}^n !\overline{f_i}\langle \text{`}Q_i\rangle$, and $\prod_{n+1}$ when there is an element added to the product. Context makes clear what is considered.

- Reduction
  1. $s_x \vdash P \mid a_1[P_1] \mid \ldots \mid a_n[P_n] \xrightarrow{\tau} s'_x \vdash P' \mid a_1[P_1] \mid \ldots \mid a_n[P_n]$.
     We have $s \vdash P \mid \prod \approx_r t \vdash Q \mid \prod$ and $s \vdash P \mid \prod \xrightarrow{\tau} s' \vdash P' \mid \prod$ with $s'_x = s_x \cup (s' \setminus s)$.
     So, by clause 1 of Definition B.5, for some t' and Q', we have $t \vdash Q \mid \prod \stackrel{\tau}{\Rightarrow} t' \vdash Q' \mid \prod$ and $s' \vdash P' \mid \prod \approx_r t' \vdash Q' \mid \prod$ since the other subprocesses in $\prod$ are guarded by a "fresh" guard f_i and thus cannot reduce nor react with anything.
     Therefore, $t \vdash Q \stackrel{\tau}{\Rightarrow} t' \vdash Q'$, hence $t_x \vdash Q \mid a_1[Q_1] \mid \ldots \mid a_n[Q_n]) \stackrel{\tau}{\Rightarrow} t'_x \vdash Q' \mid a_1[Q_1] \mid \ldots \mid a_n[Q_n]$ for $t'_x = t_x \cup (t' \setminus t)$. Indeed, since created names could not be free in $\mathcal{E}$ by definition of the LTS (Par-L, Par-R), they are not free in $\mathcal{E}_x$, and by $r_x \subseteq (s_x \cap t_x) \subseteq (s \cap t)$ we know that they will not clash with r_x either. We will henceforth assume this implicitely. Finally, $(\mathcal{E}_x, r_x, s'_x, P \mid a_1[P_1] \mid \ldots \mid a_n[P_n], t'_x, Q' \mid a_1[Q_1] \mid \ldots \mid a_n[Q_n]) \in \mathcal{X}$.

2. $P_i \xrightarrow{\tau} P_i'$.
   $s \vdash P \mid \prod \approx_r t \vdash Q \mid \prod$ so, for some fresh g and for $r' = r + g$, by clause 4 of Definition B.5 we have $s, g \vdash P \mid \prod \mid f_i(X).g[runX] \approx_{r'} t, g \vdash Q \mid \prod \mid f_i(X).g[runX]$.
   Then, we can have a reaction between a copy of $f_i\langle `P_i\rangle$ drawn from $\prod$, and $f_i(X)$, giving $s, g \vdash P \mid \prod \mid f_i(X).g[runX] \xrightarrow{\tau} s, g \vdash P \mid \prod \mid g[run`P_i]$. By clause 1 of Definition B.5, we thus have t_a, Q_a and Q_{ia} such that, by clause 2 of Definition B.5 and the fact that there is no input barb on unique f_i in LHS as well as the fact that there is an output barb on unique g, we have $t, g \vdash Q \mid \prod \mid f_i(X).g[runX] \stackrel{\tau}{\Rightarrow} t_a, g \vdash Q_a \mid \prod \mid g[Q_{ia}]$ and $s, g \vdash P \mid \prod \mid g[run`P_i] \approx_{r'} t_a, g \vdash Q_a \mid \prod \mid g[Q_{ia}]$.
   LHS can now do a *run*-transition $s, g \vdash P \mid \prod \mid g[run`P_i] \xrightarrow{\tau} s, g \vdash P \mid \prod \mid g[P_i]$ and thus, by clause 1 of Definition B.5, we have t_b, Q_b and Q_{ib} such that $t_a, g \vdash Q_a \mid \prod \mid g[Q_{ia}] \stackrel{\tau}{\Rightarrow} t_b, g \vdash Q_b \mid \prod \mid g[Q_{ib}]$ and $s, g \vdash P \mid \prod \mid g[P_i] \approx_{r'} t_b, g \vdash Q_b \mid \prod \mid g[Q_{ib}]$
   Then, we can now do the transition that corresponds to $P_i \xrightarrow{\tau} P_i'$: $s, g \vdash P \mid \prod \mid g[P_i] \xrightarrow{\tau} s', g \vdash P \mid \prod \mid g[P_i']$ and by clause 1 of Definition B.5 (and 2 on barb g), we have $t_b, g \vdash Qb \mid \prod \mid g[Q_{ib}] \stackrel{\tau}{\Rightarrow} t_c, g \vdash Qc \mid \prod \mid g[Q_{ic}]$ and $s', g \vdash P \mid \prod \mid g[P_i'] \approx_{r'} t_c, g \vdash Qc \mid \prod \mid g[Q_{ic}]$.
   Now, we can use clause 4 of Definition B.5 with known name g and fresh name f_{n+1} and have, for $e_x = g + f_{n+1}$ and $r'' = r + e_x$, $s', e_x \vdash P \mid \prod \mid g[P_i'] \mid g(X).!\overline{f_{n+1}}\langle X\rangle \approx_{r''} t_c, e_x \vdash Q_c \mid \prod \mid g[Q_{ic}] \mid g(X).!\overline{f_{n+1}}\langle X\rangle$.
   We can have a reaction between $g[\ ]$ and $g(\ )$: $s', e_x \vdash P \mid \prod \mid g[P_i'] \mid g(X).!\overline{f_{n+1}}\langle X\rangle \xrightarrow{\tau} s', e_x \vdash P \mid \prod \mid !\overline{f_{n+1}}\langle `P_i'\rangle$, hence by clause 1 of Definition B.5, as well as clause 3 on barbs g and clause 2 on barb f_{n+1}, we have $t_c, e_x \vdash Q_c \mid \prod \mid g[Q_{ic}] \mid g(X).!\overline{f_{n+1}}\langle X\rangle \stackrel{\tau}{\Rightarrow} t', e_x \vdash Q' \mid \prod \mid !\overline{f_{n+1}}\langle `Q_{id}\rangle$ with $Q_{id} = Q_i'$ or $run`Q_i$, as well as $s', e_x \vdash P \mid \prod \mid !\overline{f_{n+1}}\langle `P_i'\rangle \approx_{r''} t', e_x \vdash Q' \mid \prod \mid !\overline{f_{n+1}}\langle `Q_{id}\rangle$.
   If $Q_{id} = Q_i'$, then we have $s', e_x \vdash P \mid \prod_{n+1} \approx_{r''} t', e_x \vdash Q' \mid \prod_{n+1}$. If $Q_{id} = run`Q_i$, then $Q_i$ was not necessary in the reduction, and we have $s', e_x \vdash P \mid \prod_{n+1} \approx_{r''} t', e_x \vdash Q' \mid \prod_{n+1}$ by *run*-erasure of $\approx$, only turning $Q_{id} = run`Q_i$ into Q_i. This does not affect the transitions done by Q.
   As a result from all these transitions notwithstanding the *run*-erasure, we have that $t_x \vdash Q \mid a_1[Q_1] \mid \ldots \mid a_i[Q_i] \mid \ldots \mid a_n[Q_n] \stackrel{\tau}{\Rightarrow} t_x' \vdash Q' \mid a_1[Q_1] \mid \ldots \mid a_i[Q_i'] \mid \ldots \mid a_n[Q_n]$ and also $(\mathcal{E}_x, r_x, s_x', P \mid a_1[P_1] \mid \ldots \mid a_i[P_i'] \mid \ldots \mid a_n[P_n], t_x', Q' \mid a_1[Q_1] \mid \ldots \mid a_i[Q_i'] \mid \ldots \mid a_n[Q_n]) \in \mathcal{X}$.
3. P and P_i react.
   Similar to the above case, but with s instead of s' and s_x instead of s_x' (since no name is created by the reaction) and P' instead of P after the reaction.
   As a result from all these transitions, we have that $t_x \vdash Q \mid a_1[Q_1] \mid \ldots \mid a_i[Q_i] \mid \ldots \mid a_n[Q_n] \stackrel{\tau}{\Rightarrow} t_x' \vdash Q' \mid a_1[Q_1] \mid \ldots \mid a_i[Q_i'] \mid \ldots \mid a_n[Q_n]$ and also $(\mathcal{E}_x, r_x, s_x, P' \mid a_1[P_1] \mid \ldots \mid a_i[P_i'] \mid \ldots \mid a_n[P_n], t_x', Q' \mid a_1[Q_1] \mid \ldots \mid a_i[Q_i'] \mid \ldots \mid a_n[Q_n]) \in \mathcal{X}$
4. P_i and P_j react.

