Environmental Bisimulations for Higher-Order Languages

DAVIDE SANGIORGI
University of Bologna and INRIA
and
NAOKI KOBAYASHI and EIJIRO SUMII
Tohoku University

Developing a theory of bisimulation in higher-order languages can be hard. Particularly challenging can be: (1) the proof of congruence, as well as enhancements of the bisimulation proof method with "up-to context" techniques, and (2) obtaining definitions and results that scale to languages with different features.

To meet these challenges, we present *environment bisimulations*, a form of bisimulation for higher-order languages, and its basic theory. We consider four representative calculi: pure λ-calculi (call-by-name and call-by-value), call-by-value λ-calculus with higher-order store, and then Higher-Order π-calculus. In each case: we present the basic properties of environment bisimilarity, including congruence; we show that it coincides with contextual equivalence; we develop some up-to techniques, including up-to context, as examples of possible enhancements of the associated bisimulation method.

Unlike previous approaches (such as applicative bisimulations, logical relations, Sumii-Pierce-Koutavas-Wand), our method does not require induction/indices on evaluation derivation/steps (which may complicate the proofs of congruence, transitivity, and the combination with up-to techniques), or sophisticated methods such as Howe's for proving congruence. It also scales from the pure λ-calculi to the richer calculi with simple congruence proofs.

Categories and Subject Descriptors: D.3.1 [**Programming Languages**]: Formal Definitions and Theory; F.3.2 [**Logics and Meanings of Programs**]: Semantics of Programming Languages

General Terms: Theory, Verification

Additional Key Words and Phrases: Higher-order languages, bisimulation, congruence, λ-calculus, higher-order π-calculus

5

D. Sangiorgi was supported by the EU FET project 231620 "Hats". N. Kobayashi and E. Sumii were supported by Kakenhi 20240001 and 22300005.
A preliminary version has appeared, under the same name, in *Proceedings of the Annual IEEE Symposium on Logic in Computer Science (LICS'07)*
Author's address: D. Sangiorgi; email: davide.sangiorgi@cs.unibo.it.

DOI 10.1145/1889997.1890002 http://doi.acm.org/10.1145/1889997.1890002

ACM Reference Format:
Sangiorgi, D., Kobayashi, N., and Sumii, E. 2011. Environmental bisimulations for higher-order languages. ACM Trans. Program. Lang. Syst. 33, 1, Article 5 (January 2011), 69 pages.
DOI = 10.1145/1889997.1890002 http://doi.acm.org/10.1145/1889997.1890002

1. INTRODUCTION

Behavioral equivalence and bisimulation in higher-order languages Proving equivalence of computer programs is an important but challenging problem. Equivalence between two programs means that the programs should behave "in the same manner" under any context [Morris 1968]; this notion of equality is called *contextual equivalence*. Finding effective methods for equivalence proofs is particularly challenging in *higher-order* languages (i.e., languages where program code can be passed around, as opposed to *first-order* languages).

Bisimulation has emerged as a powerful operational method for proving equivalence of programs in various kinds of languages, due to the associated coinductive proof method. Further, a number of enhancements of the bisimulation method have been studied, usually called *up-to techniques*. To be useful, the behavioral relation resulting from bisimulation—*bisimilarity*—should be a *congruence*; and preferably it should coincide with contextual equivalence. For first-order languages, there is a common consensus about what bisimulation is and how it should be defined, and the associated proof techniques are well-developed (e.g., techniques for proving congruence, and up-to techniques).

The picture is less clear for higher-order languages, as available definitions and proof techniques are often difficult to adapt to different languages. We informally discuss some key design issues for bisimulation, and why previous proposals of bisimulations are not robust enough, using an abstract form of transitions for higher-order calculi, and occasionally referring to concrete calculi such as λ-calculi and Higher-Order π-calculus (HOπ). (The actual syntax for transitions in the calculi of the paper will be slightly different; for instance in HOπ we will use the "early" style, rather than the "late" one as below; further, below for simplicity we ignore possible extrusions of private channels.) Transitions can take three forms.

—*First-order transitions* $P \stackrel{\ell}{\rightarrow} P'$ (the label ℓ is first-order, that is, it does not contain terms). Reductions in pure λ-calculi and τ-transitions in HOπ are of this kind (in the former case, ℓ is omitted).

—*Higher-order output transitions* $P \stackrel{\ell}{\rightarrow} \langle P_1 \rangle P_2$, in which a higher-order value P_1 is produced and P_2 is the continuation. In pure λ-calculi this occurs when P is a value (thus P itself is the value emitted and there is no continuation). In HOπ it occurs when an output prefix is consumed, as in $\overline{a}P_1.\, P_2 \stackrel{\overline{a}}{\rightarrow} \langle P_1 \rangle P_2$, where process P_1 is emitted along channel a, leaving P_2 as a continuation.

—*Higher-order input transitions* $P \stackrel{\ell}{\rightarrow} (x)P'$, in which the abstraction $(x)P'$ so revealed then calls for a higher-order value to instantiate the variable x. In λ-calculus, abstractions are produced by terms such as $\lambda x.\, Q$; in HOπ, by input prefixes such as $a(x).\, Q$.

The bisimulation clause for first-order transitions is uncontroversial, and is the standard one for first-order languages. The main difficulty is finding the appropriate clauses for the higher-order inputs and outputs. transitions. Suppose $\mathcal{R}$ is a bisimulation, with (P, Q) in $\mathcal{R}$. Consider matching output transitions

$$P \xrightarrow{\ell} \langle P_1 \rangle P_2 \quad \text{and} \quad Q \xrightarrow{\ell} \langle Q_1 \rangle Q_2. \tag{$*$}$$

How should P_1, P_2, Q_1, Q_2 be related? Imposing P_1 bisimilar to Q_1, and P_2 bisimilar to Q_2, can be too strong a requirement. For instance, in HOπ and in calculi with information hiding (or generative names), it breaks the correspondence with contextual equivalence (see Sangiorgi [1996] for a discussion).

On the other hand, matching input transitions

$$P \xrightarrow{\ell} (x)P' \quad \text{and} \quad Q \xrightarrow{\ell} (x)Q' \tag{$**$}$$

raise the question of what should be substituted for x; that is, for which terms P_1 and Q_1 should we require $P'\{P_1/x\}\ \mathcal{R}\ Q'\{Q_1/x\}$? We discuss some possible choices.

—P_1 and Q_1 are all pairs of identical terms, as in applicative bisimulations (the most studied form of bisimulation for higher-order calculi, [Abramsky 1990; Ong 1988; Gordon 1993; Pitts 1997; Sands 1998; Lassen 1998]). This is unsound under the presence of generative names, data abstraction, or encryption [Jeffrey and Rathke 1999; Sumii and Pierce 2007b, 2007a]. Moreover, proving that bisimilarity is a congruence can be hard. To see why, consider an application context, and a pair of bisimilar functions M, N plus a pair of bisimilar arguments M', N'. We have to prove that MM' and NN' are bisimilar, but we are unable to apply the bisimulation hypothesis on the functions M and N since their arguments M' and N' are bisimilar but not necessarily identical. Difficulties also arise with up-to context techniques (the usefulness of these techniques in higher-order languages and its problems with applicative bisimulations have been extensively studied by Lassen [1998]; see also Sands [1998] and Koutavas and Wand [2006b]).

—P_1 and Q_1 are related by $\mathcal{R}$. This makes the above congruence argument for MM' and NN' work. However, this definition of bisimulation, which we call a *BA-bisimulation*,[1] breaks the monotonicity of the generating functional (the function from relations to relations that represents the clauses of bisimulation). Indeed, BA-bisimulations in general are unsound. For instance, take the identity function $I = \lambda x.\, x$ and $\Sigma = EE$ where $E = \lambda x.\, \lambda y.\, xx$. Term Σ is a "purely convergent term" because it always reduces to itself when applied to any argument, regardless of the input received. Of course I and Σ should not be regarded as bisimilar, yet $\{(I, \Sigma)\}$ would be a BA-bisimulation (the only related input is the pair (I, Σ) itself, and the result of the application is again the same pair).

[1]BA indicates that the bisimilarity uses "Bisimilar Arguments."

—P_1, Q_1 are fresh variables. This bisimulation method [Sangiorgi 1994; Støvring and Lassen 2007] is complete (with respect to contextual equivalence) only in certain extensions of the λ-calculus (e.g., call-by-value with *both* state *and* callcc), and would be incomplete in other languages (such as λ-calculus without state or/and callcc, and languages with constants or types). The method also reminds us of *normal bisimulation* [Sangiorgi 1992; Jeffrey and Rathke 2005]. This form of bisimulation, used in concurrency, essentially relies on a run-time translation of higher-order calculi to first-order, but is not always possible (e.g., in presence of constructs for passivation [Lenglet 2010]).

Environment bisimulations. In this article we propose *environment bisimulations* as a bisimulation method for higher-order languages.

A key idea of environment bisimulations is to make a clear distinction between the tested terms and the environment. An element of an environment bisimulation has, in addition to the tested terms P and Q, a further component $\mathcal{E}$, the environment, which expresses the observer's current knowledge. (In languages richer than pure λ-calculi, there may be other components, for instance to keep track of generated names.) The bisimulation requirements for higher-order inputs and outputs naturally follow. In the higher-order outputs $(*)$, (P_1, Q_1) are published to the environment, so they should be put into $\mathcal{E}$. Thus roughly the clause becomes:

if $(\mathcal{E}, P, Q) \in \mathcal{R}$ and $P \stackrel{\ell}{\rightarrow} \langle P_1 \rangle P_2$
then $Q \stackrel{\ell}{\rightarrow} \langle Q_1 \rangle Q_2$ and $(\mathcal{E} \cup \{(P_1, Q_1)\}, P_2, Q_2) \in \mathcal{R}$.

In the higher-order inputs $(**)$, the arguments P_1 and Q_1 should be terms that the observer can build using the current knowledge; that is, terms obtained by composing the values in $\mathcal{E}$ using the operators of the calculus. (For example, terms of the form $C[V_1, \ldots, V_n]$ and $C[W_1, \ldots, W_n]$ for a context C and values $(V_1, W_1), \ldots, (V_n, W_n)$ that are in $\mathcal{E}$.) We write $\mathcal{E}^\star$ for pairs of terms of this form. The clause roughly becomes:

if $(\mathcal{E}, P, Q) \in \mathcal{R}$ and $P \stackrel{\ell}{\rightarrow} (x)P'$ then $Q \stackrel{\ell}{\rightarrow} (x)Q'$ and $(\mathcal{E}, P'\{^{P_1}/_x\}, Q'\{^{Q_1}/_x\}) \in \mathcal{R}$ for any $(P_1, Q_1) \in \mathcal{E}^\star$.

Finally, we need clauses to express the observer's test capabilities on the environment. For instance, in λ-calculi the observer can check the consistency of values related in $\mathcal{E}$ (e.g., the outermost construct should be the same); in HOπ, the observer is allowed to run, at any time, processes that are related in the environment, which yields the clause:

$(\mathcal{E}, P, Q) \in \mathcal{R}$ implies $(\mathcal{E}, P \mid P_1, Q \mid Q_1) \in \mathcal{R}$, for $(P_1, Q_1) \in \mathcal{E}$.

As for BA-bisimulations, so in environment bisimulations testing higher-order inputs on *related* arguments facilitates proofs of congruence (and up-to contexts). But unlike BA-bisimulations, the separation between environment and tested terms maintains the monotonicity of the generating functional.

A possible drawback of environment bisimulations over, say, applicative bisimulations is that the set of arguments to related functions that have to

be considered in the bisimulation clause is larger (since it also includes nonidentical arguments). As a remedy to this, we propose the use of up-to techniques (in particular techniques involving up-to contexts), which are easier to establish for environment bisimulations than for applicative bisimulations, and which allow us to considerably enhance the bisimulation proof method.

We use a small-step, rather than big-step, semantics in environment bisimulations. This is important in nonconfluent languages but may seem cumbersome in, for example, λ-calculi, because it seems to require more elements in bisimulations than big-step semantics. Again, we remedy this with up-to techniques (such as "up-to reduction"). Further, small-step semantics together with up-to techniques sometimes *simplifies* equivalence proofs, as we can exploit the possibility of comparing terms in the middle of evaluations without having to reduce them to values (e.g., Example 4.1). Big-step versions of environment bisimulations are anyway derived as a corollary of soundness of certain up-to techniques.

Even leaving the issues of arguments of functions and big-step semantics, up-to techniques can sometimes substantially simplify the proof obligations required to establish a bisimilarity result. All examples in the paper indeed make use of nontrivial up-to techniques.

To test the robustness of environment bisimulations, we transport definitions and proof techniques from pure λ-calculi to λ-calculi with full-fledged store (a language with information hiding by generative names), and to Higher-Order π-calculi (as examples of concurrent, nondeterministic higher-order languages). In each case: we present the basic properties of environment bisimilarity, including its congruence properties (which are proved using standard bisimulation reasoning); we show that it coincides with contextual equivalence (in concurrency, this is barbed congruence); we develop a few up-to techniques as examples of possible enhancements of the associated bisimulation method. These techniques include *up-to contexts* and *up-to expansion*.

Environmental bisimulations have been inspired by bisimulations for calculi with information hiding mechanisms (π-calculus with types or encryption [Boreale and Sangiorgi 1998; Pierce and Sangiorgi 2000; Abadi and Gordon 1998; Boreale et al. 1999], and λ-calculi with encryption [Sumii and Pierce 2007a], data abstraction [Sumii and Pierce 2007b], and store [Koutavas and Wand 2006b]), (encryption [Sumii and Pierce 2007a], data abstraction [Sumii and Pierce 2007b], and store [Koutavas and Wand 2006b]), where the use of an environment collecting the information accumulated from the observer was necessary because of the information hiding. In this respect, our contribution in this paper is to isolate this idea, simplify and strengthen the method, and develop its basic theory, in order to propose it as a general method for higher-order languages. The reader will find in Section 7 more details on the differences over previous methods.

Overview of the article. The rest of this article is structured as follows. Section 2 gives definitions and notations required for what follows. Section 3 defines environmental bisimulations for pure untyped call-by-name λ-calculus, and establishes various up-to techniques to simplify equivalence proofs using

environmental bisimulations. Section 4 presents a call-by-value version of environmental bisimulations, and Section 5 extends it with with ML-like references. Section 6 presents environmental bisimulations in a higher-order calculus for concurrency, the Higher-Order π-calculus. Section 7 discusses related work, and Section 8 concludes, mentioning also some possible future work.

2. PRELIMINARIES

In this section, we introduce general notations and terminologies used throughout the article. Familiarity with standard terminologies (such as free/bound variables and α-conversion) is assumed.

We use metavariables $M, N, P, Q, \ldots$ for terms, and $V, W, \ldots$ for values (where the notion of terms and values varies depending on the calculus being considered). We identify α-convertible terms. We write $M\{^N/_x\}$ for the capture-avoiding substitution of N for x in M. A term is *closed* if it contains no free variables. The set of free variables of a term M is $\mathsf{fv}(M)$. A *context* C is an expression obtained from a term by replacing some sub-terms with *holes* of the form $[\cdot]_i$. We write $C[M_1, \ldots, M_n]$ for the term obtained by replacing each occurrence of $[\cdot]_i$ in C with M_i. Note that a context may contain no holes, and therefore any term is a context. The context may bind variables in $M_1, \ldots, M_n$; for example, if $C = \lambda x.[\cdot]_1$ and $M = x$, then $C[M]$ is $\lambda x.x$, not $\lambda y.x$.

We use metavariables $\mathcal{R}, \mathcal{S}, \mathcal{E}, \mathcal{F}, \ldots$ for binary relations; $\mathcal{R}\mathcal{S}$ is the composition of $\mathcal{R}$ and $\mathcal{S}$, whereas $\mathcal{R}^\star$ is the *closure of relation* $\mathcal{R}$ *under contexts*, that is,

$$\{(C[M_1, \ldots, M_n], C[N_1, \ldots, N_n]) \text{ s. t. } M_i \mathcal{R} N_i, \forall i\}$$

By definition $\mathcal{R}^\star$ contains both R and the identity relation. By default, we restrict $\mathcal{R}^\star$ to closed terms unless noted otherwise.

Sequences $M_1, \ldots, M_n$ are often abbreviated to $\widetilde{M}$, and notations are extended to tuples componentwise. Hence, we often write $C[\widetilde{M}]$ for $C[M_1, \ldots, M_n]$, and $\widetilde{M}\mathcal{R}\widetilde{N}$ for $(M_1 \mathcal{R} N_1) \wedge \cdots \wedge (M_n \mathcal{R} N_n)$.

We have some remarks on the results in the remainder of this article.