$s \vdash P|\prod \approx_r t \vdash Q|\prod$ so, for some fresh g and h, $e_x = g{+}h$ and $r' = r{+}e_x$, using clause 4 of Definition B.5, we have $s, e_x \vdash P \mid \prod |f_i(X).g[runX] \mid f_j(X).h[runX] \approx_{r'} t, e_x \vdash Q \mid \prod |f_i(X).g[runX] \mid f_j(X).h[runX]$.

We can have a reaction between (a copy of) $!f_i\langle\ \rangle$ and $f_i(\ )$: $s, e_x \vdash P \mid \prod |f_i(X).g[runX]|f_j(X).h[runX] \xrightarrow{\tau} s, e_x \vdash P|\prod |g[run`P_i]|f_j(X).h[runX]$. By clause 1 of Definition B.5, RHS will do a reduction too, and by clause 3 of Definition B.5 on f_i, g and 2 on f_j, $!\overline{f_i}\langle\ \rangle$ and $f_i(\ )$ will react too in RHS, while $!\overline{f_j}\langle\ \rangle$ and $f_j(\ )$ will stay. In other words: $t, e_x \vdash Q \mid \prod |f_i(X).g[runX] \mid f_j(X).h[runX] \stackrel{\tau}{\Rightarrow} t_a, e_x \vdash Q_a \mid \prod |g[Q_{ia}] \mid f_j(X).h[runX]$ with $s, e_x \vdash P \mid \prod |g[run`P_i] \mid f_j(X).h[runX] \approx_{r'} t_a, e_x \vdash Q_a \mid \prod |g[Q_{ia}] \mid f_j(X).h[runX]$.

Similarly, with the reaction $s, e_x \vdash P \mid \prod |g[run`P_i] \mid f_j(X).h[runX] \xrightarrow{\tau} s, e_x \vdash P \mid \prod |g[P_i] \mid h[run`P_j]$ between $!\overline{f_j}\langle\ \rangle$ and $f_j(\ )$, we have $t_a, e_x \vdash Q_a \mid \prod |g[Q_{ia}] \mid f_j(X).h[runX] \stackrel{\tau}{\Rightarrow} t_b, e_x \vdash Q_b \mid \prod |g[Q_{ib}] \mid h[Q_{jb}]$ with $s, e_x \vdash P \mid \prod |g[run`P_i] \mid h[run`P_j] \approx_{r'} t_b, e_x \vdash Q_b \mid \prod |g[Q_{ib}] \mid h[Q_{jb}]$.

Then, we can do 2 *run* transitions which will be weakly followed by RHS according to clause 1 of Definition B.5, while preserving $g[\ ]$ and $h[\ ]$ by their uniqueness and clause 3 of Definition B.5, and we obtain $s, e_x \vdash P \mid \prod |g[run`P_i] \mid h[run`P_j] \xrightarrow{run} \xrightarrow{run} \xrightarrow{\tau} s, e_x \vdash P \mid \prod |g[P_i] \mid h[P_j]$, $t_b, e_x \vdash Q_b \mid \prod |g[Q_{ib}] \mid h[Q_{jb}] \stackrel{\tau}{\Rightarrow} t_c, e_x \vdash Q_c \mid \prod |g[Q_{ic}] \mid h[Q_{jc}]$ and $s, e_x \vdash P \mid \prod |g[P_i] \mid h[P_j] \approx_{r'} t_c, e_x \vdash Q_c \mid \prod |g[Q_{ic}] \mid h[Q_{jc}]$.

It is now possible to mimick the reaction of P_i and P_j in $\mathcal{X}$: $s, e_x \vdash P \mid \prod |g[P_i] \mid h[P_j] \xrightarrow{\tau} s, e_x \vdash P \mid \prod |g[P'_i] \mid h[P'_j]$ which is matched by $t_c, e_x \vdash Q_c \mid \prod |g[Q_{ic}] \mid h[Q_{jc}] \stackrel{\tau}{\Rightarrow} t_d, e_x \vdash Q_d \mid \prod |g[Q_{id}] \mid h[Q_{jd}]$ such that $s, e_x \vdash P \mid \prod |g[P'_i] \mid h[P'_j] \approx_{r'} t_d, e_x \vdash Q_d \mid \prod |g[Q_{id}] \mid h[Q_{jd}]$.

Using clause 4 of Definition B.5, we can now spawn another process with fresh names f_{n+1} and f_{n+2} and have, for $e'_x = e_x + f_{n+1} + f_{n+2}$ and $r'' = r{+}e'_x$: $s, e'_x \vdash P|\prod |g[P'_i]|h[P'_j]|g(X).!\overline{f_{n+1}}\langle X\rangle|h(X).!\overline{f_{n+2}}\langle X\rangle \approx_{r''} t_d, e'_x \vdash Q_d \mid \prod |g[Q_{id}] \mid h[Q_{jd}] \mid g(X).!\overline{f_{n+1}}\langle X\rangle \mid h(X).!\overline{f_{n+2}}\langle X\rangle$.

We can now do a reaction between $g[\ ]$ and $g(\ )$, which will be followed (by clause 1), consuming $g[\ ]$ and $g(\ )$too but leaving $h[\ ]$ and $h(\ )$ (by clause 3 and uniqueness of each barb on g and h), giving $s, e'_x \vdash P \mid \prod |g[P'_i] \mid h[P'_j] \mid g(X).!\overline{f_{n+1}}\langle X\rangle \mid h(X).!\overline{f_{n+2}}\langle X\rangle \xrightarrow{\tau} s, e'_x \vdash P \mid \prod |h[P'_j] \mid !\overline{f_{n+1}}\langle `P'_i\rangle \mid h(X).!\overline{f_{n+2}}\langle X\rangle$ as well as $t_d, e'_x \vdash Q_d \mid \prod |g[Q_{id}] \mid h[Q_{jd}] \mid g(X).!\overline{f_{n+1}}\langle X\rangle \mid h(X).!\overline{f_{n+2}}\langle X\rangle \stackrel{\tau}{\Rightarrow} t_e, e'_x \vdash Qe \mid \prod |h[Q_{ie}] \mid !\overline{f_{n+1}}\langle `Q_{ie}\rangle \mid h(X).!\overline{f_{n+2}}\langle X\rangle$ and $s, e'_x \vdash P|\prod |h[P'_j]|!\overline{f_{n+1}}\langle `P'_i\rangle|h(X).!\overline{f_{n+2}}\langle X\rangle \approx_{r''} t_e, e'_x \vdash Qe|\prod |h[Q_{ie}]| !\overline{f_{n+1}}\langle `Q_{ie}\rangle \mid h(X).!\overline{f_{n+2}}\langle X\rangle$.

Similarly, we can have and follow a reaction between $!h[\ ]$ and $h(\ )$ and have $s, e'_x \vdash P \mid \prod |!\overline{f_{n+1}}\langle `P'_i\rangle \mid !\overline{f_{n+2}}\langle `P'_j\rangle \approx_{r''} t', e'_x \vdash Q' \mid \prod |!\overline{f_{n+1}}\langle `Q_{ie}\rangle \mid !\overline{f_{n+2}}\langle `Q_{ie}\rangle$.

As in the previous cases, we have that Q_{ie} (resp. Q_{ie}) is either $run`Q_i$ or $Q'_i$ (resp. $run`Q_j$ or Q'_j) and we can use *run*-erasure if necessary without interfering with the transitions.

As a result from all these transitions, we have that $t_x \vdash Q \mid a_1[Q_1] \mid \ldots \mid a_i[Q_i] \mid a_j[Q_j] \mid \ldots \mid a_n[Q_n] \stackrel{\tau}{\Rightarrow} t'_x \vdash Q' \mid a_1[Q_1] \mid \ldots \mid a_i[Q'_i] \mid a_j[Q'_j] \mid \ldots \mid a_n[Q_n]$ and also $(\mathcal{E}_x, r_x, s_x, P \mid a_1[P_1] \mid \ldots \mid a_i[P'_i] \mid a_j[P'_j] \mid \ldots \mid a_n[P_n], t'_x, Q' \mid a_1[Q_1] \mid \ldots \mid a_i[Q'_i] \mid a_j[Q'_j] \mid \ldots \mid a_n[Q_n]) \in \mathcal{X}$