—Although the results are often stated for closed values only, they can be generalized to open terms in a common way. In λ-calculi, properties between open terms M and N can be derived from the corresponding properties between the closed terms $\lambda\widetilde{x}.M$ and $\lambda\widetilde{x}.N$, for $\{\widetilde{x}\} \supseteq \mathsf{fv}(M) \cup \mathsf{fv}(N)$. Note that the congruence of M and $(\lambda\widetilde{x}.M)\widetilde{x}$ can be proved by defining an ad hoc relation—the least congruence containing $(M, (\lambda x.M)x)$ for every M—and proving its preservation under evaluation, as in Sumii-Pierce [Sumii and Pierce 2007a] and Koutavas-Wand [Koutavas and Wand 2006b]. (Alternatively, we may also consider a bisimulation between M and $(\lambda x.M)x$. The proof is straightforward in either case.) In HOπ, it is similar; for instance, if $\mathsf{fv}(M) \subseteq \{x\}$ then one relates processes M and $\nu a\,(a(x).M \mid \overline{a}x.\mathbf{0})$, where a is a fresh name.

—The results in this article are stated for untyped languages. Adapting them to languages with a simply typed discipline is straightforward. (We use a simply typed calculus in an example.)

3. CALL-BY-NAME λ-CALCULUS

3.1 Syntax and Reduction

The set Λ of pure λ-terms is defined by:

$$M, N ::= x \mid \lambda x.\, M \mid MN.$$

We write $\Lambda^\bullet$ for the subset of closed terms. The *call-by-name reduction relation* $\longrightarrow$ is the least relation over $\Lambda^\bullet$ that is closed under the following rules.

$$\beta: \ (\lambda x.\, M)N \longrightarrow M\{^N/_x\} \qquad \mu: \ \frac{M \longrightarrow M'}{MN \longrightarrow M'N}.$$

We write $\Longrightarrow$ for the reflexive and transitive closure of $\longrightarrow$. The values are the terms of the form $\lambda x.\, M$. In call-by-name *evaluation contexts* are described by the following grammar.

$$C := CM \mid [\cdot].$$

Note in particular that the hole is not under a lambda and occurs exactly once. (Symbol C is used also for arbitrary contexts; it will be explicitly indicated when C refers to evaluation contexts.)

3.2 Environmental Bisimilarity

An *environmental relation* is a set of elements each of which is of the form $(\mathcal{E}, M, N)$ or $\mathcal{E}$, and where M, N are closed terms and $\mathcal{E}$ is a relation on closed values. We use $\mathcal{X}, \mathcal{Y}$ to range over environmental relations. In a triple $(\mathcal{E}, M, N)$ the relation component $\mathcal{E}$ is the *environment*, and M, N are the *tested terms*. We write $M\ \mathcal{X}_\mathcal{E}\ N$ for $(\mathcal{E}, M, N) \in \mathcal{X}$.

Definition 3.1. An environmental relation $\mathcal{X}$ is an *environmental bisimulation* if

(1) $M\ \mathcal{X}_\mathcal{E}\ N$ implies:
   (a) if $M \longrightarrow M'$ then $N \Longrightarrow N'$ and $M'\ \mathcal{X}_\mathcal{E}\ N'$;
   (b) if $M = V$ then $N \Longrightarrow W$ and $\mathcal{E} \cup \{(V, W)\} \in \mathcal{X}$;
   (c) the converse of the preceeding two conditions, on N;
(2) if $\mathcal{E} \in \mathcal{X}$ then for all $(\lambda x.\, P, \lambda x.\, Q) \in \mathcal{E}$ and for all $(M_1, N_1) \in \mathcal{E}^\star$ it holds that $P\{^{M_1}/_x\}\ \mathcal{X}_\mathcal{E}\ Q\{^{N_1}/_x\}$.

We write $\approx$ for the union of all environmental bisimulations, and call it *environmental bisimilarity*.

Relations $\approx_\mathcal{E}$, as well as the other relations in the paper, are extended to open terms using closing abstractions. Thus if $\widetilde{x} = \mathsf{fv}(M, N)$ then $M \approx_\mathcal{E} N$ if $\lambda\widetilde{x}.\, M \approx_\mathcal{E} \lambda\widetilde{x}.\, N$.

For examples and applications of $\approx_\mathcal{E}$, the most important case is when $\mathcal{E} = \emptyset$: this states the equivalence between two terms without any predefined knowledge from the observer. We therefore introduce a special symbol for it, and write $M \simeq N$ as an abbreviation for $M \approx_\emptyset N$.

Example 3.2. We have $I_1 \simeq I_2$ for $I_1 \stackrel{\text{def}}{=} \lambda x.\, x$ and $I_2 \stackrel{\text{def}}{=} (\lambda x.\, x)(\lambda x.\, x)$, by taking $\mathcal{X} = \{(\mathcal{E}, I_1, I_2) \mid \mathcal{E} \subseteq Id\} \cup \{(\mathcal{E}, M, M) \mid \mathcal{E} \subseteq Id \wedge M \in \Lambda^\bullet\} \cup \{\mathcal{E} \mid \mathcal{E} \subseteq Id\}$

for $Id = \{(V, V) \mid V \in \Lambda^\bullet\}$. Note that the singleton set $\{(\emptyset, I_1, I_2)\}$ by itself is not an environmental bisimulation because of the $\mathcal{E}^\star$ in clause (2). Burdens like this (or more) are common to almost every bisimulation proof based on the definition above, and will be removed by the up-to techniques described later in this section. Specifically, the finite set $\{(\emptyset, I_1, I_2), \emptyset\}$ will be an environmental bisimulation "up to context."

Example 3.3. We have $I_1 \simeq I_3$ for $I_1 \stackrel{\text{def}}{=} \lambda x.x$ and $I_3 \stackrel{\text{def}}{=} \lambda x.(\lambda y.y)x$, by taking $\mathcal{X} = \{(\mathcal{E}, M, N), (\mathcal{E}, M, (\lambda y.y)N) \mid \mathcal{E} \subseteq \mathcal{S}^\star \wedge M\ \mathcal{S}^\star\ N\} \cup \{\mathcal{E} \mid \mathcal{E} \subseteq \mathcal{S}^\star\}$ for $\mathcal{S} = \{(I_1, I_3)\}$. Proving this to be an environmental bisimulation only using the definition above is even harder: we need an induction on contexts in clause (1.a). Again, with up-to techniques, the finite set $\{(\emptyset, I_1, I_3), \mathcal{S}\}$ suffices (it is an environmental bisimulation up to context and reduction).

3.3 Basic Properties

It is immediate to check that the union of bisimulations is a bisimulation itself; hence we derive the following.

LEMMA 3.4. *$\approx$ is the largest environmental bisimulation.*

The next lemma shows that environments can be weakened at will.

LEMMA 3.5 (WEAKENING). *$M \approx_{\mathcal{E}} N$ and $\mathcal{E}' \subseteq \mathcal{E}$ imply $M \approx_{\mathcal{E}'} N$.*

LEMMA 3.6. *Suppose $\mathcal{X}$ is an environmental bisimulation, $M\ \mathcal{X}_{\mathcal{E}}\ N$ and $M \Longrightarrow M'$. Then $N \Longrightarrow N'$, for some N' with $M'\ \mathcal{X}_{\mathcal{E}}\ N'$.*

The following property of composition of environment bisimilarities immediately gives us the transitivity of $\simeq$.

LEMMA 3.7. *$M \approx_{\mathcal{E}_1} N$ and $N \approx_{\mathcal{E}_2} L$ imply $M \approx_{\mathcal{E}_1 \mathcal{E}_2} L$.*

PROOF. One shows that $\{\mathcal{E}' \mid \mathcal{E}' \subseteq \mathcal{E}_1 \mathcal{E}_2 \text{ for } \mathcal{E}_1, \mathcal{E}_2 \in \approx\} \cup \{(\mathcal{E}', P, R) \mid \mathcal{E}' \subseteq \mathcal{E}_1 \mathcal{E}_2,\ P \approx_{\mathcal{E}_1} Q \approx_{\mathcal{E}_2} R, \text{ for some } \mathcal{E}_1, \mathcal{E}_2\}$ is an environmental bisimulation. □

COROLLARY 3.8. *$\simeq$ is an equivalence relation.*

3.4 Basic Up-To Techniques

We now introduce a few basic and simple "up-to" techniques, as enhancements of the bisimulation proof method. We will use them to derive the congruence properties of environmental bisimilarity. Up-to techniques allow us to prove bisimulation results using relations that in general are not themselves bisimulations, but are contained in a bisimulation. Thus they can simplify a lot the work needed to derive bisimilarity results. We only sketch the proof of soundness of the techniques, whose details are straightforward.

Up to environment. This technique introduces some flexibility in the choice of the environment for two tested terms, by allowing environments that are larger than those requested by Definition 3.1 (by Lemma 3.5, a larger environment gives a stronger requirement). For instance, the technique can allow us

to avoid environments that incrementally grow during the bisimulation game, retaining instead only their limit (i.e., the largest environment). In clause (1.a), we replace "$M' \; \mathcal{X}_{\mathcal{E}} \; N'$" with "$M' \; \mathcal{X}_{\mathcal{E}'} \; N'$ for some $\mathcal{E}'$ with $\mathcal{E} \subseteq \mathcal{E}'$," and similarly in (2); in (1.b), we replace "$\mathcal{E} \cup \{(V, W)\} \in \mathcal{X}$" with "$\mathcal{E}' \in \mathcal{X}$ for some $\mathcal{E}'$ with $\mathcal{E} \cup \{(V, W)\} \subseteq \mathcal{E}'$."

Definition 3.9 (*up to environment*). An environmental relation $\mathcal{X}$ is an *environmental bisimulation up to environment* if

(1) $M \; \mathcal{X}_{\mathcal{E}} \; N$ implies:
  (a) if $M \longrightarrow M'$ then $N \Longrightarrow N'$ and $M' \; \mathcal{X}_{\mathcal{E}'} \; N'$ for some $\mathcal{E}' \supseteq \mathcal{E}$;
  (b) if $M = V$ then $N \Longrightarrow W$ and $\mathcal{E}' \in \mathcal{X}$ for some $\mathcal{E}' \supseteq \mathcal{E} \cup \{(V, W)\}$;
  (c) the converse of the above two conditions, on N;
(2) if $\mathcal{E} \in \mathcal{X}$ then for all $(\lambda x.\, P, \lambda x.\, Q) \in \mathcal{E}$ and for all $(M_1, N_1) \in \mathcal{E}^\star$ it holds that $P\{M_1/x\} \; \mathcal{X}_{\mathcal{E}'} \; Q\{N_1/x\}$ for some $\mathcal{E}' \supseteq \mathcal{E}$.

LEMMA 3.10. *If $\mathcal{X}$ is an environmental bisimulation up to environment then $\mathcal{X} \subseteq \approx$.*

PROOF. One shows that

$$\{(\mathcal{E}, M, N) \text{ s.t. } \exists \mathcal{E}' \text{ with } \mathcal{E} \subseteq \mathcal{E}' \text{ and } M \; \mathcal{X}_{\mathcal{E}'} \; N\} \cup \{\mathcal{E} \text{ s.t. } \exists \mathcal{E}' \text{ with } \mathcal{E} \subseteq \mathcal{E}' \text{ and } \mathcal{E}' \in \mathcal{X}\}$$

is an environmental bisimulation. □

Up to bisimilarity. This technique introduces a (limited) use of $\simeq$ on tested terms. This allows us to avoid bisimulations with elements that, behaviorally, are the same. In clause (1.a), we replace "$M' \; \mathcal{X}_{\mathcal{E}} \; N''$" with "$M' \; \mathcal{X}_{\mathcal{E}} \simeq N''$" (where $\mathcal{X}_{\mathcal{E}} \simeq$ is relational composition; that is, there is N'' with $M' \; \mathcal{X}_{\mathcal{E}} \; N''$ and $N'' \simeq N'$); and similarly in (1.b). In (2), we replace "$P\{M_1/x\} \; \mathcal{X}_{\mathcal{E}} \; Q\{N_1/x\}$" with "$P\{M_1/x\} \simeq \mathcal{X}_{\mathcal{E}} \simeq Q\{N_1/x\}$."

Definition 3.11 (*up to bisimilarity*). An environmental relation $\mathcal{X}$ is an *environmental bisimulation up to bisimilarity* if

(1) $M \; \mathcal{X}_{\mathcal{E}} \; N$ implies:
  (a) if $M \longrightarrow M'$ then $N \Longrightarrow N'$ and $M' \; \mathcal{X}_{\mathcal{E}} \simeq N'$;
  (b) if $M = V$ then $N \Longrightarrow W$ and $\mathcal{E} \cup \{(V, W')\} \in \mathcal{X}$ for some $W' \simeq W$;
  (c) the converse of the above two conditions, on N;
(2) if $\mathcal{E} \in \mathcal{X}$ then for all $(\lambda x.\, P, \lambda x.\, Q) \in \mathcal{E}$ and for all $(M_1, N_1) \in \mathcal{E}^\star$ it holds that $P\{M_1/x\} \simeq \mathcal{X}_{\mathcal{E}} \simeq Q\{N_1/x\}$.

We cannot strengthen up-to bisimilarity by using $\simeq$ also on the left-hand side of $\mathcal{X}$ in clauses (1.a) and (1.b), for the technique would be unsound; this is similar to the problems of up-to bisimilarity in standard small-step bisimilarity [Milner 1989].

LEMMA 3.12. *If $\mathcal{X}$ is an environmental bisimulation up to $\simeq$ then $\mathcal{X} \subseteq \approx$.*

PROOF. For this proof, we extend relation $\simeq$ to environments by requiring that the pairs of two environments be componentwise related: $\mathcal{E} \simeq \mathcal{E}'$ if

$$\begin{array}{l}\mathcal{E} = \cup_{i\in I}\{(V_i, W_i)\}, \text{ for some indexing set } I\\ \text{and } \forall i\ \exists V_i', W_i' \text{ with } V_i \simeq V_i', W_i \simeq W_i'\\ \text{and } \mathcal{E}' = \cup_{i\in I}\{(V_i', W_i')\}\end{array}$$

With this notation at hand, we prove the lemma. Given an environmental bisimulation $\mathcal{X}$ up to $\simeq$, consider the environmental relation

$$\begin{array}{rl}\mathcal{Y} \stackrel{\text{def}}{=} & \{(\mathcal{E}, M, N) \text{ s.t. } \mathcal{E} \simeq \mathcal{E}' \text{ and } M \simeq \mathcal{X}_{\mathcal{E}'} \simeq N, \text{ for some } \mathcal{E}'\}\\ & \cup\{\mathcal{E} \text{ s.t. } \mathcal{E} \simeq \mathcal{E}' \text{ for some } \mathcal{E}' \in \mathcal{X}\}\end{array}$$

One shows that $\mathcal{Y}$ is an environmental bisimulation. This is sufficient for the lemma, as $\mathcal{X} \subseteq \mathcal{Y}$ (by the reflexivity of $\simeq$). The proof is similar to the soundness of the up-to-bisimilarity technique for ordinary bisimulation in the literature.

First one shows that if $M\ \mathcal{X}_{\mathcal{E}} \simeq\ N$ and $M \Longrightarrow M_1$ then there is N_1 s.t. $N \Longrightarrow N_1$ and $M_1\ \mathcal{X}_{\mathcal{E}} \simeq\ N_1$. This fact is proved proceeding by induction on the length of the weak transition $M \Longrightarrow M_1$ (the number of steps that compose it), using Lemma 3.6 and the transitivity of $\simeq$.

Using the above fact, the first clause of the definition of environmental bisimulation is proved as follows. Suppose $M \simeq M'\ \mathcal{X}_{\mathcal{E}} \simeq\ N$ and $M \longrightarrow M_1$. Since $\simeq\ = \approx_{\emptyset}$ and $\approx$ is an environmental bisimulation, also $M' \Longrightarrow M_1' \simeq M_1$, for some M_1'. Using this and the fact above, we find that there is N_1 with $N \Longrightarrow N_1$ and $M_1 \simeq M_1'\ \mathcal{X}_{\mathcal{E}} \simeq\ N_1$. The other clauses of the bisimulation are proved reasoning similarly. □

More sophisticated up to techniques will be discussed in Section 3.6.

3.5 Compositionality Properties

We now consider the congruence properties of the bisimilarity. In bisimilarities for higher-order languages, these are usually the most delicate basic properties to establish. With the exception of Lemma 3.22, Corollary 3.23, and Corollary 3.24, the properties below will hold in all the λ-calculi we consider in the paper (with occasional minor adjustments).

Notation 3.13. For a relation $\mathcal{R}$ we write $\mathcal{R}^{\widehat{\star}}$ for the subset of $\mathcal{R}^{\star}$ that only relates pairs of values. Accordingly, $\widehat{\approx}_{\mathcal{E}}$ is the restriction of $\approx_{\mathcal{E}}$ to values, and similarly for $\widehat{\simeq}$.

Definition 3.14. If C is a context with possibly several holes, then a hole $[\cdot]_i$ is in a *redex position in C* if the context obtained by filling all the holes but $[\cdot]_i$ with values is an evaluation context.

LEMMA 3.15 (CONGRUENCE FOR VALUES). *For all $\mathcal{E}$, relation $\widehat{\approx}_{\mathcal{E}}$ is a congruence. In particular, $\widehat{\simeq}$ is a congruence.*

LEMMA 3.16 (ARBITRARY TERMS UNDER EVALUATION CONTEXTS). *For all $\mathcal{E}$, relation $\approx_{\mathcal{E}}$ is preserved by evaluation contexts (i.e., if $M \approx_{\mathcal{E}} N$ and C is an evaluation context, then also $C[M] \approx_{\mathcal{E}} C[N]$).*

Lemmas 3.15 and 3.16 are proved simultaneously, in the following proof.

PROOF. Suppose $\mathcal{Y}$ is a bisimulation, and take

$$\begin{array}{rl}\mathcal{X} \stackrel{\text{def}}{=} & \{(\widehat{\mathcal{E}^\star}, C[M, \widetilde{V}], C[N, \widetilde{W}]) \text{ s.t.} \\ & M\ \mathcal{Y}_{\mathcal{E}}\ N, \\ & \text{the first hole of } C \text{ is in redex position}, \\ & V_i\ \mathcal{E}\ W_i\} \\ & \cup\ \{(\widehat{\mathcal{E}^\star}, C[\widetilde{V}], C[\widetilde{W}]) \text{ s.t. } \mathcal{E} \in \mathcal{Y} \text{ and } V_i\ \mathcal{E}\ W_i\} \\ & \cup\ \{\widehat{\mathcal{E}^\star} \text{ s.t. } \mathcal{E} \in \mathcal{Y}\ \}\end{array}$$

We show that this is a bisimulation up to environment. We first prove the bisimulation for elements of the form

$$(\widehat{\mathcal{E}^\star}, C[\widetilde{V}], C[\widetilde{W}]). \tag{1}$$

On these elements, clause (1.b) is immediate: if $C[\widetilde{V}]$ is a value, then also $C[\widetilde{W}]$ is a value and they are in $\widehat{\mathcal{E}^\star}$. This is sufficient, because $\widehat{\mathcal{E}^\star} \in \mathcal{X}$. Clause (1.a) is proved by an induction on C. The details are easy: the only possible case is $C = C_1 C_2$; we use the fact that if $\lambda x.P\ \mathcal{E}^\star\ \lambda x.Q$, and $V\ \mathcal{E}^\star\ W$, and $\mathcal{E} \in \mathcal{Y}$, then $(P\{^V/_x\}, Q\{^W/_x\}) \in \mathcal{X}_{\mathcal{E}'}$ for some $\mathcal{E}'$ with $\mathcal{E} \subseteq \mathcal{E}'$, which follows from the definition of bisimulation (we exploit here the peculiarity of clause (2) of environmental bisimulations on abstractions).

The requirements for elements of the form

$$(\widehat{\mathcal{E}^\star}, C[M, \widetilde{V}], C[N, \widetilde{W}]) \tag{2}$$

and $\widehat{\mathcal{E}^\star}$ follow from the proof for the elements (1) above together with some basic properties of evaluation contexts on transitions (the fact that hole $[\cdot]_1$ of an evaluation context is active and therefore the term that replaces this hole is free to reduce). In elements of the form (2) we distinguish the cases when M is a value and when it is not.

More details are found in Appendix A. □

The environmental bisimulation constructed in the proof of the previous lemmas, where an environment $\mathcal{E}$ is lifted to $\widehat{\mathcal{E}^\star}$, also proves the following result.

LEMMA 3.17. *$M \approx_{\mathcal{E}} N$ implies $M \approx_{\widehat{\mathcal{E}^\star}} N$.*

The lemma above can also be proved directly, setting $\mathcal{X}$ to be $\approx \cup \{(\mathcal{E}^\star, M, N) \text{ s.t. } M \approx_{\mathcal{E}} N\}$, and showing that $\mathcal{X}$ is a bisimulation up to environment.

From the two Lemmas 3.15 and 3.16 we derive, for the case of an empty environment:

— $V \simeq W$ implies $C[V] \simeq C[W]$, for any context C;
— $M \simeq N$ implies $C[M] \simeq C[N]$ for any evaluation context C.

We next present a few other properties of environmental bisimilarity, obtained from the congruence results above, and which we will use to establish further compositionality properties.

LEMMA 3.18. *$\mathcal{E} \in \approx$ if and only if $\mathcal{E} \subseteq \approx_{\mathcal{E}}$.*

PROOF. For the implication from left to right, suppose $\mathcal{X}$ is a bisimulation with $\mathcal{E} \in \mathcal{X}$. Take $V \mathcal{E} W$. Then also $\mathcal{X} \cup \{(\mathcal{E}, V, W)\}$ is a bisimulation.

For the opposite implication, take $\mathcal{E} \subseteq \approx_{\mathcal{E}}$. If $\mathcal{E} = \emptyset$, it vacuously satisfies the bisimulation clause, so trivially $\mathcal{E} \in \approx$. Otherwise, let $V \ \mathcal{E} \ W$, that is, $V \approx_{\mathcal{E}} W$. We then have $\mathcal{E} \cup \{(V, W)\} \in \approx$, i.e., $\mathcal{E} \in \approx$, from the bisimulation clause. □

We can now prove a stronger form of transitivity than that given by Lemma 3.7.

LEMMA 3.19. *$M \approx_{\mathcal{E}_1} N$ and $N \approx_{\mathcal{E}_2} L$ imply $M \approx_{\mathcal{E}_1^{\widehat{\star}} \ \mathcal{E}_2^{\widehat{\star}}} L$.*

PROOF. From $M \approx_{\mathcal{E}_1} N$ and $N \approx_{\mathcal{E}_2} L$ we derive, by Lemma 3.17, $M \approx_{\mathcal{E}_1^{\widehat{\star}}} N$ and $N \approx_{\mathcal{E}_2^{\widehat{\star}}} L$. Then we conclude with Lemma 3.7. □

COROLLARY 3.20. *(1) $M \approx_{\mathcal{E}} N$ and $N \simeq L$ imply $M \approx_{\mathcal{E}} L$;*
(2) $M \approx_{\mathcal{E}_1} N$ and $N \approx_{\mathcal{E}_2} L$ imply $M \approx_{\mathcal{E}_1 \cup \mathcal{E}_2} L$.

PROOF. For (1), suppose $(P, Q) \in \mathcal{E}$. We also have $(Q, Q) \in \widehat{\emptyset^{\star}}$. Therefore $(P, Q) \in (\widehat{\mathcal{E}^{\star}} \ \widehat{\emptyset^{\star}})$ (the composition of the two relations), which proves $\mathcal{E} \subseteq (\widehat{\mathcal{E}^{\star}} \ \widehat{\emptyset^{\star}})$. Since from Lemma 3.19 we have $M \approx_{\widehat{\mathcal{E}^{\star}} \ \widehat{\emptyset^{\star}}} L$, applying Weakening (Lemma 3.5) we deduce $M \approx_{\mathcal{E}} L$. Part (2) of the corollary is similar. □

We now continue with the compositionality properties of environmental bisimilarity. We write $\lambda . M$ for the thunk obtained from M (i.e., a term $\lambda x . M$ with $x \notin \mathsf{fv}(M)$).

LEMMA 3.21 (CONGRUENCE FOR ARBITRARY TERMS, THUNKED). *For all $\mathcal{E}$, if $\lambda . M \approx \mathcal{E} \lambda . N$, then also $C[M] \approx_{\mathcal{E}} C[N]$, for all contexts C.*

PROOF. The proof uses Lemma 3.15 and validity of β-conversion (the fact that $(\lambda x . M)N \simeq M\{^N/_x\}$, for any $\lambda x . M$ and N closed, is preserved by any context; see Lemma A.9). We then have $C[M] \simeq C[(\lambda . M)\star]$ (where $\star$ indicates an arbitrary term) and $C[N] \simeq C[(\lambda . N)\star]$, and using the hypothesis of the lemma, $C[(\lambda . M)\star] \approx_{\mathcal{E}} C[(\lambda . N)\star]$. Therefore, using Corollary 3.20, $C[M] \approx_{\mathcal{E}} C[N]$. □

In the case of call-by-name and other pure λ-calculi, the above result can be simplified and strengthened, obtaining that each relation $\approx_{\mathcal{E}}$ is a congruence.

LEMMA 3.22. *$M \simeq N$ implies $C[M] \approx_{\widehat{\simeq^{\star}}} C[N]$, for all contexts C.*

PROOF. The following relation is a bisimulation up to bisimilarity:

$$\mathcal{X} \stackrel{\text{def}}{=} \{(\widehat{\simeq^{\star}}, C[\widetilde{M}], C[\widetilde{N}]) \ \text{ s.t. } \ \widetilde{M} \simeq \widetilde{N}\} \ \cup \ \{\widehat{\simeq^{\star}}\}.$$

For elements $(\widehat{\simeq^{\star}}, C[\widetilde{M}], C[\widetilde{N}])$, clauses (1.a) and (1.b) are proved by a (straightforward) induction on the size of the context. Lemma A.7 is used to handle the up-to-bisimilarity that originates from the induction hypothesis.

The case of elements of $\widehat{\simeq^{\star}}$, is immediate, using the definition of bisimulations. We have either $\lambda x . P \simeq \lambda x . Q$, or $\lambda x . P$ and $\lambda x . Q$ have a common non-empty context and only the arguments of such context are related. In the first

case, since $\simeq$ is a bisimulation we have.

$$P\{^V/_x\}\ \mathcal{X}_{\simeq^\star}\ P\{^W/_x\} \simeq Q\{^W/_x\},$$

which is sufficient, as $\mathcal{X}$ is bisimulation up to bisimilarity. In the second case, we have $P\{^V/_x\}\ \mathcal{X}_{\simeq^\star}\ Q\{^W/_x\}$. See Lemma A.10 for more details. □

COROLLARY 3.23. *For all $\mathcal{E} \in \approx$, relation $\approx_{\mathcal{E}}$ is a congruence.*

PROOF. We use Lemma 3.22 plus weakening, and Lemma 3.18 (which, together with weakening, allows to infer $\mathcal{E} \subseteq \simeq$). Precisely, if $M \approx_{\mathcal{E}} N$, then, by weakening, $M \simeq N$. By Lemma 3.22, $C[M] \approx_{\simeq^\star} C[N]$, and then again by weakening, $C[M] \approx_{\mathcal{E}} C[N]$. □

COROLLARY 3.24. *Relation $\simeq$ is a congruence relation.*

Corollary 3.23 needs the condition $\mathcal{E} \in \approx$. For instance, if M and N are divergent, then $M \approx_{\mathcal{E}} N$ for all $\mathcal{E}$. However, $\lambda x.\, M \not\approx_{\mathcal{E}} \lambda x.\, N$ if $\mathcal{E} \notin \approx$ (clause (1.b) of Definition 3.1 fails because it requires $\mathcal{E} \cup \{(\lambda x.\, M, \lambda x.\, N)\} \in \approx$, which implies $\mathcal{E} \in \approx$ by Lemma 3.5).

3.6 Further Up-To Techniques

We present here some further up-to techniques. We first show some distinct techniques (up to reduction, up to expansion, up to context), and then discuss combinations of them. We should stress that these techniques, like those in Section 3.4, are well-known and established techniques for bisimulation (see, e.g., Sangiorgi and Walker [2001]), and are adapted to environmental bisimulation in the expected way. For this reason, we only sketch their soundness proofs. See also Section 5.4 for details on proofs of up-to techniques for an imperative λ-calculus (the imperative features make the techniques and their proofs more delicate).

Up to reduction. This technique exploits the confluence property of reduction so to replace tested terms with derivatives of them. When reduction is confluent this technique avoids the main disadvantage of small-step bisimulations over the big-step ones, namely the need of considering each single derivative of a tested term. In clause (1.a) of Definition 3.1, we replace "$M'\ \mathcal{X}_{\mathcal{E}}\ N'$" with "there are M'', N'' with $M' \Longrightarrow M''$ and $N' \Longrightarrow N''$ such that $M''\ \mathcal{X}_{\mathcal{E}}\ N''$"; similarly, we act on clause (2).

Definition 3.25 (*up to reduction*). An environmental relation $\mathcal{X}$ is an *environmental bisimulation up to reduction* if

(1) $M\ \mathcal{X}_{\mathcal{E}}\ N$ implies:
   (a) if $M \longrightarrow M'$ then $N \Longrightarrow N'$ and there are M'', N'' with $M' \Longrightarrow M''$ and $N' \Longrightarrow N''$ such that $M''\ \mathcal{X}_{\mathcal{E}}\ N''$;
   (b) if $M = V$ then $N \Longrightarrow W$ and $\mathcal{E} \cup \{(V, W)\} \in \mathcal{X}$;
   (c) the converse of the above two conditions, on N;
(2) if $\mathcal{E} \in \mathcal{X}$ then for all $(\lambda x.\, P, \lambda x.\, Q) \in \mathcal{E}$ and for all $(M_1, N_1) \in \mathcal{E}^\star$ there are M', N' with $P\{^{M_1}/_x\} \Longrightarrow M'$ and $Q\{^{N_1}/_x\} \Longrightarrow N'$ such that $M'\ \mathcal{X}_{\mathcal{E}}\ N'$.

LEMMA 3.26. *If $\mathcal{X}$ is an* environmental bisimulation up to reduction*, then* $\mathcal{X} \subseteq \approx$.

PROOF. One shows that

$$\mathcal{Y} \stackrel{\text{def}}{=} \{(\mathcal{E}, M, N) \text{ s.t. } \exists M', N' \text{ with } M \Longrightarrow M', N \Longrightarrow N' \text{ and } M' \; \mathcal{X}_{\mathcal{E}} \; N'\} \cup \{\mathcal{E} \text{ s.t. } \mathcal{E} \in \mathcal{X}\}$$

is an environmental bisimulation. □

The technique allows us to derive the soundness of the "big-step" version of environmental bisimulation, in which clauses (1.a) and (1.b) are unified, as by clause (1) below.

Definition 3.27 (*big-step environmental bisimulation*). An environmental relation $\mathcal{X}$ is a *big-step environmental bisimulation* if

(1) $M \; \mathcal{X}_{\mathcal{E}} \; N$ implies:
   (a) if $M \Longrightarrow V$ then $N \Longrightarrow W$ and $\mathcal{E} \cup \{(V, W)\} \in \mathcal{X}$;
   (b) the converse of the above condition, on N;
(2) if $\mathcal{E} \in \mathcal{X}$ then for all $(\lambda x.\, P, \lambda x.\, Q) \in \mathcal{E}$ and for all $(M_1, N_1) \in \mathcal{E}^\star$ it holds that $P\{^{M_1}/_x\} \; \mathcal{X}_{\mathcal{E}} \; Q\{^{N_1}/_x\}$.

LEMMA 3.28. *If $\mathcal{X}$ is a big-step environmental bisimulation, then* $\mathcal{X} \subseteq \approx$.

PROOF. One shows that

$$\begin{array}{l} \{(\mathcal{E}, M, N) \text{ s.t. } M \; \mathcal{X}_{\mathcal{E}} \; N\} \\ \cup \{(\mathcal{E}, M, N) \text{ s.t. } (M, N) \in \mathcal{E} \text{ and } \mathcal{E} \in \mathcal{X}\} \\ \cup \{\mathcal{E} \text{ s.t. } \mathcal{E} \in \mathcal{X}\} \\ \cup \{(\mathcal{E}, M, N) \text{ s.t. } M \text{ and } N \text{ diverge}\} \end{array}$$

is an environmental bisimulation up to reduction. □

Up to expansion. In concurrency, a useful auxiliary relation for up-to techniques is the *expansion relation*. (A similar relation is Sands' improvement for functional languages [Sands 1998].) We adapt here the concept of expansion to the λ-calculus. We write $M \Longrightarrow_n M'$ if M reduces to M' in n steps.