5. P passivates $a[P_i]$.
   $s \vdash P \mid \prod \approx_r t \vdash Q \mid \prod$ so, for some fresh d, and by clause 4 of Definition B.5 $s, d \vdash P \mid \prod \mid f_i(X).!\overline{a}\langle X\rangle.d \approx_{r+d} t, d \vdash Q \mid \prod \mid f_i(X).!\overline{a}\langle X\rangle.d$.
   Then, we can have a reaction between (a copy of) $!\overline{f_i}\langle\ \rangle$ and $f_i(\ )$: $s, d \vdash P \mid \prod \mid f_i(X).!\overline{a}\langle X\rangle.d \xrightarrow{\tau} s, d \vdash P \mid \prod \mid !\overline{a}\langle `P_i\rangle.d$. By clause 1 of Definition B.5, RHS can weakly follow: $t, d \vdash Q \mid \prod \mid f_i(X).!\overline{a}\langle X\rangle.d \stackrel{\tau}{\Rightarrow} t_a, d \vdash Q_a \mid \prod \mid R_a$ for some $R_a$ (which can only be one of $!\overline{a}\langle `Q_i\rangle.d$, or d (since d is fresh and since clause 3 enforces we consumed $f_i(X)$)) and we have $s, d \vdash P \mid \prod \mid !\overline{a}\langle X\rangle.d \approx_{r+d} t_a, d \vdash Q_a \mid \prod \mid R_a$.
   Then, P can input P_i: $s, d \vdash P \mid \prod \mid !\overline{a}\langle X\rangle.d \xrightarrow{\tau} s, d \vdash P' \mid \prod \mid d$ and, weakly, we get $t_a, d \vdash Q_a \mid \prod \mid R_a \stackrel{\tau}{\Rightarrow} t_b, d \vdash Q_b \mid \prod \mid R_b$ and $s, d \vdash P' \mid \prod \mid d \approx_{r+d} t_b, d \vdash Q_b \mid \prod \mid R_b$ for some R_b which has still to be one of $!\overline{a}\langle `Q_i\rangle.d$ or d because of clause 2 of Definition B.5 on barb d.
   We can now spawn with clause 4 a process: $s, d \vdash P' \mid \prod \mid d \mid \overline{d} \approx_{r+d} t_b, d \vdash Q_b \mid \prod \mid R_b \mid \overline{d}$ and have a reaction: $s, d \vdash P \mid \prod \mid d \mid \overline{d} \xrightarrow{\tau} s, d \vdash P' \mid \prod$ By clause 1 of Definition B.5, we have $t_b, d \vdash Q_b \mid \prod \mid R_b \mid \overline{d} \stackrel{\tau}{\Rightarrow} t', d \vdash Q' \mid \prod$ and $s, d \vdash P' \mid \prod \approx_{r+d} t', d \vdash Q' \mid \prod$ since necessarily by clause 3 of Definition B.5, there must be no d left in RHS otherwise LHS could not exhibit the same barb.
   This means that Q has weakly been able to input Q_i on channel a at some point, as: $Q \stackrel{\tau}{\Rightarrow} \xrightarrow{a(`Q_i)} \stackrel{\tau}{\Rightarrow} Q'$. Therefore, we have $(\mathcal{E}_x, r_x, s_x, P' \mid a_1[P_1] \mid \ldots (\mid 0) \mid \ldots \mid a_n[P_n], t'_x, Q'' \mid a_1[Q_1] \mid \ldots (\mid 0) \mid \ldots \mid a_n[Q_n]) \in \mathcal{X}$.
6. P_j passivates $a_i[P_i]$.
   $s \vdash P \mid \prod \approx_r t \vdash Q \mid \prod$ so, for some fresh d and g, by clause 4 of Definition B.5, for $e_x = d + g$ and $r' = r + e_x$, we have $s, e_x \vdash P \mid \prod \mid f_i(X).!\overline{a}\langle X\rangle.d \mid f_j(X).g[runX] \approx_{r'} t, e_x \vdash Q \mid \prod \mid f_i(X).!\overline{a}\langle X\rangle.d \mid f_j(X).g[runX]$.
   Then, we can, as usual do a reaction between $!\overline{f_i}\langle\ \rangle$ and $f_i(\ )$ (which is followed by consuming $f_i(\ )$ too in RHS by clause 3 of Definition B.5, but does not consume $f_j(\ )$ for the same reasons), then a reaction between $!\overline{f_j}\langle\ \rangle$ and f_j which is followed, and we get $s, e_x \vdash P \mid \prod \mid f_i(X).!\overline{a}\langle X\rangle.d \mid f_j(X).g[runX] \xrightarrow{\tau} \xrightarrow{\tau} s, e_x \vdash P \mid \prod \mid !\overline{a}\langle `P_i\rangle.d \mid g[run `P_j], t, e_x \vdash Q \mid \prod \mid f_i(X).!\overline{a}\langle X\rangle.d \mid f_j(X).g[runX] \stackrel{\tau}{\Rightarrow} \stackrel{\tau}{\Rightarrow} t_a, e_x \vdash Q_a \mid \prod \mid R_a \mid g[Q_{ja}]$, and $s, e_x \vdash P \mid \prod \mid !\overline{a}\langle `P_i\rangle.d \mid g[run `P_j] \approx_{r'} t_a, e_x \vdash Q_a \mid \prod \mid R_a \mid g[Q_{ja}]$ for some R_a in $!\overline{a}\langle `Q_i\rangle.d$, d.
   Then, we can do a *run*-transition, weakly followed as usual, to get $s, e_x \vdash P \mid \prod \mid !\overline{a}\langle `P_i\rangle.d \mid g[run `P_j] \xrightarrow{run} \xrightarrow{\tau} s, e_x \vdash P \mid \prod \mid !\overline{a}\langle `P_i\rangle.d \mid g[P_j],\ t_a, e_x \vdash Q_a \mid \prod \mid R_a \mid g[Q_{ja}] \stackrel{\tau}{\Rightarrow} t_b, e_x \vdash Q_b \mid \prod \mid R_b \mid g[Q_{jb}]$, and $s, e_x \vdash P \mid \prod \mid !\overline{a}\langle `P_i\rangle.d \mid g[P_j] \approx_{r'} t_b, e_x \vdash Q_b \mid \prod \mid R_b \mid g[Q_{jb}]$ for some R_b in $!\overline{a}\langle `Q_i\rangle.d$, d.
   Then, we can have P_j input $`P_i$: $s, e_x \vdash P \mid \prod \mid !\overline{a}\langle `P_i\rangle.d \mid g[P_j] \xrightarrow{\tau} s, e_x \vdash P \mid \prod \mid d \mid g[P'_j]$, hence $t_b, e_x \vdash Q_b \mid \prod \mid R_b \mid g[Q_{jb}] \stackrel{\tau}{\Rightarrow} t_c, e_x \vdash Q_c \mid \prod \mid R_c \mid g[Q_{jc}]$