Definition 3.29 (*expansion*). An environmental relation $\mathcal{X}$ is an *environmental expansion relation* (or, briefly, an *expansion relation*) if

(1) $M \; \mathcal{X}_{\mathcal{E}} \; N$ implies:
   (a) if $M \Longrightarrow_m V$ then $N \Longrightarrow_n W$ with $m \leq n$ and $\mathcal{E} \cup \{(V, W)\} \in \mathcal{X}$;
   (b) the converse of the above condition, on N; that is, if $N \Longrightarrow_n W$ then $M \Longrightarrow_m V$ with $m \leq n$ and $\mathcal{E} \cup \{(V, W)\} \in \mathcal{X}$;
(2) if $\mathcal{E} \in \mathcal{X}$ then for all $(\lambda x.\, P, \lambda x.\, Q) \in \mathcal{E}$ and for all $(M_1, N_1) \in \mathcal{E}^\star$ it holds that $P\{^{M_1}/_x\} \; \mathcal{X}_{\mathcal{E}} \; Q\{^{N_1}/_x\}$.

The union of all environmental expansion relations is *environmental expansion* (or, briefly, *expansion*).

Proceeding similarly to the proof of congruence for $\approx$, we obtain, the following.

LEMMA 3.30. *Expansion is the largest expansion relation, and is a pre-congruence.*

We write $M \preceq N$, and also $N \succeq M$, if $(\emptyset, M, N)$ is in the environmental expansion. In examples in this paper we will only need this form of expansion.

In the *bisimulation up to expansion* technique we replace the occurrences of $\mathcal{X}_{\mathcal{E}}$ in the conclusion of the clauses with $\succeq\mathcal{X}_{\mathcal{E}}\preceq$.

Definition 3.31 (*up to expansion*). An environmental relation $\mathcal{X}$ is an *environmental bisimulation up to expansion* if

(1) $M\ \mathcal{X}_{\mathcal{E}}\ N$ implies:
   (a) if $M \longrightarrow M'$ then $N \Longrightarrow N'$ and $M' \succeq\mathcal{X}_{\mathcal{E}}\preceq N'$;
   (b) if $M = V$ then $N \Longrightarrow W$ and $\mathcal{E} \cup \{(V', W')\} \in \mathcal{X}$ for some $V' \preceq V$ and $W' \preceq W$;
   (c) the converse of the above two conditions, on N;
(2) if $\mathcal{E} \in \mathcal{X}$ then for all $(\lambda x.\, P, \lambda x.\, Q) \in \mathcal{E}$ and for all $(M_1, N_1) \in \mathcal{E}^\star$ it holds that $P\{^{M_1}/_x\} \succeq \mathcal{X}_{\mathcal{E}} \preceq Q\{^{N_1}/_x\}$.

As usual with expansion techniques, in the given definition a few occurrences of $\preceq$ could be replaced by $\simeq$ (the two occurrences in clause (2) and the last occurrence in clauses (1.a) and (1.b)).

Relation $\preceq$ is extended to environments as follows: $\mathcal{E} \preceq \mathcal{E}'$ if

$$\begin{array}{l}\mathcal{E} = \cup_{i\in I}\{(V_i, W_i)\} \text{ for some indexing set } I\\ \text{and } \forall i\ \exists V_i', W_i' \text{ with } V_i \preceq V_i', W_i \preceq W_i'\\ \text{and } \mathcal{E}' = \cup_{i\in I}\{(V_i', W_i')\}\}\end{array}$$

LEMMA 3.32. *An environmental bisimulation up to $\preceq$ is included in $\approx$.*

PROOF. Similar to the proof of Lemma 3.12. We use the environmental relation

$$\begin{array}{l}\{(\mathcal{E}, M, N) \text{ s.t. } \mathcal{E} \succeq \mathcal{E}' \text{ and } M \succeq\mathcal{X}_{\mathcal{E}'}\preceq N, \text{ for some } \mathcal{E}'\}\\ \cup\{\mathcal{E} \text{ s.t. } \mathcal{E} \succeq \mathcal{E}' \text{ for some } \mathcal{E}' \in \mathcal{X}\}\end{array}$$

□

Since $\longrightarrow$ is contained in the expansion relation, the up-to-expansion technique subsumes, and is more powerful than, the up to reduction. Still, up-to-reduction is interesting because it is simpler to use, combine with other techniques, and to adapt to richer languages (the reader may consult [Pous 2008; Pous and Sangiorgi] for detailed discussions on the issue of combination of up-to techniques).

Up to context. This technique allows us to cancel a common context in tested terms, requiring instead that only the arguments of such context be pairwise related. We extend the context-closure notation for relations to environmental relations.

We write $M\ \mathcal{X}_{\mathcal{E}}^{(\star)}\ N$ if

—$M = C[M', \widetilde{V}], N = C[N', \widetilde{W}]$, and $[\cdot]_1$ is in redex position in C (Definition 3.14), and $V_i\ \mathcal{E}\ W_i$, and $M'\ \mathcal{X}_{\mathcal{E}}\ N'$;

—or else $M = C[\widetilde{V}], N = C[\widetilde{W}]$, and $V_i\ \mathcal{E}\ W_i$ and $\mathcal{E} \in \mathcal{X}$.

In the first clause, nonvalue arguments for the context may only appear in redex position. While it is possible to give more flexibility to the technique, the formulation above yields a simpler proof, is easy to generalise to other calculi (indeed, we will consider it in all calculi in the paper) and to combine with other up-to techniques, and was sufficient for all examples we have considered.

Definition 3.33 (*up to context*). An environmental relation $\mathcal{X}$ is an *environmental bisimulation up to context* if

(1) $M\ \mathcal{X}_{\mathcal{E}}\ N$ implies:
   (a) if $M \longrightarrow M'$ then $N \Longrightarrow N'$ and $M'\ \mathcal{X}_{\mathcal{E}}^{(\star)}\ N'$;
   (b) if $M = V$ then $N \Longrightarrow W$ and $\mathcal{E} \cup \{(V, W)\} \subseteq \mathcal{E}'^{\widehat{\star}}$, for some $\mathcal{E}' \in \mathcal{X}$;
   (c) the converse of the above two conditions, on N;
(2) if $\mathcal{E} \in \mathcal{X}$ then for all $(\lambda x.\, P, \lambda x.\, Q) \in \mathcal{E}$ and for all $(M_1, N_1) \in \mathcal{E}^\star$ it holds that $P\{^{M_1}/_x\}\ \mathcal{X}_{\mathcal{E}}^{(\star)}\ Q\{^{N_1}/_x\}$.

Note that in clause (1.b) a weakening of the environment is allowed. This flexibility, which may be useful in applications of the techniques, is actually needed in the proof of soundness.

Lemma 3.34. *If $\mathcal{X}$ is an* environmental bisimulation up to context*, then* $\mathcal{X} \subseteq \approx$.

Proof. One shows that

$$\{(\mathcal{E}^{\widehat{\star}}, M, N) \text{ s.t. } M\ \mathcal{X}_{\mathcal{E}}^{(\star)}\ N\}$$
$$\cup\{\mathcal{E}^{\widehat{\star}} \text{ s.t. } \mathcal{E} \in \mathcal{X}\}$$

is a bisimulation up to environment. The details of the proof are similar to (in fact, simpler than) those for the up-to technique shown in Section C for the imperative call-by-value λ-calculus. □

Combinations of up-to. The techniques in this section and in Section 3.4 can be combined together, in the expected manner. As an example, we present the definition of up to expansion and context. Another example of combination of techniques can be found in Section 5 (up to environment, reduction and context).

Definition 3.35 (*up to expansion and context*). An environmental relation $\mathcal{X}$ is an *environmental bisimulation up to expansion and context* if

(1) $M\ \mathcal{X}_{\mathcal{E}}\ N$ implies:
   (a) if $M \longrightarrow M'$ then $N \Longrightarrow N'$ and $M' \succeq \mathcal{X}_{\mathcal{E}}^{(\star)} \preceq N'$;
   (b) if $M = V$ then $N \Longrightarrow W$ and $\mathcal{E} \cup \{(V', W')\} \subseteq \mathcal{E}'^{\star}$, for some $\mathcal{E}' \in \mathcal{X}$, and some V', W' with $V' \preceq V$ and $W' \preceq W$;
   (c) the converse of the above two conditions, on N;

(2) if $\mathcal{E} \in \mathcal{X}$ then for all $(\lambda x.\, P, \lambda x.\, Q) \in \mathcal{E}$ and for all $(M_1, N_1) \in \mathcal{E}^\star$ it holds that $P\{^{M_1}/_x\} \succeq \mathcal{X}_{\mathcal{E}}^{(\star)} \preceq Q\{^{N_1}/_x\}$.

LEMMA 3.36. *If $\mathcal{X}$ is an* environmental bisimulation up to expansion and context*, then $\mathcal{X} \subseteq \approx$.*

PROOF. One shows that

$$\{(\widehat{\mathcal{E}^\star}, M, N) \text{ s.t. } \exists \mathcal{E}' \text{ with } \mathcal{E} \succeq \mathcal{E}' \text{ and } M \succeq \mathcal{X}_{\mathcal{E}'}^{(\star)} \preceq N\}$$
$$\cup \{\widehat{\mathcal{E}^\star} \text{ s.t. } \exists \mathcal{E}' \text{ with } \mathcal{E} \succeq \mathcal{E}' \text{ and } \mathcal{E}' \in \mathcal{X}\}$$

is a bisimulation up to environment. □

3.7 Contextual Equivalence

We now show that the environmental bisimulation is sound and complete with respect to the standard contextual equivalence. We write $M \Downarrow$ if $M \Longrightarrow \lambda x.\, N$ for some x and N.

Definition 3.37 (*contextual equivalence*). Terms M and N are *contextually equivalent*, written $M \equiv N$, if, for any context C such that $C[M]$ and $C[N]$ are closed, $C[M] \Downarrow$ iff $C[N] \Downarrow$.

LEMMA 3.38. *$\equiv$ is an equivalence relation.*

PROOF. This follows immediately from the definition. □

LEMMA 3.39. $\equiv^\star = \equiv$.

PROOF. $\equiv \subseteq \equiv^\star$ is trivial. To show $\equiv^\star \subseteq \equiv$, suppose $(M, N) \in \equiv^\star$. Then $M = C_1[M_1, \ldots, M_n]$ and $N = C_1[N_1, \ldots, N_n]$ for some $C_1, M_1, \ldots, M_n, N_1, \ldots, N_n$ such that $M_i \equiv N_i$ for $i = 1, \ldots, n$. Let C be any context (with one hole). Then,

$$C[M] = C[C_1[M_1, \ldots, M_n]] \Downarrow \textit{ iff } C[C_1[N_1, M_2, \ldots, M_n]] \Downarrow \textit{ iff } \cdots$$
$$\textit{iff } C[C_1[N_1, \ldots, N_{n-1}, M_n]] \Downarrow \textit{ iff } C[C_1[N_1, \ldots, N_n]] (= C[N]) \Downarrow.$$

Hence, we have $M \equiv N$ as required. □

LEMMA 3.40. *If $M \longrightarrow N$, then $M \equiv N$.*

PROOF. In this proof, $\longrightarrow^\star$ is the context closure of the relation $\longrightarrow$.

To show $M \longrightarrow N$ and $C[M] \Downarrow$ imply $C[N] \Downarrow$, it suffices to prove that $M \longrightarrow^\star N$ and $M \Longrightarrow V$ imply $N \Longrightarrow W$ and $V \longrightarrow^\star W$ (note here that $M \longrightarrow N$ implies $C[M] \longrightarrow^\star C[N]$). We show this by induction on the length of the reductions $M \Longrightarrow V$.

If $M = V$, then the result holds for $W = N$. Otherwise, either $M \longrightarrow N$ or $M = M_1 M_2$ and $N = N_1 N_2$, with $M_1 \longrightarrow^\star N_1$ and $M_2 \longrightarrow^\star N_2$ holds. In the former case, the result follows immediately from the fact that the reduction is deterministic. In the latter case, by the assumption $M \Longrightarrow V$, we have

$$M_1 \Longrightarrow \lambda x.\, M_3 \qquad M_3\{^{M_2}/_x\} \Longrightarrow V.$$

By the induction hypothesis, there exists N_3 such that

$$N_1 \Longrightarrow \lambda x.\, N_3 \qquad \lambda x.\, M_3 \longrightarrow^\star \lambda x.\, N_3.$$

From the conditions $M_2 \longrightarrow^\star N_2$ and $\lambda x.M_3 \longrightarrow^\star \lambda x.N_3$, we obtain $M_3\{^{M_2}/_x\} \longrightarrow^\star N_3\{^{N_2}/_x\}$. By the induction hypothesis (note that the reductions $M_3\{^{M_2}/_x\} \Longrightarrow V$ is shorter than $M \Longrightarrow V$), there exists W such that $N_3\{^{N_2}/_x\} \Longrightarrow W$ and $V \longrightarrow^\star W$. Furthermore, we have $N \Longrightarrow W$ as required.

To show the converse, it suffices to prove that $M \longrightarrow^\star N$ and $N \Longrightarrow W$ imply $M \Longrightarrow V$ and $V \longrightarrow^\star W$. This can be proved in a similar manner, by induction on the length of reductions $N \Longrightarrow W$, with case analysis on the structure of C. □

THEOREM 3.41 (COMPLETENESS OF BISIMULATION). *If $M \equiv N$, then $M \simeq N$.*

PROOF. It is sufficient to show that $\mathcal{X}$ is a big-step environmental bisimulation, appealing to Lemma 3.28, where

$$\mathcal{X} \stackrel{\text{def}}{=} \{(\widehat{\equiv}, M, N) \text{ s.t. } M \equiv N\} \cup \widehat{\equiv}.$$

Then the result follows from weakening (Lemma 3.5). We consider the first clause (1) of the bisimulation. Suppose $M \equiv N$ and $M \Longrightarrow \lambda x.M'$. Since $M \equiv N$, there exists N' such that $N \Longrightarrow \lambda x.N'$. We need $\lambda x.M' \equiv \lambda x.N'$, which we get by the transitivity of $\equiv$ (Lemma 3.38), Lemma 3.40, and the assumptions, thus: $\lambda x.M' \equiv M \equiv N \equiv \lambda x.N'$.

Consider now clause (2) of bisimulation, and take $\lambda x.M' \equiv \lambda x.N'$ and $(M_1, N_1) \in \equiv^\star$. We need to show $M'\{^{M_1}/_x\} \equiv N'\{^{N_1}/_x\}$. As $\lambda x.M' \equiv \lambda x.N'$, by the congruence of $\equiv$ (Lemma 3.39), we have $(\lambda x.M')M_1 \equiv (\lambda x.N')N_1$. By applying Lemma 3.40, we get $M'\{^{M_1}/_x\} \equiv (\lambda x.M')M_1 \equiv (\lambda x.N')N_1 \equiv N'\{^{N_1}/_x\}$. By the transitivity of $\equiv$, we obtain $M'\{^{M_1}/_x\} \equiv N'\{^{N_1}/_x\}$ as required. □

THEOREM 3.42 (SOUNDNESS OF BISIMULATION). *If $M \simeq N$, then $M \equiv N$.*

PROOF. Suppose $M \simeq N$. By Corollary 3.24, we have $C[M] \simeq C[N]$ for any context C. By the definition of $\simeq$, $C[M] \Downarrow$ if and only if $C[N] \Downarrow$. Hence, we have $M \equiv N$ as required. □

3.8 Example

This example gives the proof of the equivalence between the two fixed-point combinators:

$$Y \stackrel{\text{def}}{=} \lambda y.\, y(Dy(Dy))$$
$$\Theta \stackrel{\text{def}}{=} \Delta\Delta$$

where

$$\Delta \stackrel{\text{def}}{=} \lambda x.\, \lambda y.\, (y(xxy))$$
$$D \stackrel{\text{def}}{=} \lambda y.\, \lambda x.\, y(xx)$$

We establish $Y \simeq \Theta$ by showing that the following relation $\mathcal{X}$ is an environmental bisimulation up to expansion and context, and then applying Lemma 3.5.

$$\mathcal{X} \stackrel{\text{def}}{=} \{(\mathcal{E}, Y, \Theta), (\mathcal{E}, Y, \Theta_1), \mathcal{E}\} \qquad \mathcal{E} \stackrel{\text{def}}{=} \{(Y, \Theta_1)\} \qquad \Theta_1 \stackrel{\text{def}}{=} \lambda y.\, (y(\Delta\Delta y))$$

First we note that, for any term M,

$$DM(DM) \succeq YM. \tag{3}$$

This holds because $DM(DM) \Longrightarrow_2 M(DM(DM))$ and $YM \Longrightarrow_1 M(DM(DM))$. Since $\Theta \longrightarrow \Theta_1$, we also have:

$$\Theta \succeq \Theta_1, \tag{4}$$

We now check the bisimilarity clause (2) of Definition 3.35 on the pair (Y, Θ_1). Consider now any argument $M\ \mathcal{E}^\star\ N$ for Y and Θ_1. The results are $M(DM(DM))$ and $N(\Delta\Delta N)$, respectively. Now, by (3), it holds that

$$M(DM(DM)) \succeq M(YM)$$

and by (4),

$$N(\Delta\Delta N) = N(\Theta N) \succeq N(\Theta_1 N),$$

and we are done, since $M(YM)\ \mathcal{E}^\star\ N(\Theta_1 N)$.

4. CALL-BY-VALUE λ-CALCULUS

The *one-step call-by-value reduction relation* $\longrightarrow \subseteq \Lambda^\bullet \times \Lambda^\bullet$ is defined by these rules.

$$\beta_{\mathrm{v}}: \quad (\lambda x.\, M)V \longrightarrow M\{^V/_x\}$$

$$\mu: \quad \frac{M \longrightarrow M'}{MN \longrightarrow M'N} \qquad \nu_{\mathrm{v}}: \quad \frac{N \longrightarrow N'}{VN \longrightarrow VN'}$$

We highlight what changes in the theory for call-by-name of the previous sections. Recall that for a relation $\mathcal{R}$ we write $\mathcal{R}^{\hat{\star}}$ for the subset of $\mathcal{R}^\star$ that only relate pairs of values.

—The input for two functions must be values. Therefore, in the definition of environmental bisimulation, the input terms M_1 and N_1 should be in $\mathcal{E}^{\hat{\star}}$ (rather than $\mathcal{E}^\star$). A similar modification on the quantification over inputs of functions is needed in all definitions of bisimulations and up-to techniques.

—In the grammar for evaluation contexts, we add the production VC.

With these modifications, all definitions and results in Section 3 are valid for call-by-value. The structure of the proof also remains the same, with the expected differences in technical details due to the change in reduction strategy.

Example 4.1. This example uses a simply-typed call-by-value λ-calculus extended with integers, an operator for subtraction ($\hat{-}$), a conditional, and a fixed-point operator Y. The reduction rule for Y is $YV \longrightarrow V(\lambda x.\, YVx)$. As mentioned in Section 2, it is straightforward to accommodate such additions in the theory developed. (We could also encode arithmetic into the untyped calculus and adapt the example, but it would become harder to read.) Let P, Q be the terms

$$P \stackrel{\text{def}}{=} \lambda f.\lambda g.\lambda x.\lambda y.\, \texttt{if}\ x = 0\ \texttt{then}\ y\ \texttt{else}\ g(f\,g\,(x \hat{-} 1)\,y)$$

$$Q \stackrel{\text{def}}{=} \lambda f.\lambda g.\lambda x.\lambda y.\, \texttt{if}\ x = 0\ \texttt{then}\ y\ \texttt{else}\ f\,g\,(x \hat{-} 1)\,(g\,y)$$

Let $F_1 \stackrel{\text{def}}{=} \lambda z.\, Y\,P\,z$ and $F_2 \stackrel{\text{def}}{=} \lambda z.\, Y\,Q\,z$.

The terms $F_1\,g\,n\,m$ and $F_2\,g\,n\,m$ (where g is a function value from integers to integers and n, m are integers) computes $g^n(m)$ (that is, n applications of g starting from m) if $n \geq 0$, diverge otherwise. In both cases, however, the computations made are different. We show $F_1\,g\,n\,m \simeq F_2\,g\,n\,m$ using an up-to technique for environmental bisimulations.

For this, we use the following relation $\mathcal{X}$:

$$\begin{array}{l}\{(\emptyset, g^r(F_1\,g\,n\,m), F_2\,g\,n\,(g^r(m))) \mid \\ \quad r, m, n \in \mathbf{Z}, r \geq 0, \text{ and} \\ \quad g \text{ is a closed value of type } \mathsf{int} \to \mathsf{int}\}.\end{array}$$

We show that $\mathcal{X}$ is an environmental bisimulation, up-to expansion and context.

Let us consider the pair $(g^r(F_1\,g\,n\,m), F_2\,g\,n\,(g^r(m)))$. If $n = 0$, then we have:

$$g^r(F_1\,g\,n\,m) \longrightarrow\!\!\!\!\Longrightarrow g^r(m)$$

and

$$\begin{array}{rcl} F_2\,g\,n\,(g^r(m)) & \longrightarrow & Y\;Q g\,0\,(g^r(m)) \\ & \Longrightarrow & (\lambda y.\,\mathtt{if}\ 0 = 0\ \mathtt{then}\ y\ \mathtt{else}\ \cdots)(g^r(m)) \\ & \succeq & (\lambda y.\,y)(g^r(m)) \\ & \succeq & g^r(m) \end{array}$$

(where the terms for the dots does not contribute to the final outcome). So, the required condition holds, as $g^r(m)\ \mathcal{X}_\emptyset^\star\ g^r(m)$.

If $n \neq 0$, then we have

$$\begin{array}{rcl} g^r(F_1\,g\,n\,m) & \longrightarrow\!\!\!\!\Longrightarrow & g^r(g(F_1\,g\,(n\hat{-}1)\,m)) \\ & \succeq & g^{r+1}(F_1\,g\,(n-1)\,m). \end{array}$$

and

$$\begin{array}{rcl} F_2\,g\,n\,(g^r(m)) & \longrightarrow & Y\;Q g\,n\,(g^r(m)) \\ & \Longrightarrow & (\lambda y.\,\mathtt{if}\ n = 0\ \mathtt{then}\ y\ \mathtt{else}\ F_2\,g\,(n\hat{-}1)\,(g\,y))(g^r(m)) \\ & \succeq & (\lambda y.\,F_2\,g\,(n-1)\,(g\,y))(g^r(m)) \\ & \succeq & F_2\,g\,(n-1)\,(g^{r+1}(m)) \end{array}$$

(the first $\succeq$ comes from the fact that y is not copied inside the function F_2). We are done, since

$$(g^{r+1}(F_1\,g\,(n-1)\,m), F_2\,g\,(n-1)\,(g^{r+1}(m))) \in \mathcal{X}_\emptyset.$$

The example above makes use of key features of our method: the ability to compare terms in the middle of evaluations, and (some of) its up-to techniques.

5. IMPERATIVE CALL-BY-VALUE λ-CALCULUS

In this section, we study the addition of imperative features (higher-order references, that we call locations). Finding powerful techniques for reasoning on imperative higher order languages is known to be a hard problem. We carry out our study on Koutavas and Wand's language [Koutavas and Wand 2006b]: a call-by-value λ-calculus, with locations and a few other auxiliary operators.

5.1 Syntax and Reduction Rules

The language is a standard call-by-value λ-calculus with constants, tuples, creation of new locations, dereferencing, and assignment. We use s, t to range over stores, that is, finite mappings from locations to closed values and l, k over locations. Then $s[l \leftarrow V]$ is the update of s (possibly an extension of s if l is not in the domain of s). We write $s \uplus s'$ for the union of the two stores when $\mathsf{dom}(s)$ and $\mathsf{dom}(s')$ are disjoint. We write $\emptyset$ for the empty store, and $\mathsf{dom}(s)$ for the domain of s. We often write $M_1; M_2$ for $(\lambda x.\, M_2)M_1$ when $x \notin \mathsf{fv}(M_2)$. The syntax of terms is given as follows.

$M ::=$	x	variables
	c	constants
	$\lambda x.\, M$	functions
	$M_1 M_2$	applications
	$\mathtt{op}(M_1, \ldots, M_n)$	primitive operations
	$\mathtt{if}\ M_1\ \mathtt{then}\ M_2\ \mathtt{else}\ M_3$	if-then-else
	$\#_i(M)$	projection
	$(M_1, \ldots, M_n)$	tuples
	l	locations
	$\nu x\, M$	new locations
	$!M$	dereferencing
	$M_1 := M_2$	assignments

Values are

$$V ::= c \mid \lambda x.\, M \mid (V_1, \ldots, V_n) \mid l.$$

We assume that the set of primitive operations contains the equality function on constants, and that the set of constants include the special symbol $\star$, representing the unit value (i.e., nullary tuple).

The formal definition of the small-step reduction relation is given below. We assume that the semantics of primitive operations are already given by the function Prim.

$$\langle s\,;\, (\lambda x.\, M)V\rangle \longrightarrow \langle s\,;\, M\{^V/_x\}\rangle$$

$$\langle s\,;\, \mathtt{if}\ \mathrm{true}\ \mathtt{then}\ M_1\ \mathtt{else}\ M_2\rangle \longrightarrow \langle s\,;\, M_1\rangle$$

$$\langle s\,;\, \mathtt{if}\ \mathrm{false}\ \mathtt{then}\ M_1\ \mathtt{else}\ M_2\rangle \longrightarrow \langle s\,;\, M_2\rangle$$

$$\langle s\,;\, \#_i(\widetilde{V})\rangle \longrightarrow \langle s\,;\, V_i\rangle$$

$$\langle s\,;\, l := V\rangle \longrightarrow \langle s[l \leftarrow V]\,;\, \star\rangle$$

$$\frac{\mathsf{Prim}(\mathtt{op}, \widetilde{c}) = c'}{\langle s\,;\, \mathtt{op}(\widetilde{c})\rangle \longrightarrow \langle s\,;\, c'\rangle}$$

$$\frac{l \notin \mathsf{dom}(s)}{\langle s\,;\, \nu x\, M\rangle \longrightarrow \langle s[l \leftarrow 0]\,;\, M\{^l/_x\}\rangle}$$

$$\frac{s(l) = V}{\langle s\,;\, !l\rangle \longrightarrow \langle s\,;\, V\rangle}$$

$$\frac{C \text{ is an evaluation context} \quad \langle s\,;\, M\rangle \longrightarrow \langle s'\,;\, M'\rangle}{\langle s\,;\, C[M]\rangle \longrightarrow \langle s'\,;\, C[M']\rangle},$$

where evaluation contexts are

$$C \;:=\; [\cdot] \mid CM \mid VC \mid \mathtt{op}(\widetilde{c}, C, \widetilde{M}) \mid (\widetilde{V}, C, \widetilde{M}) \mid \#_i C \mid$$
$$\mathtt{if}\ C\ \mathtt{then}\ M_1\ \mathtt{else}\ M_2 \mid\ !C \mid C := M \mid l := C.$$

In this section, $\mathcal{E}^\star$ is the closure of a relation under *location-free* contexts (i.e., contexts without free locations). So, for example, (l, l) is in $\mathcal{E}^\star$ only if $(l, l) \in \mathcal{E}$. We discuss in Section 5.3 how to take into account arbitrary contexts. Intuitively, free locations should appear as related values in the environment, and thus available to the observer; as shown in the section, this condition is necessary for soundness.

5.2 Environmental Bisimilarity and Some of Its Basic Properties

Let $\mathcal{E}, \mathcal{F}$ range over relations on values. The notion of environmental relation is modified to accommodate stores, which are needed to run terms. Thus the elements of an environmental relation are now of the form $(\mathcal{E}, \langle s\,;\, M\rangle, \langle t\,;\, N\rangle)$ or $(\mathcal{E}, s, t)$. Further, the relation must be *well-formed*, in the sense that in a tuple $(\mathcal{E}, \langle s\,;\, M\rangle, \langle t\,;\, N\rangle)$ the free locations in M and s and all left components of pairs in $\mathcal{E}$ (resp. N and t and all right components of pairs in $\mathcal{E}$) must appear in the domain of s (resp. t). Similarly for an element $(\mathcal{E}, s, t)$. We write $\langle s\,;\, M\rangle\ \mathcal{X}_\mathcal{E}\ \langle t\,;\, N\rangle$ when $(\mathcal{E}, \langle s\,;\, M\rangle, \langle t\,;\, N\rangle) \in \mathcal{X}$.

Definition 5.1. An environmental relation $\mathcal{X}$ is an *environmental bisimulation* if

(1) $\langle s\,;\, M\rangle\ \mathcal{X}_\mathcal{E}\ \langle t\,;\, N\rangle$ implies:
  (a) if $\langle s\,;\, M\rangle \longrightarrow \langle s'\,;\, M'\rangle$ then $\langle t\,;\, N\rangle \Longrightarrow \langle t'\,;\, N'\rangle$ and $\langle s'\,;\, M'\rangle\ \mathcal{X}_\mathcal{E}\ \langle t'\,;\, N'\rangle$;
  (b) if $M = V$ then $\langle t\,;\, N\rangle \Longrightarrow \langle t'\,;\, W\rangle$ and $(\mathcal{E} \cup \{(V, W)\}, s, t') \in \mathcal{X}$;
  (c) the converse of the above two conditions, on N;
(2) if $(\mathcal{E}, s, t) \in \mathcal{X}$ then for all $(V, W) \in \mathcal{E}$ we have the following.
  (a) $V = c$ implies $W = c$;
  (b) $V = (V_1, \ldots, V_n)$ implies $W = (W_1, \ldots, W_n)$ and $(\mathcal{E} \cup \{(V_1, W_1), \ldots, (V_n, W_n)\}, s, t) \in \mathcal{X}$;
  (c) for all fresh l, l', we have $(\mathcal{E} \cup \{(l, l')\}, s[l \leftarrow 0], t[l' \leftarrow 0]) \in \mathcal{X}$;
  (d) if $V = l$ then $W = l'$, for some l', and moreover,
    i. $(\mathcal{E} \cup \{(s(l), t(l'))\}, s, t) \in \mathcal{X}$;
    ii. for all $(V_1, W_1) \in \mathcal{E}^{\widehat{\star}}$, we have $(\mathcal{E}, s[l \leftarrow V_1], t[l' \leftarrow W_1]) \in \mathcal{X}$;
  (e) if $V = \lambda x.\, P$ then $W = \lambda x.\, Q$ and for all $(V_1, W_1) \in \mathcal{E}^{\widehat{\star}}$ it holds that $\langle s\,;\, P\{^{V_1}/_x\}\rangle\ \mathcal{X}_\mathcal{E}\ \langle t\,;\, Q\{^{W_1}/_x\}\rangle$;
  (f) the converse of the above five conditions, on W.

We write $\approx$ for environmental bisimilarity, the union of all bisimulations. We write $\langle s\,;\, M\rangle \approx_\mathcal{E} \langle t\,;\, N\rangle$ if $(\mathcal{E}, \langle s\,;\, M\rangle, \langle t\,;\, N\rangle) \in \approx$.

The definition of the environmental bisimulation above may seem much more complex than the definition for pure λ-calculi discussed so far. That is due to the presence of stores and additional language constructs. However, the essential structure of the definition is the same as the case for pure λ-calculi: the conditions (1.a)–(1.c) are the same except for the existence of stores; the conditions (2.a)–(2.f) state that the top-level language constructors of related values are the same (in (2.a) they are constants, in (2.b) tuples, in (2.d) locations, in (2.e) functions; (2.c) accounts for the possibility of extending the stores), and that for every operation on related values, the results are also related.

All the basic properties for environmental bisimulations in Section 3.3 remain valid with due adjustments. For instance, we have the following.

LEMMA 5.2 (WEAKENING). *$\langle s\,;M\rangle \approx_{\mathcal{E}} \langle t\,;N\rangle$ and $\mathcal{E}' \subseteq \mathcal{E}$ implies $\langle s\,;M\rangle \approx_{\mathcal{E}'} \langle t\,;N\rangle$. Similarly, if $(\mathcal{E},s,t)\in\approx$ and $\mathcal{E}'\subseteq\mathcal{E}$ then also $(\mathcal{E}',s,t)\in\approx$.*

We give more details for the congruence results, however, as the presence of locations introduces some subtleties.

5.3 Compositionality Properties

As in pure λ-calculi, the main results are the congruence for values, and for all terms in evaluation contexts, the correspondent of Lemmas 3.15 and 3.16.

LEMMA 5.3 (CONGRUENCE FOR VALUES). *If $\langle s\,;V\rangle \approx_{\mathcal{E}} \langle t\,;W\rangle$, then $\langle s\,;C[V]\rangle \approx_{\mathcal{E}} \langle t\,;C[W]\rangle$ for any location-free context C.*

LEMMA 5.4 (CONGRUENCE FOR ALL TERMS IN EVALUATION CONTEXTS). *$\langle s\,;M\rangle \approx_{\mathcal{E}} \langle t\,;N\rangle$ implies $\langle s\,;C[M]\rangle \approx_{\mathcal{E}} \langle t\,;C[N]\rangle$ for any location-free evaluation context C.*

As for the pure λ-calculus, the two results are proved at the same time, using a single bisimulation and proceeding by induction on the contexts. The schema of the proof is the same as for pure λ-calculi; the details can be found in Section B.