and $s, e_x \vdash P \mid \prod |d \mid g[P'_j] \approx_{r'} t_c, e_x \vdash Q_c \mid \prod |R_c \mid g[Q_{jc}]$ for some R_c in $!\overline{a}\langle `Q_i\rangle.d$, d because of clause 2 on barb d.

Then, we can remove the d by spawning $\overline{d}$ and causing a reaction: $s, e_x \vdash P \mid \prod |d \mid g[P'_j] \mid \overline{d} \approx_{r'} t_c, e_x \vdash Q_c \mid \prod |R_c \mid g[Q_{jc}] \mid \overline{d}$ and then $s, e_x \vdash P \mid \prod |d \mid g[P'_j] \mid \overline{d} \xrightarrow{\tau} s, e_x \vdash P \mid \prod |g[P'_j]$, hence $t_c, e_x \vdash Q_c \mid \prod |R_c \mid g[Q_{jc}] \mid \overline{d} \stackrel{\tau}{\Rightarrow} t_d, e_x \vdash Q_d \mid \prod |g[Q_{jd}]$, and $s, e_x \vdash P \mid \prod |g[P'_j] \approx_{r'} t_d, e_x \vdash Q_d \mid \prod |g[Q_{jd}]$ because RHS had to consume $!\overline{a}\langle `Q_i\rangle$ otherwise LHS could not exhibit barb d.

Now, we spawn another process with fresh f_{n+1}, and for $e'_x = e_x + f_{n+1}$ and $r'' = r + e'_x$ we have $s, e'_x \vdash P \mid \prod |g[P'_j] \mid g(X).!\overline{f_{n+1}}\langle X\rangle \approx_{r''} t_d, e'_x \vdash Q_d \mid \prod |g[Q_{jd}] \mid g(X).!\overline{f_{n+1}}\langle X\rangle$.

And we can do one last reaction which will be followed by the same reaction, and some internal reductions of Q_d: $s, e'_x \vdash P | \prod |g[P'_j] | g(X).!\overline{f_{n+1}}\langle X\rangle \xrightarrow{\tau} s, e'_x \vdash P \mid \prod |!\overline{f_{n+1}}\langle `P_j\rangle \ , \ t_d, e'_x \vdash Q_d \mid \prod |g[Q_{id}] \mid g(X).!\overline{f_{n+1}}\langle X\rangle \stackrel{\tau}{\Rightarrow} t', e'_x \vdash Q' \mid \prod |!\overline{f_{n+1}}\langle `Q_{ie}\rangle$ with $s, e'_x \vdash P \mid \prod |!\overline{f_{n+1}}\langle `P_j\rangle \approx_{r''} t', e'_x \vdash Q' \mid \prod |!\overline{f_{n+1}}\langle `Q_{ie}\rangle$ and Q_{ie} being either $run `Q_j$ or Q'_j.

There are two possible cases, either Q did the input at some time, or Q_i did it. If Q has done the passivation, as in the above case, proper transitions exist and membership to $\mathcal{X}$ is guaranteed (via erasure of $run `Q_j$'s *run* if necessary) If Q_j has done the passivation, then necessarily its outer *run* has been consumed, and there is no need for any erasure.

Again, this shows that the transition could be done by the member of $\mathcal{X}$, and that we still are in $\mathcal{X}$.

– Outputs

1. $P \xrightarrow{\overline{a}\langle M\rangle} P'$.

   so, for fresh g and f_{n+1}, using clause 4 of Definition B.5, for $e_x = g + f_{n+1}$ and $r' = r + e_x$, we have $s, e_x \vdash P \mid \prod |a(X).g.!\overline{f_{n+1}}\langle X\rangle \approx_{r'} t, e_x \vdash Q \mid \prod |a(X).g.!\overline{f_{n+1}}\langle X\rangle$.

   We can have a reaction that consumes $a(X)$: $s, e_x \vdash P | \prod |a(X).g.!\overline{f_{n+1}}\langle X\rangle \xrightarrow{\tau} s, e_x \vdash P' | \prod |g.!\overline{f_{n+1}}\langle M\rangle$, followed weakly: $t, e_x \vdash Q | \prod |a(X).g.!\overline{f_{n+1}}\langle X\rangle \stackrel{\tau}{\Rightarrow} t_a, e_x \vdash Q_a \mid \prod |R_a$ such that $s, e_x \vdash P' \mid \prod |g.!\overline{f_{n+1}}\langle M\rangle \approx_{r'} t_a, e_x \vdash Q_a \mid \prod |R_a$ with R_a one of $a(X).g.!\overline{f_{n+1}}\langle X\rangle$ or $g.!\overline{f_{n+1}}\langle X\rangle$ since clause 2 of Definition B.5 enforces that R_a has barb g.