In Lemma 5.3, the restriction that C is location-free is important (especially concerning those locations in the domain of s and t). For example, $\langle [l=0]\,;\lambda x.l:=0\rangle \approx_{\mathcal{E}} \langle [l=0]\,;\lambda x.l:=1\rangle$ holds for $\mathcal{E}=\{(\lambda x.l:=0,\lambda x.l:=1)\}$ (intuitively because the effect on updates on l is invisible if the observer can access locations only via $\lambda x.l:=0$ and $\lambda x.l:=1$). Let $C=([\cdot]_1\star;\texttt{if}\ !l=0\ \texttt{then}\ 0\ \texttt{else}\ \Omega)$ (where Ω is a divergent term). Obviously, $\langle [l=0]\,;C[\lambda x.l:=0]\rangle \approx_{\mathcal{E}} \langle [l=0]\,;C[\lambda x.l:=1]\rangle$ does not hold.

To extend Lemma 5.3 to arbitrary contexts (that may contain free locations), we have to make sure that the locations that appear free in the context are related in the bisimulation. One way of expressing this is the result below. It follows from the "polyadic-context" version of Lemma 5.3: if $\langle s\,;V_i\rangle \approx_{\mathcal{E}} \langle t\,;W_i\rangle$, for $i=1,\ldots,n$, then $\langle s\,;C[V_1,\ldots,V_n]\rangle \approx_{\mathcal{E}} \langle t\,;C[W_1,\ldots,W_n]\rangle$ for any location-free context C. This extension does not need a separate proof, as the proof of Lemma 5.3 in Section B already requires contexts with multiple holes.

COROLLARY 5.5. *If* $\langle s\,;\,V\rangle \approx_{\mathcal{E}} \langle t\,;\,W\rangle$, *and* $(l_i, l_i) \in \mathcal{E}$ *for all* $l_i \in \widetilde{l}$, *then also* $\langle s\,;\,C[V]\rangle \approx_{\mathcal{E}} \langle t\,;\,C[W]\rangle$ *for any context* C *with free locations in* $\widetilde{l}$.

In a similar manner, the result of Lemma 5.4 can be extended to contexts with free locations.

In contrast, the correspondent of Corollary 3.24 does not hold, that is, we cannot replace evaluation contexts with arbitrary contexts in Lemma 5.4. For example, let M and N be $\mathtt{if}\ !l = 0\ \mathtt{then}\ l := 1\ \mathtt{else}\ \Omega$ and $l := 1$ respectively. Then, $\langle [l = 0]\,;\,M\rangle \approx_{\emptyset} \langle [l = 0]\,;\,N\rangle$ but $\langle [l = 0]\,;\,C[M]\rangle \not\approx_{\emptyset} \langle [l = 0]\,;\,C[N]\rangle$ for $C = [\cdot]_1; [\cdot]_1$.

Given bisimilar terms M and N, to show that $C[M]$ and $C[N]$ are bisimilar for any context C, we need to show that the thunk of M and N are bisimilar.

COROLLARY 5.6. *If* $\langle s\,;\,\lambda.\,M\rangle \approx_{\mathcal{E}} \langle t\,;\,\lambda.\,N\rangle$, *then* $\langle s\,;\,C[M]\rangle \approx_{\mathcal{E}} \langle t\,;\,C[N]\rangle$ *for any location-free context* C.

PROOF. The result follows from Lemma 5.3 in the same way as in pure λ-calculi Lemma 3.21 followed from Lemma 3.15 and validity of β-conversion. The latter is the property that $\langle s_1\,;\,C[(\lambda x.\,M_1)N_1]\rangle \approx_{\emptyset} \langle s_1\,;\,C[M_1\{^{N_1}/_x\}]\rangle$ for any location-free context C, store s_1, and term $(\lambda x.\,M_1)N_1$, and is proved similarly to Lemma A.9 for pure λ-calculi (a similar form of β-conversion is considered in Lemma 5.8 below). □

To obtain a result of congruence that also allows free locations in contexts, we make sure that the free locations in the tested terms are made available, as values, to the observer. We write $s_{\widetilde{l}}$ for a store that is closed, with domain $\widetilde{l}$, and well-formed (i.e., all its free locations must be in $\widetilde{l}$).

LEMMA 5.7. *Suppose* $\mathsf{Loc}(M) \cup \mathsf{Loc}(N) = \{\widetilde{l}\}$. *If,*

$$\langle [\widetilde{l} = 0]\,;\,\lambda.\,M\rangle \approx_{\{(\widetilde{l},\widetilde{l})\}} \langle [\widetilde{l} = 0]\,;\,\lambda.\,N\rangle$$

then for all contexts C, *with* $\mathsf{Loc}(C) \subseteq \{\widetilde{l}\}$, *and for all* $s_{\widetilde{l}}$

$$\langle s_{\widetilde{l}}\,;\,C[M]\rangle \approx_{\{(\widetilde{l},\widetilde{l})\}} \langle s_{\widetilde{l}}\,;\,C[N]\rangle\,.$$

To prove the lemma we first prove a simple instance of it, corresponding to the case when M and N are β-convertible (the proof is similar to that of Lemma A.9).

LEMMA 5.8. *Suppose* $M = (\lambda x.\,M')M''$ *and* $N = M'\{^{M''}/_x\}$, *with* $\mathsf{Loc}(M) \subseteq \{\widetilde{l}\}$. *Then for all for all contexts* C, *with* $\mathsf{Loc}(C) \subseteq \{\widetilde{l}\}$, *and for all* $s_{\widetilde{l}}$, *we have* $\langle s_{\widetilde{l}}\,;\,C[M]\rangle \approx_{\{(\widetilde{l},\widetilde{l})\}} \langle s_{\widetilde{l}}\,;\,C[M']\rangle$.

We can now prove Lemma 5.7.

PROOF OF LEMMA 5.7. Take a context C with free locations in $\widetilde{l}$, and let $C' \stackrel{\text{def}}{=} C[[\cdot]\star]$ (that is, we impose that the hole should contain a thunk). Applying Lemma 5.3 (where we view C' as obtained from a location-free context in which the locations $\widetilde{l}$ were the arguments of additional holes), we get

$$\langle s_{\widetilde{l}}\,;\,C'[\lambda.\,M]\rangle \approx_{\{(\widetilde{l},\widetilde{l})\}} \langle s_{\widetilde{l}}\,;\,C'[\lambda.\,N]\rangle. \tag{5}$$

Using Lemma 5.8 we derive

$$\langle s_{\widetilde{l}}\,;\, C'[\lambda.\, M]\rangle \approx_{\{(\widetilde{l},\widetilde{l})\}} \langle s_{\widetilde{l}}\,;\, C[M]\rangle$$

and similarly for $C'[\lambda.\, N]$. Thus from (5) we get, by transitivity,

$$\langle s_{\widetilde{l}}\,;\, C[M]\rangle \approx_{(\{\widetilde{l},\widetilde{l})\}} \langle s_{\widetilde{l}}\,;\, C[N]\rangle.$$

□

5.4 Contextual Equivalence

We recall the standard definition of contextual equivalence.

Definition 5.9 (*contextual equivalence*). M and N are *contextually equivalent*, written $M \equiv N$, if, for any store s and context C such that $\langle s\,;\, C[M]\rangle$ and $\langle s\,;\, C[N]\rangle$ are well-formed, $\langle s\,;\, C[M]\rangle \Downarrow$ if and only if $\langle s\,;\, C[N]\rangle \Downarrow$.

The theorems below state that, for closed values, the bisimulation is sound and complete with respect to contextual equivalence.

THEOREM 5.10 (SOUNDNESS AND COMPLETENESS). *Suppose that V and W are closed and* $\mathsf{Loc}(V) \cup \mathsf{Loc}(W) = \{\widetilde{l}\}$. *Then we have $V \equiv W$ if and only if* $\langle[\widetilde{l} = \widetilde{0}]\,;\, (V, \widetilde{l})\rangle \approx_{\emptyset} \langle[\widetilde{l} = \widetilde{0}]\,;\, (W, \widetilde{l})\rangle$.

To prove the completeness (the "only if" part of Theorem 5.10), we prepare the following lemma.

LEMMA 5.11. *The following relation $\mathcal{X}$ is an environmental bisimulation.*

$$\begin{aligned}\mathcal{X} \stackrel{def}{=}\ & \{(\mathcal{E}, s, t) \text{ s.t. } \forall (M, N) \in \mathcal{E}^{\star}.\, \langle s\,;\, M\rangle \Downarrow \textit{ iff } \langle t\,;\, N\rangle \Downarrow\} \\ & \cup \{(\mathcal{E}, \langle s\,;\, M\rangle, \langle t\,;\, N\rangle) \text{ s.t. } (\langle s\,;\, M\rangle \Downarrow \textit{ iff } \langle t\,;\, N\rangle \Downarrow) \\ & \quad \wedge \forall s', t', V, W.\, (\langle s\,;\, M\rangle \Longrightarrow \langle s'\,;\, V\rangle \wedge \langle t\,;\, N\rangle \Longrightarrow \langle t'\,;\, W\rangle \\ & \qquad \textit{implies } \forall (M_1, N_1) \in (\mathcal{E} \cup \{(V, W)\})^{\star}.\, (\langle s'\,;\, M_1\rangle \Downarrow \textit{ iff } \langle t'\,;\, N_1\rangle \Downarrow)\}\end{aligned}$$

PROOF. We check each condition of the bisimulation.

(1.a) If $\langle s\,;\, M\rangle \longrightarrow \langle s'\,;\, M'\rangle$, then let $t' = t$ and $N = N'$. Then we have $\langle s'\,;\, M'\rangle\ \mathcal{X}_{\mathcal{E}}\ \langle t'\,;\, N'\rangle$ as required. (Notice that $\langle s\,;\, M\rangle \Longrightarrow \langle s''\,;\, V\rangle$ *iff* $\langle s'\,;\, M'\rangle \Longrightarrow \langle s''\,;\, V\rangle$ modulo renaming of locations.)

(1.b) If $M = V$, then by the construction of $\mathcal{X}$, there exist t' and W such that $\langle t\,;\, N\rangle \Longrightarrow \langle t'\,;\, W\rangle$. Again, by the construction of $\mathcal{X}$, we have $(\mathcal{E} \cup \{(M, W)\}, s, t') \in \mathcal{X}$.

(2.a) Let $C = \texttt{if } [\cdot] = c \texttt{ then } 0 \texttt{ else } \Omega$. Then by the construction of $\mathcal{X}$, $\langle s\,;\, C[V]\rangle \Downarrow$ *iff* $\langle t\,;\, C[W]\rangle \Downarrow$, which implies $V = c$ *iff* $W = c$.

(2.b) Let $C_i = \#_i[\cdot]$. By the construction of $\mathcal{X}$, $C_i[V] \Downarrow$ *iff* $C_i[W] \Downarrow$ for any i, which implies that V is a n-tuple if and only if W is a n-tuple.
Let $\mathcal{E}' = \mathcal{E} \cup \{(V_1, W_1), \ldots, (V_n, W_n)\}$. To check that $(\mathcal{E}', s, t) \in \mathcal{X}$, suppose $(M, N) \in \mathcal{E}'^{\star}$, i.e. $M = C[\widetilde{V}', V, V_1, \ldots, V_n]$ and $N = C[\widetilde{W}', W, W_1, \ldots, W_n]$ for some C, where $(\widetilde{V}', \widetilde{W}') \in \mathcal{E}$. Let C' be:

$$(\lambda y.\, (\lambda z_1. \cdots \lambda z_n.\, C[[\widetilde{\ }], y, z_1, \ldots, z_n])(\#_1[y]) \cdots (\#_n[y]))[\ ].$$

Then, $\langle s\,;\, C'[\widetilde{V}', V]\rangle \Downarrow\ \langle t\,;\, C'[\widetilde{W}', W]\rangle \Downarrow$, which implies that $\langle s\,;\, M\rangle \Downarrow$ if and only if $\langle t\,;\, N\rangle \Downarrow$, as required.

(2.c) Assuming that l and l' are fresh, let $\mathcal{E}' = \mathcal{E} \cup \{(l, l')\}$. We need to show $(\mathcal{E}', s[l = 0], t[l' = 0]) \in \mathcal{X}$. Suppose $(M, N) \in \mathcal{E}'^{\star}$. So, there exists a context C such that $M = C[\widetilde{V}, l]$ and $N = C[\widetilde{W}, l']$ with $(\widetilde{V}, \widetilde{W}) \in \mathcal{E}$. Let $M' = \nu x\, C[\widetilde{V}, x]$ and $N' = \nu x\, C[\widetilde{W}, x]$ where x is a fresh variable. Since $(M', N') \in \mathcal{E}^{\star}$, we have $\langle s\,;\, M'\rangle \Downarrow$ *iff* $\langle t\,;\, N'\rangle \Downarrow$. Since $\langle s\,;\, M'\rangle \longrightarrow \langle s[l = 0]\,;\, M\rangle$ and $\langle t\,;\, N'\rangle \longrightarrow \langle t[l' = 0]\,;\, N\rangle$, we have $\langle s[l = 0]\,;\, M\rangle \Downarrow$ *iff* $\langle t[l' = 0]\,;\, N\rangle \Downarrow$ (we have used here the fact that the evaluation is deterministic modulo renaming of locations). Thus, we have $(\mathcal{E}', s[l = 0], t[l' = 0]) \in \mathcal{X}$ as required.

(2.d) To see that $V = l$ implies $W = l'$ for some l', let $C = ![\cdot]$. Then, by the construction of $\mathcal{X}$, $\langle s\,;\, C[V]\rangle \Downarrow$ if and only if $\langle t\,;\, C[W]\rangle \Downarrow$, which implies that W must be a location.
Next, we show condition (i). Let $\mathcal{E}' = \mathcal{E} \cup \{(s(l), t(l'))\}$. Suppose $(M, N) \in \mathcal{E}'^{\star}$, i.e. there exists a context C such that $M = C[\widetilde{V}, s(l)]$ and $N = C[\widetilde{W}, t(l')]$ with $(\widetilde{V}, \widetilde{W}) \in \mathcal{E}$. Let $M' = (\lambda x.\, C[\widetilde{V}, x])(!l)$ and $N' = (\lambda x.\, C[\widetilde{W}, x])(!l')$. Since $(M', N') \in \mathcal{E}^{\star}$, we have $\langle s\,;\, M'\rangle \Downarrow$ *iff* $\langle t\,;\, N'\rangle \Downarrow$. Since $\langle s\,;\, M'\rangle \longrightarrow \langle s\,;\, M\rangle$ and $\langle s\,;\, N'\rangle \longrightarrow \langle s\,;\, N\rangle$, we have $\langle s\,;\, M\rangle \Downarrow$ *iff* $\langle t\,;\, N\rangle \Downarrow$. Hence, we have $(\mathcal{E}', s, t) \in \mathcal{X}$ as required.
Next, we show condition (ii). Suppose $(M, N) \in \mathcal{E}^{\star}$. Let $M' = (l := V_1); M$ and $N' = (l := W_1); N$. Then, (M', N') is also in $\mathcal{E}^{\star}$. By the condition $(\mathcal{E}, s, t) \in \mathcal{X}$, we have $\langle s\,;\, M'\rangle \Downarrow$ *iff* $\langle t\,;\, N'\rangle \Downarrow$, which implies $\langle s[l = V_1]\,;\, M\rangle \Downarrow$ *iff* $\langle t[l' = W_1]\,;\, N\rangle \Downarrow$. Hence, we have $(\mathcal{E}, s[l = V_1], t[l' = W_1]) \in \mathcal{X}$ as required.

(2.e) Suppose $V = \lambda x.\, P$. By the same argument as 2(a)–(d), W cannot be a constant, a tuple, or a location. So, W must be an abstraction $\lambda x.\, Q$.
Next, suppose $(V_1, W_1) \in \mathcal{E}^{\star}$. We need to show that $\langle s\,;\, P\{^{V_1}/_x\}\rangle\ \mathcal{X}_{\mathcal{E}}\ \langle t\,;\, Q\{^{W_1}/_x\}\rangle$. Since $(VV_1, WW_1) \in \mathcal{E}^{\star}$, we have $\langle s\,;\, VV_1\rangle \Downarrow$ *iff* $\langle t\,;\, WW_1\rangle \Downarrow$, which implies

$$\langle s\,;\, P\{^{V_1}/_x\}\rangle \Downarrow \text{ iff } \langle t\,;\, Q\{^{W_1}/_x\}\rangle.$$

Suppose that $\langle s\,;\, P\{^{V_1}/_x\}\rangle \Longrightarrow \langle s'\,;\, V_2\rangle$ and $\langle t\,;\, Q\{^{W_1}/_x\}\rangle \Longrightarrow \langle t'\,;\, W_2\rangle$ and that $(M, N) \in (\mathcal{E} \cup \{(V_2, W_2)\})^{\star}$. We need to show $\langle s'\,;\, M\rangle \Downarrow$ *iff* $\langle t'\,;\, N\rangle \Downarrow$, which will complete the proof. By the condition $(M, N) \in (\mathcal{E} \cup \{(V_2, W_2)\})^{\star}$, $M = C[\widetilde{V}, V_2]$ and $N = C[\widetilde{W}, W_2]$ for some context C. Let $M' = (\lambda x.\, C[\widetilde{V}, x])(VV_1)$ and $N' = (\lambda x.\, C[\widetilde{W}, x])(WW_1)$. Then, since $(M', N') \in \mathcal{E}^{\star}$, we have $\langle s\,;\, M'\rangle \Downarrow$ *iff* $\langle t\,;\, N'\rangle \Downarrow$. The result follows, since $\langle s\,;\, M'\rangle \Longrightarrow \langle s'\,;\, M\rangle$ and $\langle t\,;\, N'\rangle \Longrightarrow \langle t'\,;\, N\rangle$. □

We are now ready to prove Theorem 5.10.

PROOF. (Theorem 5.10) To show the "only if," suppose $V \equiv W$. By Lemma 5.11, it suffices to show that $(\emptyset, \langle[\widetilde{l} = \widetilde{0}]\,;\, (V, \widetilde{l})\rangle, \langle[\widetilde{l} = \widetilde{0}]\,;\, (W, \widetilde{l})\rangle)$ is an element of the second set of $\mathcal{X}$. The first condition follows immediately. Suppose $(M_1, N_1) \in \{(V, \widetilde{l}), (W, \widetilde{l})\}^{\star}$, i.e. there exists C such that $M_1 = C[(V, \widetilde{l})]$ and $N_1 = C[(W, \widetilde{l})]$. Let $C_1 = C[([\,], \widetilde{l})]$. By the assumption $V \equiv W$, we have $\langle[\widetilde{l} = \widetilde{0}]\,;\, C_1[V]\rangle \Downarrow$ if and only if $\langle[\widetilde{l} = \widetilde{0}]\,;\, C_1[W]\rangle \Downarrow$, which implies that $\langle[\widetilde{l} = \widetilde{0}]\,;\, M_1\rangle \Downarrow$ if and only if $\langle[\widetilde{l} = \widetilde{0}]\,;\, N_1\rangle \Downarrow$. Thus, we have $(\emptyset, \langle[\widetilde{l} = \widetilde{0}]\,;\, (V, \widetilde{l})\rangle, \langle[\widetilde{l} = \widetilde{0}]\,;\, (W, \widetilde{l})\rangle) \in \mathcal{X}$, as required.

To show the converse, suppose $\langle [\widetilde{l} = \widetilde{0}]\,;\,(V, \widetilde{l})\rangle \approx_{\emptyset} \langle [\widetilde{l} = \widetilde{0}]\,;\,(W, \widetilde{l})\rangle$. Given a context C and a store s such that $\mathsf{Loc}(C) \subseteq \{\widetilde{l}\} \cup \{\widetilde{l'}\} = \mathsf{dom}(s)$. Let C_1 be a location-free context:

$$(\lambda(x, \widetilde{y}).(\nu\widetilde{z})(\widetilde{y} := s(\widetilde{l}); \widetilde{z} := s(\widetilde{l'}); [\widetilde{y}/\widetilde{l}, \widetilde{z}/\widetilde{l'}]C[x]))[\,].$$

By Lemma 5.3, we have $\langle [\widetilde{l} = \widetilde{0}]\,;\,C_1[(V, \widetilde{l})]\rangle \approx_{\mathcal{E}} \langle [\widetilde{l} = \widetilde{0}]\,;\,C_1[(W, \widetilde{l}]\rangle$, which implies $\langle s\,;\,C[V]\rangle \Downarrow$ if and only if $\langle s\,;\,C[W]\rangle \Downarrow$ as required. □

5.5 Up-To Techniques

The up-to techniques developed for environmental bisimulation in pure λ-calculi are also valid for the imperative calculus, again with the expected modifications due to the enriched language. As an example we report the full definition of "up to environment, reduction and contexts" which we use in the example of Section 5.6. We reuse definitions from pure λ-calculi, such as Definition 3.14.

We write $\langle s\,;\,M\rangle\ \mathcal{X}^{\rightarrow\star}_{\mathcal{E}}\ \langle t\,;\,N\rangle$ if there are s', M', t', N' with $\langle s\,;\,M\rangle \Longrightarrow \langle s'\,;\,M'\rangle$, $\langle t\,;\,N\rangle \Longrightarrow \langle t'\,;\,N'\rangle$, and either

—$M' = C[M'', \widetilde{V}]$, $N' = C[N'', \widetilde{W}]$, and $[\cdot]_1$ is in redex position in C, and $V_i\ \mathcal{E}\ W_i$, and $\langle s'\,;\,M''\rangle\ \mathcal{X}_{\mathcal{E}}\ \langle t'\,;\,N''\rangle$;

—or else $M' = C[\widetilde{V}]$, $N' = C[\widetilde{W}]$, and $V_i\ \mathcal{E}\ W_i$ and $(\mathcal{E}, s', t') \in \mathcal{X}$.

Definition 5.12 (*up to environment, reduction, and context*). An environmental relation $\mathcal{X}$ is a *bisimulation up to environment, reduction, and context* if

(1) $(\mathcal{E}, \langle s\,;\,M\rangle, \langle t\,;\,N\rangle) \in \mathcal{X}$ implies:
  (a) if $\langle s\,;\,M\rangle \longrightarrow \langle s'\,;\,M'\rangle$ then $\langle s'\,;\,M'\rangle\ \mathcal{X}^{\rightarrow\star}_{\mathcal{E}'}\ \langle t\,;\,N\rangle$ for some $\mathcal{E}'$ with $\mathcal{E} \subseteq \mathcal{E}'$;
  (b) if $M = V$ then $\langle t\,;\,N\rangle \Longrightarrow \langle t'\,;\,W\rangle$ and $V\ \mathcal{E}'^{\widehat{\star}}\ W$ for some $\mathcal{E}'$ with $\mathcal{E} \subseteq \mathcal{E}'$ and $(\mathcal{E}', s, t') \in \mathcal{X}$;
  (c) the converse of the previous two conditions, on N;

(2) if $(\mathcal{E}, s, t) \in \mathcal{X}$ then for all $(V, W) \in \mathcal{E}$ we have:
  (a) $V = c$ implies $W = c$;
  (b) $V = (V_1, \ldots, V_n)$ implies $W = (W_1, \ldots, W_n)$ and for all i, we have $V_i\ \mathcal{E}'^{\star}\ W_i$ for $\mathcal{E} \subseteq \mathcal{E}'$ and $(\mathcal{E}', s, t) \in \mathcal{X}$;
  (c) for all fresh l, l', we have $(\mathcal{E}', s[l = 0], t[l' = 0]) \in \mathcal{X}$, for $\mathcal{E} \subseteq \mathcal{E}'$ and $(l, l') \in \mathcal{E}'$;
  (d) if $V = l$ then $W = l'$, for some l', and moreover,
    i. $(s(l), t(l')) \in \mathcal{E}'^{\widehat{\star}}$, for $(\mathcal{E}', s, t) \in \mathcal{X}$ and $\mathcal{E} \subseteq \mathcal{E}'$;
    ii. for all $(V_1, W_1) \in \mathcal{E}^{\widehat{\star}}$, we have $(\mathcal{E}', s[l = V_1], t[l' = W_1]) \in \mathcal{X}$ and $\mathcal{E} \subseteq \mathcal{E}'$;
  (e) if $V = \lambda x.\,P$ then $W = \lambda x.\,Q$ and for all $V_1, W_1 \in \mathcal{E}^{\widehat{\star}}$ it holds that $\langle s\,;\,P\{^{V_1}/_x\}\rangle\ \mathcal{X}^{\rightarrow\star}_{\mathcal{E}'}\ \langle t\,;\,Q\{^{W_1}/_x\}\rangle$, for some $\mathcal{E}'$ with $\mathcal{E} \subseteq \mathcal{E}'$;
  (f) the converse of the above five conditions, on W.

LEMMA 5.13. *If $\mathcal{X}$ is a bisimulation up to environment, reduction, and context, then $\mathcal{X} \subseteq \approx$.*

PROOF. The details of the proof can be found in Appendix C. □

5.6 Example

The example is a minor modification of Koutavas and Wand's example [Koutavas and Wand 2006b, Section 6.2]. In Koutavas and Wand's bisimulation, the built-in induction is crucial for the proof. In our method, the example is handled using up-to techniques, in a way that—in our view—makes the reasoning more direct. The other examples in Koutavas and Wand [2006b] can be handled similarly, with the same up-to techniques. Take

$$\begin{aligned} M &\stackrel{\text{def}}{=} \lambda g.\, \nu x\, (g\ \lambda z.\, (x := !x + 2)); \\ &\qquad \lambda z.\, \texttt{if}\ !x\ \texttt{mod}\ 2 = 0\ \texttt{then}\ \star\ \texttt{else}\ \Omega \\ N &\stackrel{\text{def}}{=} \lambda g.\, (g\ \lambda z.\, \star); \lambda z.\, \star. \end{aligned}$$

Intuitively, the two terms are equivalent because the location bound by x will be initialized to 0 and then incremented by 2, and therefore maintains an even number as its content. We define, as abbreviations,

$$\begin{aligned} P &\stackrel{\text{def}}{=} \lambda z.\, (l := !l + 2) \\ Q &\stackrel{\text{def}}{=} \lambda z.\, \star \\ R &\stackrel{\text{def}}{=} \lambda z.\, \texttt{if}\ !l\ \texttt{mod}\ 2 = 0\ \texttt{then}\ \star\ \texttt{else}\ \Omega. \end{aligned}$$

We set

$$\mathcal{E}_{\widetilde{l_1};\widetilde{l_2};\widetilde{l'}} \stackrel{\text{def}}{=} \{(M, N), (\widetilde{l_1}, \widetilde{l_2})\} \cup (\cup_{l \in \widetilde{l'}} \{(P, Q), (R, Q)\}),$$

where $\widetilde{l_1} \cap \widetilde{l'} = \emptyset$. Recall that $s \uplus r$ indicates the store composed by s and r, with the implicit assumption that s and r have disjoint domains. We now take $\mathcal{X}$ to be the set consisting of $(\emptyset, \langle \emptyset\,;\, M\rangle, \langle \emptyset\,;\, N\rangle)$, and all triples of the form $(\mathcal{E}_{\widetilde{l_1};\widetilde{l_2};\widetilde{l'}}, s \uplus r, t)$ where $\mathsf{dom}(s) = \widetilde{l_1}$, $\mathsf{dom}(t) = \widetilde{l_2}$, $s(\widetilde{l_1})\, \mathcal{E}^{\star}_{\widetilde{l_1};\widetilde{l_2};\widetilde{l'}}\, t(\widetilde{l_2})$, $\mathsf{dom}(r) = \widetilde{l'}$ and for all $l \in \widetilde{l'}$ we have $r(l) = 2n$, for some n. Note that $\widetilde{l'}$ can also be empty. We show that $\mathcal{X}$ is a bisimulation up to environment, reduction, and context. The clauses (2.c) and (2.d) for store extension and locations in the definition are easy. We check the conditions for the pairs (M, N), (P, Q), and (R, Q). First, (M, N). Take $(V, W) \in \mathcal{E}^{\star}_{\widetilde{l_1};\widetilde{l_2};\widetilde{l'}}$. Using the abbreviations $M_1 \stackrel{\text{def}}{=} (V\ \ P); R$ and $N_1 \stackrel{\text{def}}{=} (W\ \ Q); Q$, the terms obtained from MV and NW are $\nu x\, (M_1\{x/l\})$ and N_1. Assuming l fresh, we have

$$\langle s \uplus r\,;\, \nu x\, M_1\{x/l\}\rangle \longrightarrow \langle s \uplus r[l = 0]\,;\, M_1\rangle$$

$$\langle t\,;\, N_1\rangle \Longrightarrow \langle t\,;\, N_1\rangle.$$

Define the context $C \stackrel{\text{def}}{=} ([\cdot]_1\ \ [\cdot]_2); [\cdot]_3$. We have $M_1 = C[V, P, R]$ and $N_1 = C[W, Q, Q]$. This is sufficient, up to context, because the arguments of the context are pairwise related in $\mathcal{E}_{\widetilde{l_1};\widetilde{l_2};\widetilde{l'},l}$ and $(\mathcal{E}_{\widetilde{l_1};\widetilde{l_2};\widetilde{l'},l}, s \uplus r[l = 0], t) \in \mathcal{X}$.

For the pair (P, Q), the results of applications to related values are $l := !l + 2$ and $\star$, respectively. Further, we have

$$\langle s \uplus r\,;\, l := !l + 2\rangle \Longrightarrow \langle s \uplus r[l = r(l) + 2]\,;\, \star\rangle$$

and this is sufficient, up to reduction, since

$$(\mathcal{E}_{\widetilde{l_1};\widetilde{l_2};\widetilde{l'}}, s \uplus r[l = r(l) + 2], t) \in \mathcal{X}.$$

Finally, the pair (R, Q). The results of applications to related values are $\texttt{if}\ !l\ \texttt{mod}\ 2 = 0\ \texttt{then}\ \star\ \texttt{else}\ \Omega$ and $\star$, respectively. Further, we have

$$\langle s \uplus r\,;\ \texttt{if}\ !l\ \texttt{mod}\ 2 = 0\ \texttt{then}\ \star\ \texttt{else}\ \Omega\rangle \Longrightarrow \langle s \uplus r\,;\ \star\rangle$$

since l is a location in r and all locations in r store an even number. This is sufficient, since $(\mathcal{E}_{\widetilde{l_1};\widetilde{l_2};\widetilde{v}}, s \uplus r, t) \in \mathcal{X}$.

In Koutavas and Wand's example, the terms Q and R are not thunked. That is, one has terms $Q' \stackrel{\text{def}}{=} \star$ and $R' \stackrel{\text{def}}{=} \texttt{if}\ !l\ \texttt{mod}\ 2 = 0\ \texttt{then}\ \star\ \texttt{else}\ \Omega$. In such a case, the equality can be proved by first thunking the terms (that is, β-expanding Q' into Qc and R' into Rc, where c can be any value). The resulting terms are then proved equal using essentially the same proof above. The desired equality is finally derived by appealing to the transitivity of bisimilarity.

6. HIGHER-ORDER π-CALCULUS

6.1 Syntax and Transitions

In this section we discuss environmental bisimulations in higher-order calculi for concurrency. We consider the Higher-Order π-calculus [Sangiorgi 1996, 2001] in its simplest form, where only processes can be communicated (thus the calculus is similar to Plain CHOCS [Thomsen 1993]). Syntax and the LTS for HOπ are standard; here is the syntax.

P	::=	$\overline{a}P.\,Q$	output prefix
	$\mid$	$a(x).\,P$	input prefix
	$\mid$	x	process variable
	$\mid$	$\nu a\ P$	restriction
	$\mid$	$P \mid Q$	parallel composition
	$\mid$	$\mathbf{0}$	nil

The replication operator ($!P$) is derivable [Thomsen 1993; Sangiorgi and Walker 2001]. We use $a, b, c, \ldots$ to range over names (sometimes also called channels), and $x, y, z, \ldots$ to range over variables; the sets of names and variables are disjoint. An input $a(x).\,P$ binds the free occurrences of variable x in P; similarly a restriction $\nu a\ P$ binds the free occurrences of name a in P. We write $\mathsf{fv}(P)$ and $\mathsf{fn}(P)$ for the free variables and the free names of P, respectively; $\mathsf{bv}(P)$ and $\mathsf{bn}(P)$ for the bound variables and the bound names. We identify processes that are the same up to a renaming of bound names and bound variables. In a statement, we say that a name is *fresh* to mean that it is different from any other name occurring in objects of the statement. A process is *closed* if it does not have free variables (in contrast, it can have free names). We write $\mathcal{P}$ for the set of all processes. If r is a set of names, we sometimes write $r, \widetilde{a}$ as an abbreviation for $r \cup \{\widetilde{a}\}$, and similarly $\mathsf{fn}(P, \widetilde{a})$ for $\mathsf{fn}(P) \cup \{\widetilde{a}\}$.