   Then, we can spawn $\overline{g}$: $s, e_x \vdash P' | \prod |g.!\overline{f_{n+1}}\langle M\rangle | \overline{g} \approx_{r'} t_a, e_x \vdash Q_a | \prod |R_a | \overline{g}$ and have a reaction to remove g and $\overline{g}$: $s, e_x \vdash P' \mid \prod |g.!\overline{f_{n+1}}\langle M\rangle \mid \overline{g} \xrightarrow{\tau} s, e_x \vdash P' | \prod |!\overline{f_{n+1}}\langle M\rangle$ which is followed by $t_a, e_x \vdash Q_a | \prod |R_a | \overline{g} \stackrel{\tau}{\Rightarrow} t', e_x \vdash Q' \mid \prod |R'$ such that $s, e_x \vdash P' \mid \prod |!\overline{f_{n+1}}\langle M\rangle \approx_{r'} t', e_x \vdash Q' \mid \prod |R'$ with necessarily $R' = !\overline{f_{n+1}}\langle N\rangle$ since LHS has no barb g left.

   This means that Q has weakly been able to output N on channel a at some point, that is that $t_x \vdash Q \mid a_1[Q_1] \mid \ldots \mid a_n[Q_n] \stackrel{\overline{a}\langle N\rangle}{\Longrightarrow} t'_x \vdash Q' \mid a_1[Q_1] \mid \ldots \mid a_n[Q_n]$ and, as it happens, $((M, N) \oplus \mathcal{E}_x, r_x, s_x, P' \mid a_1[P_1] \mid \ldots \mid a_n[P_n], t'_x, Q' \mid a_1[Q_1] \mid \ldots \mid a_n[Q_n]) \in \mathcal{X}$ since (M, N) is now in $\mathcal{E}$.
2. Passivation of $a[P_i]$.

   Immediate.
3. $P_i \xrightarrow{\overline{a}\langle M\rangle} P'_i$.

For fresh g, using clause 4 of Definition B.5, we have $s, g \vdash P | \prod |f_i(X).g[runX] \approx_{r+g} t, g \vdash Q \mid \prod |f_i(X).g[runX]$.

We can do a reaction to consume $f_i(\,)$ and get $s, g \vdash P | \prod |f_i(X).g[runX] \xrightarrow{\tau} s, g \vdash P \mid \prod |g[run`P_i]$, hence $t, g \vdash Q \mid \prod |f_i(X).g[runX] \stackrel{\tau}{\Rightarrow} t_a, g \vdash Q_a \mid \prod |g[Q_{ia}]$ and $s, g \vdash P \mid \prod |g[run`P_i] \approx_{r+g} t_a, g \vdash Q_a \mid \prod |g[Q_{ia}]$.

Then, we can remove the *run*: $s, g \vdash P \mid \prod |g[run`P_i] \xrightarrow{run} \xrightarrow{\tau} s, g \vdash P \mid \prod |g[P_i]$, hence $t_a, g \vdash Q_a \mid \prod |g[Q_{ia}] \stackrel{\tau}{\Rightarrow} t_b, g \vdash Q_b \mid \prod |g[Q_{ib}]$ and $s, g \vdash P \mid \prod |g[P_i] \approx_{r+g} \stackrel{\tau}{\Rightarrow} t_b, g \vdash Q_b \mid \prod |g[Q_{ib}]$.

We can now spawn a receiver on a with fresh names f_{n+1} and f_{n+2}. For $e_x = g + f_{n+1} + f_{n+2}$ and $r' = r + e_x$, we have: $s, e_x \vdash P \mid \prod |g[P_i] \mid a(X).g(Y)(!\overline{f_{n+1}}\langle X\rangle \mid !\overline{f_{n+2}}\langle Y\rangle) \approx_{r'} t_b, e_x \vdash Q_b \mid \prod |g[Q_{ib}] \mid a(X).g(Y)(!\overline{f_{n+1}}\langle X\rangle \mid !\overline{f_{n+2}}\langle Y\rangle)$ for $r' = r + e_x$, and have LHS react on $a(X)$: $s, e_x \vdash P \mid \prod |g[P_i] \mid a(X).g(Y)(!\overline{f_{n+1}}\langle X\rangle \mid f_{n+2}\langle Y\rangle) \xrightarrow{\tau} s, e_x \vdash P \mid \prod |g[P_i'] \mid g(Y)(!\overline{f_{n+1}}\langle M\rangle \mid !\overline{f_{n+2}}\langle Y\rangle)$ hence $t_b, e_x \vdash Q_b \mid \prod |g[Q_{ib}] \mid a(X).g(Y)(!\overline{f_{n+1}}\langle X\rangle \mid !\overline{f_{n+2}}\langle Y\rangle) \stackrel{\tau}{\Rightarrow} t_c, e_x \vdash Q_c \mid \prod |g[Q_{ib}] \mid R_c$ and $s, e_x \vdash P \mid \prod |g[P_i'] \mid g(Y)(!\overline{f_{n+1}}\langle M\rangle \mid !\overline{f_{n+2}}\langle Y\rangle) \approx_{r'} t_c, e_x \vdash Q_c \mid \prod |g[Q_{ib}] \mid R_c$ with R_c in $a(X).g(Y).(!\overline{f_{n+1}}\langle X\rangle | !\overline{f_{n+2}}\langle Y\rangle)$ or $g(Y).(!\overline{f_{n+1}}\langle N\rangle | !\overline{f_{n+2}}\langle Y\rangle)$ because LHS still has barb g.

Then, LHS can react on g: $s, e_x \vdash P | \prod |g[P_i'] | g(Y)(!\overline{f_{n+1}}\langle M\rangle | !\overline{f_{n+2}}\langle Y\rangle) \xrightarrow{\tau} s, e_x \vdash P \mid \prod |!\overline{f_{n+1}}\langle M\rangle \mid !\overline{f_{n+2}}\langle `P_i'\rangle$, hence $t_c, e_x \vdash Q_c \mid \prod |g[Q_{ib}] \mid R_c \stackrel{\tau}{\Rightarrow} t', e_x \vdash Q' \mid \prod |!\overline{f_{n+1}}\langle N\rangle \mid !\overline{f_{n+2}}\langle `Q_{ic}\rangle$ and $s, e_x \vdash P \mid \prod |!\overline{f_{n+1}}\langle M\rangle \mid !\overline{f_{n+2}}\langle `P_i'\rangle \approx_{r'} t', e_x \vdash Q' \mid \prod |!\overline{f_{n+1}}\langle N\rangle \mid !\overline{f_{n+2}}\langle `Q_{ic}\rangle$ since LHS has no more barb on g, which enforced the reaction with $a(X)$ in R_c or its 'parent' at some point.

Again, Q_{ic} is either $run`Q_i$ or some $Q_i'$. If it is $run`Q_i$, then we can use *run*-erasure of $\approx$ and we are done since it was not used in the transitions. If not, then its outer has been consumed, and we're done.

Again, this shows that the transition could be followed by RHS of the member of $\mathcal{X}$, and that we still are in $\mathcal{X}$.

– Inputs

1. $P \xrightarrow{a(M)} P'$.

   For some fresh d, and some O such that $fn(O) \subseteq r_x, O\{`P_a/X_a, \ldots, `P_b/X_b\} = M$, we have $s, d \vdash P \mid \prod |f_a(X_a).\ldots.f_b(X_b).\overline{a}\langle O\rangle.d \approx_{r+d} t, d \vdash Q \mid \prod |f_a(X_a).\ldots.f_b(X_b).\overline{a}\langle O\rangle.d$.

   We can do all the reactions on $f_a(\,), \ldots, f_b(\,)$, and they'll be followed by clauses 2 and 3 on barbs, hence we have $s, d \vdash P | \prod |f_a(X_a).\ldots.f_b(X_b).\overline{a}\langle O\rangle.d \xrightarrow{\tau} \ldots \xrightarrow{\tau} s, d \vdash P \mid \prod |\overline{a}\langle M\rangle.d \; t, d \vdash Q \mid \prod |f_a(X_a).\ldots.f_b(X_b).\overline{a}\langle O\rangle.d \stackrel{\tau}{\Rightarrow} \ldots \stackrel{\tau}{\Rightarrow} t_a, d \vdash Q_a \mid \prod |R_a$, $s, d \vdash P \mid \prod |\overline{a}\langle M\rangle.d \approx_{r+d} t_a, d \vdash Q_a \mid \prod |R_a$ with R_a being $\overline{a}\langle N\rangle.d$ or d by clause 2 on barb d and 3 on f_b.

   Then, LHS can react on $\overline{a}\langle M\rangle$: $s, d \vdash P | \prod |\overline{a}\langle M\rangle.d \xrightarrow{\tau} s, d \vdash P' | \prod |d$ hence $t_a, d \vdash Q_a | \prod |R_a \stackrel{\tau}{\Rightarrow} t_b, d \vdash Q_b | \prod |R_b$, $s, d \vdash P' | \prod |d \approx_{r+d} t_b, d \vdash Q_b | \prod |R_b$ with R_b being $\overline{a}\langle N\rangle.d$ or d by clause 2 on barb d.

   By clause 4 of Definition B.5, we can spawn $\overline{d}$: $s, d \vdash P' \mid \prod |d \mid \overline{d} \approx_{r+d} t_b, d \vdash Q_b \mid \prod |R_b \mid \overline{d}$ and have a reaction: $s, d \vdash P' \mid \prod |d \mid \overline{d} \xrightarrow{\tau} s, d \vdash P' \mid \prod$

hence $t_b, d \vdash Q_b \mid \prod \mid R_b \mid \overline{d} \stackrel{\tau}{\Rightarrow} t', d \vdash Q' \mid \prod$ with $s, d \vdash P' \mid \prod \approx_{r+d} t', d \vdash Q' \mid \prod$ by the clause 3 of Definition B.5 on barb d.
Again, this shows that the transition could be done by the member of $\mathcal{X}$, and that we still are in $\mathcal{X}$.

2. $P_i \xrightarrow{a(M)} P_i'$.
For some fresh d, and some O such that $fn(O) \subseteq r_x$, $O\{`P_a/X_a, \ldots, `P_b/X_b\} = M$, we have $s, d \vdash P \mid \prod \mid f_a(X_a).\ldots.f_b(X_b).\overline{a}\langle O\rangle.d \approx_{r+d} t, d \vdash Q \mid \prod \mid f_a(X_a).\ldots.f_b(X_b).\overline{a}\langle O\rangle.d$. We can do all the reactions on $f_a(\,), \ldots, f_b(\,)$, and they'll be followed by clauses 2 and 3 on barbs, hence we have $s, d \vdash P \mid \prod \mid f_a(X_a).\ldots.f_b(X_b).\overline{a}\langle O\rangle.d \xrightarrow{\tau} \ldots \xrightarrow{\tau} s, d \vdash P \mid \prod \mid \overline{a}\langle M\rangle.d$ and $t, d \vdash Q \mid \prod \mid f_a(X_a).\ldots.f_b(X_b).\overline{a}\langle O\rangle.d \stackrel{\tau}{\Rightarrow} \ldots \stackrel{\tau}{\Rightarrow} t_a, d \vdash Q_a \mid \prod \mid R_a$ and $s, d \vdash P \mid \prod \mid \overline{a}\langle M\rangle.d \approx_{r+d} t_a, d \vdash Q_a \mid \prod \mid R_a$ with R_a being $\overline{a}\langle N\rangle.d$ or d by clause 2 on barb d.
We can now spawn with clause 4 of Definition B.5 a process with free name g, and, for $e_x = d + g$ and $r' = r + e_x$, erase the *run*'s from $s, e_x \vdash P \mid \prod \mid \overline{a}\langle M\rangle.d \mid f_i(X).g[runX] \approx_{r'} t_a, e_x \vdash Q_a \mid \prod \mid R_a \mid f_i(X).g[runX]$ to get $s, e_x \vdash P \mid \prod \mid \overline{a}\langle M\rangle.d \mid g[P_i] \approx_{r'} t_{aa}, e_x \vdash Q_{aa} \mid \prod \mid R_{aa} \mid g[Q_{ia}]$ by clause 3 of Definition B.5 on barb f_i and 2 on g.
Then, LHS can react on $\overline{a}\langle M\rangle$: $s, e_x \vdash P \mid \prod \mid \overline{a}\langle M\rangle.d \mid g[P_i] \xrightarrow{\tau} s, e_x \vdash P \mid \prod \mid d \mid g[P_i']$ hence $t_{aa}, e_x \vdash Q_a a \mid \prod \mid R_a a \mid g[Q_{iaa}] \stackrel{\tau}{\Rightarrow} t_b, e_x \vdash Q_b \mid \prod \mid R_b \mid g[Q_{ib}]$, and $s, e_x \vdash P \mid \prod \mid d \mid g[P_i'] \approx_{r'} t_b, e_x \vdash Q_b \mid \prod \mid R_b \mid g[Q_{ib}]$ with R_b being $\overline{a}\langle N\rangle.d$ or d by (now) clause 2 (too) on barb d.
By clause 4 of Definition B.5, we can spawn $\overline{d}$: $s, e_x \vdash P \mid \prod \mid d \mid g[P_i'] \mid \overline{d} \approx_{r'} t_b, e_x \vdash Q_b \mid \prod \mid R_b \mid g[Q_{ib}] \mid \overline{d}$ and have a reaction: $s, e_x \vdash P \mid \prod \mid d \mid \overline{d} \mid g[P_i'] \xrightarrow{\tau} s, e_x \vdash P \mid \prod \mid g[P_i']$ hence $t_b, e_x \vdash Q_b \mid \prod \mid R_b \mid g[Q_{ib}] \mid \overline{d} \stackrel{\tau}{\Rightarrow} t_c, e_x \vdash Q_c \mid \prod \mid g[Q_{ic}]$ with $s, e_x \vdash P \mid \prod \mid g[P_i'] \approx_{r'} t_c, e_x \vdash Q_c \mid \prod \mid g[Q_{ic}]$ by the clause 3 of Definition B.5 on barb d, which enforced the reaction on channel a in R_b (or its parent) in the RHS.
Finally, to put P_i' and Q_{ic} in the environment, we spawn a process with free name f_{n+1}, and for $e_x' = e_x + f_{n+1}$ and $r'' = r + e_x'$ we have: $s, e_x' \vdash P \mid \prod \mid g[P_i'] \mid g(X).!\overline{f_{n+1}}\langle X\rangle \approx_{r''} t_c, e_x' \vdash Q_c \mid \prod \mid g[Q_{ic}] \mid g(X).!\overline{f_{n+1}}\langle X\rangle$ and derive, as expected, $s, e_x' \vdash P \mid \prod_{n+1} \approx_{r''} t_c, e_x' \vdash Q_c \mid \prod_{n+1}$ with the right transitions, using the *run*-erasure if necessary.
Again, this shows that the transition could be done by the member of $\mathcal{X}$, and that we still are in $\mathcal{X}$.

– Spawn clause.
Immediate by definition of $\mathcal{X}$.
– Name creation.
We can add any name not in s, t to r (hence not in s_x, t_x, to r_x), so that they do not clash with other names, by choosing another r.
– Converses of 1, 2 and 3.
Similarly.

Corollary B.51. [Completeness of environmental bisimulation]
If $f \vdash P \approx_r f \vdash Q$ with $r \subseteq f = fn(P, Q)$, then $f \vdash P \sim_{\emptyset;r} f \vdash Q$.

Proof. From Lemma B.50.

Corollary B.52. [Completeness of environmental bisimulation w.r.t. reduction-closed barbed congruence]
If $f, a \vdash P \dot{\approx}_{ra} f, a \vdash Q$ *with* $r \subseteq f = fn(P, Q)$*, then* $f, a \vdash \overline{a}\langle P\rangle \sim_{\emptyset;ra} f, a \vdash \overline{a}\langle Q\rangle$.

Proof. By $f, a \vdash P \dot{\approx}_{ra} f, a \vdash Q$, we have $f, a \vdash \overline{a}\langle P\rangle \dot{\approx}_{ra} f, a \vdash \overline{a}\langle Q\rangle$, hence $f, a \vdash \overline{a}\langle P\rangle \approx_{ra} f, a \vdash \overline{a}\langle Q\rangle$, hence $f, a \vdash \overline{a}\langle P\rangle \sim_{\emptyset;ra} f, a \vdash \overline{a}\langle Q\rangle$ by Corollary B.51.

Definition B.53. *We write* $P \doteq_r Q$ *if* $(f \vdash 0) \sim_{\{(`P,`Q)\};r} (f \vdash 0)$ *with* $r \subseteq f = fn(P, Q)$.

Corollary B.54. [Characterisation of reduction-closed barbed congruence]
$P \doteq_r Q$ *if and only if* $f \vdash P \dot{\approx}_r f \vdash Q$ *with* $r \subseteq f = fn(P, Q)$.

Proof. ($\Rightarrow$): from B.47.
($\Leftarrow$): from B.52, and then by output to a.