Now the LTS. It is defined on open processes (it could also be given on closed processes, but in a few proofs, namely Lemmas 6.5 and 6.6 used for proving congruence for parallel composition, it will be convenient for us to define transitions also for open processes). There are three forms of transitions:

τ transitions $P \xrightarrow{\tau} P'$; input transitions $P \xrightarrow{a(x)} P'$, meaning that P can receive at a a process that will replace x in the continuation P'; and output transitions $P \xrightarrow{(\nu \widetilde{b})\overline{a}P'} P''$ meaning that P emits P' at a, and in doing so it extrudes the names $\widetilde{b}$ and evolves to P'' (names $\widetilde{b}$ are private between P' and P''). We use α to indicate a generic label of a transition. The *bound names* of α, written $\mathsf{bn}(\alpha)$, are $\emptyset$ if α is a τ or an input; they are $\widetilde{b}$ if α is an output $(\nu \widetilde{b})\overline{a}P$; the *names* of α, written $\mathsf{n}(\alpha)$, are all the names that appear in α. We write $P \xrightarrow{aM} Q$ if $P \xrightarrow{a(x)} Q'$ and $Q'\{^M/_x\} = Q$ (this form of action is called *early input* in the literature).

$$a(x).\, P \xrightarrow{a(x)} P$$

$$\overline{a}Q.\, P \xrightarrow{\overline{a}Q} P$$

$$\frac{P_1 \xrightarrow{\alpha} P_1' \qquad \mathsf{bn}(\alpha) \cap \mathsf{fn}(P_2) = \emptyset}{P_1 \mid P_2 \xrightarrow{\alpha} P_1' \mid P_2}$$

$$\frac{P_1 \xrightarrow{(\nu \widetilde{b})\overline{a}P} P_1' \qquad P_2 \xrightarrow{aP} P_2' \qquad \widetilde{b} \cap \mathsf{fn}(P_2) = \emptyset}{P_1 \mid P_2 \xrightarrow{\tau} \nu \widetilde{b}\,(P_1' \mid P_2')}$$

$$\frac{P \xrightarrow{\alpha} P' \qquad a \notin \mathsf{n}(\alpha)}{\nu a\, P \xrightarrow{\alpha} \nu a\, P'}$$

$$\frac{P \xrightarrow{(\nu \widetilde{b})\overline{a}Q} P' \qquad c \in \mathsf{fn}(Q) - \{\widetilde{b}, a\}}{\nu c\, P \xrightarrow{(\nu \widetilde{b}, c)\overline{a}Q} P'}$$

In the remainder, α may also be an early input. We sometimes write $\prod_{j \in J} P_j$ for the parallel composition of all processes P_j, for $j \in J$.

Weak transitions are defined in the usual way. Thus $\Longrightarrow$ is the reflexive and transitive closure of $\xrightarrow{\tau}$; and $\overset{\alpha}{\Longrightarrow}$ stands for $\Longrightarrow\xrightarrow{\alpha}\Longrightarrow$. Finally we define the *barbs*, and write $P \Downarrow_a$ if there is α and P' s.t. $P \xrightarrow{\alpha} P'$ where α is an input or output action at a.

LEMMA 6.1. *Suppose* $\mathsf{fn}(P) \subseteq r$ *and* $P \xrightarrow{(\nu \widetilde{b})\overline{a}P_1} P'$. *Then* $\mathsf{fn}(P_1, P') \subseteq r, \widetilde{b}$.

It will be useful, in a number of places, to have the auxiliary relation of *structural equivalence*, written $=_{\mathrm{str}}$; it is the standard structural relation for processes [Milner 1999], except that here we do not close it under all contexts, because we do not apply $=_{\mathrm{str}}$ inside prefixes. Thus we just close $=_{\mathrm{str}}$ under parallel composition and restriction.

Definition 6.2 (*structural equivalence*). *Structural equivalence*, written $=_{\mathrm{str}}$, is the smallest equivalence on on $\mathcal{P} \times \mathcal{P}$ that is closed under the following rules:

$$P \mid Q =_{\mathrm{str}} Q \mid P \qquad P \mid (Q \mid R) =_{\mathrm{str}} (P \mid Q) \mid R \qquad P \mid \mathbf{0} =_{\mathrm{str}} P$$

$$\nu a\, \nu b\, P =_{\mathrm{str}} \nu b\, \nu a\, P \qquad \nu a\, \mathbf{0} =_{\mathrm{str}} \mathbf{0}$$

$$\frac{P =_{\mathrm{str}} P'}{P \mid Q =_{\mathrm{str}} P' \mid Q} \qquad \frac{P =_{\mathrm{str}} P'}{\nu a\, P =_{\mathrm{str}} \nu a\, P'}$$

LEMMA 6.3. *Let* $M, N \in \mathcal{P}$, *and suppose* $M =_{\mathrm{str}} N$ *and* $M \xrightarrow{\alpha} M'$; *then also* $N \xrightarrow{\alpha} N'$ *and* $M' =_{\mathrm{str}} N'$.

In the remainder, we will often need to switch between open and closed processes. Indicatively, we use G, H for open processes, and P, Q, M, N, L, A, B for closed processes. We begin by presenting a few results on open processes. These results will play a central role in the proof of congruence for parallel composition in Section 6.4, to be able to relate the behaviour of two terms obtained by the same open expression for different instantiations of the free variables.

Definition 6.4. A variable x *is guarded in* $G \in \mathcal{P}$ (or simply *guarded*, when the statement makes it clear the term that is referred to) if x only occurs free in subexpressions of G of the form $\pi.B$, where π is any prefix.

A process $G \in \mathcal{P}$ is *guarded* (or has *guarded variables*) if all its free variables are guarded.

In the following lemma, we recall that an output action from an open process may contain free variables, thus $\alpha\{\widetilde{M}/\widetilde{x}\}$ is the action obtained from α applying the substitution. Further, with some abuse of notation, if $\alpha = a(x)$ then $\alpha\{Q/x\} = aQ$.

LEMMA 6.5. *Suppose that $G \in \mathcal{P}$ is guarded. Then, for all $\widetilde{H} \in \mathcal{P}$ and variables $\widetilde{x}$, we have:*

(1) If $G \xrightarrow{\alpha} G'$, with bound variables in α fresh, then $G\{\widetilde{H}/\widetilde{x}\} \xrightarrow{\alpha\{\widetilde{H}/\widetilde{x}\}} G'\{\widetilde{H}/\widetilde{x}\}$;

(2) If $G\{\widetilde{H}/\widetilde{x}\} \xrightarrow{\alpha'} M'$, with bound variables in α' fresh, then there is G' s.t. $G \xrightarrow{\alpha} G'$ and $M' = G'\{\widetilde{H}/\widetilde{x}\}$, $\alpha' = \alpha\{\widetilde{H}/\widetilde{x}\}$.

PROOF. By transition induction. □

We write $(P|)^n$ as an abbreviation for the parallel composition of n copies of P.

LEMMA 6.6. *For all $G \in \mathcal{P}$ and x there is $G' \in \mathcal{P}$ with x guarded in G', and $n \geq 0$ such that*

$$G =_{\mathrm{str}} (x\,|)^n \mid G'.$$

Furthermore, for all $H \in \mathcal{P}$ we have:

$$G\{H/x\} =_{\mathrm{str}} (H\,|)^n \mid G'\{H/x\}.$$

6.2 Environmental Bisimulation

Definition 6.7 (*environmental relation, in HOπ*). In HOπ, an *environmental relation* is a set of elements each of which is of the form $(r\,;\,\mathcal{E}\,;\,M\,;\,N)$, where M, N are closed processes, $\mathcal{E}$ is a relation on closed processes, and r is a finite set of names.

Intuitively: M and N are the tested processes; $\mathcal{E}$ is the set of processes (the "values") that the tested processes have produced earlier (by means of outputs towards the observer); r are the names that are known, and freely available, to the observer; these names may occur free in processes of $\mathcal{E}$ and in M, N. To simplify notations, we do not keep track of the other free names in $\mathcal{E}$, M, and N:

these represent private names that the tested processes have extruded earlier, and that the observer cannot directly access.

We first present the definition of environmental bisimulation and then we comment it. The definition is given in the "early" style, which makes the comparison with contextual equivalence easier. We write

$$(r\,;\,\mathcal{E})^{\star} \stackrel{\text{def}}{=} \{(G\{\widetilde{M}/\widetilde{x}\}, G\{\widetilde{N}/\widetilde{x}\}) \text{ s. t. } \mathsf{fn}(G) \subseteq r, \mathsf{fv}(G) \subseteq \widetilde{x} \text{ and } \widetilde{M}\,\mathcal{E}\,\widetilde{N}\}$$

That is, $(r\,;\,\mathcal{E})^{\star}$ is an abbreviation for the subset of $\mathcal{E}^{\star}$ (the context closure of $\mathcal{E}$) in which the free names of the contexts are in r. As usual, $\mathcal{X}, \mathcal{Y}$ range over environmental relations. We write $M\ \mathcal{X}_{\mathcal{E}\,;r}\ N$ if $(r\,;\,\mathcal{E}\,;\,M\,;\,N) \in \mathcal{X}$.

In a statement of the reminder of the section, a name is *fresh* if it does not appear among the free names of the processes and other objects in the statement. For instance, in clauses (3) and (5) below, "fresh" means that the name(s) do not appear among the free names of $\mathcal{E}$, M, N or in r.

Definition 6.8. An environmental relation $\mathcal{X}$ is an *environmental bisimulation* if $M\ \mathcal{X}_{\mathcal{E}\,;r}\ N$ implies:

(1) if $M \stackrel{\tau}{\rightarrow} M'$ then $N \Longrightarrow N'$ and $M'\ \mathcal{X}_{\mathcal{E}\,;r}\ N'$;
(2) if $M \stackrel{aP}{\rightarrow} M'$ with $a \in r$, and $(P, Q) \in (r\,;\,\mathcal{E})^{\star}$, then $N \stackrel{aQ}{\Longrightarrow} N'$ and $M'\ \mathcal{X}_{\mathcal{E}\,;r}\ N'$;
(3) if $M \stackrel{(\nu\widetilde{b})\overline{a}P}{\longrightarrow} M'$ with $a \in r$ and $\widetilde{b}$ fresh, then $N \stackrel{(\nu\widetilde{c})\overline{a}Q}{\Longrightarrow} N'$ with $\widetilde{c}$ fresh and $M'\ \mathcal{X}_{\mathcal{E}\cup\{(P,Q)\}\,;r}\ N'$;
(4) if $(P, Q) \in \mathcal{E}$ then $P \mid M\ \mathcal{X}_{\mathcal{E}\,;r}\ Q \mid N$;
(5) for all r' fresh we have $;M\ \mathcal{X}_{\mathcal{E}\,;r,r'}\ N$;
(6) the converse of (1-3), on the actions from N.

We write $\approx$ for the union of all environmental bisimulations, thus $M\ \approx_{\mathcal{E}\,;r}\ N$ holds if $(r\,;\,\mathcal{E}\,;\,M\,;\,N) \in \mathcal{X}$, for some bisimulation $\mathcal{X}$. We sometimes write $M \simeq N$ if $M\ \approx_{\emptyset\,;\mathsf{fn}(M,N)}\ N$.

Definition 6.8 has five clauses (plus the converse of some of them). The first is the usual one for τ-actions. The second is analogous to the clause for abstractions in λ-calculi of earlier sections. (e.g., clause (2.e) in Definition 5.1). The inputs P and Q for the tested processes are constructed using the environment $\mathcal{E}$ and the names r available to the observer. The third clause is for outputs: the environment is updated with the processes emitted. Both in the input and in the output clause, the action from the processes is at a name that is known to the observer (condition $a \in r$). The fourth clause is new—it does not appear in the sequential calculi of the previous sections. It intuitively shows that the observer can run the values in the environment at any time. The fifth clause allows creation of fresh names by the observer. Relations $\approx_{\mathcal{E}\,;r}$ are extended to *open terms* using closing input abstractions. For instance if $\{x\} = \mathsf{fv}(M, N)$ then $M\ \approx_{\mathcal{E}\,;r}\ N$ if $a(x).M\ \approx_{\mathcal{E}\,;r,a}\ a(x).N$, for a fresh.

In the remainder, unless otherwise stated, *all processes are closed*. We state a few simple lemmas, partly mimicking analogous results for the sequential languages in the previous sections.

6.3 Basic Properties

We study in this section some basic properties of environmental bisimilarity, such as transitivity and weakening (on environments and names), and a simple up-to technique that allows us manipulation of environments and restricted names, and structural rearrangements of terms.

LEMMA 6.9 (TRANSITIVITY). *$P \approx_{\mathcal{E}_1;r} Q \approx_{\mathcal{E}_2;r} Q'$ implies $P \approx_{\mathcal{E}_1\mathcal{E}_2;r} Q'$.*

LEMMA 6.10 (WEAKENING ON ENVIRONMENTS). *$P \approx_{\mathcal{E};r} Q$ and $\mathcal{E}' \subseteq \mathcal{E}$ imply $P \approx_{\mathcal{E}';r} Q$.*

LEMMA 6.11 (WEAKENING ON NAMES). *$P \approx_{\mathcal{E};r} Q$ and $r' \subseteq r$ imply $P \approx_{\mathcal{E};r'} Q$.*

LEMMA 6.12. *$P \approx_{\mathcal{E};r} Q$ and name a is fresh imply $P \approx_{\mathcal{E};r,a} Q$.*

LEMMA 6.13. *$Q \simeq P$ implies, for all r, $Q \approx_{\emptyset;r} P$.*

PROOF. A consequence of Lemmas 6.11 and 6.12. □

LEMMA 6.14. *$P =_{\mathrm{str}} Q \approx_{\mathcal{E};r} Q'$ implies $P \approx_{\mathcal{E};r} Q'$.*

COROLLARY 6.15. *$\simeq$ is an equivalence relation.*

With reasoning similar to that for previous languages we prove the soundness of a few basic up-to techniques: up to environment, up to restriction, up to $=_{\mathrm{str}}$. In the up-to environment, we allow, in the conclusion of the clauses, environments that are larger than requested in the definition of bisimulation (i.e., $\mathcal{X}_{\mathcal{E};r}$ is replaced by $\mathcal{X}_{\mathcal{E}';r}$, for $\mathcal{E} \subseteq \mathcal{E}'$). In the up-to restriction, we allow a larger set of free names (i.e., $\mathcal{X}_{\mathcal{E};r}$ is replaced by $\mathcal{X}_{\mathcal{E};r,r'}$) and moreover outermost identical restrictions in the derivative processes can be erased. In the up-to $=_{\mathrm{str}}$, we allow to rewrite the derivative processes, using $=_{\mathrm{str}}$; further, restrictions at the top level of such processes can be erased. We only report the definition of the combination of these three techniques (which subsumes the three techniques themselves).

We write $M \; \mathcal{X}^{\nu,=_{\mathrm{str}},\mathrm{env}}_{\mathcal{E};r} \; N$ if there are $M', N', \tilde{x}, \tilde{y}, \mathcal{E}', r'$ with $M =_{\mathrm{str}} \nu\tilde{x}\, M'$, $N =_{\mathrm{str}} \nu\tilde{y}\, N'$, $\mathcal{E} \subseteq \mathcal{E}'$, $r \subseteq r'$; and $M' \; \mathcal{X}_{\mathcal{E}';r'} \; N'$. Note that names in r' may include names in $\tilde{x}$ and $\tilde{y}$, which means that the observer will then have access to names that were originally restricted.

Definition 6.16. An environment relation $\mathcal{X}$ is a *bisimulation up to restriction, environment and* $=_{\mathrm{str}}$ if $M \; \mathcal{X}_{\mathcal{E};r} \; N$ implies:

(1) if $M \xrightarrow{\tau} M'$ then $N \Longrightarrow N'$ and $M' \; \mathcal{X}^{\nu,=_{\mathrm{str}},\mathrm{env}}_{\mathcal{E};r} \; N'$;

(2) if $M \xrightarrow{aP} M'$ with $a \in r$, and $(P, Q) \in (r;\mathcal{E})^\star$, then $N \stackrel{aQ}{\Longrightarrow} N'$ and $M' \; \mathcal{X}^{\nu,=_{\mathrm{str}},\mathrm{env}}_{\mathcal{E};r} \; N'$;

(3) if $M \xrightarrow{(\nu\tilde{b})\bar{a}P} M'$, with $a \in r$ and $\tilde{b}$ fresh, then $N \stackrel{(\nu\tilde{c})\bar{a}Q}{\Longrightarrow} N'$, for $\tilde{c}$ fresh, and $M' \; \mathcal{X}^{\nu,=_{\mathrm{str}},\mathrm{env}}_{\mathcal{E},(P,Q);r} \; N'$;

(4) for all $(P, Q) \in \mathcal{E}$ we have $M \mid P \; \mathcal{X}^{\nu,=_{\mathrm{str}},\mathrm{env}}_{\mathcal{E};r} \; Q \mid N$;

(5) for all r' fresh (i.e., not in $\mathsf{fn}(\mathcal{E}, M, N)$) we have $M\ \mathcal{X}_{\mathcal{E};r,r'}\ N$;
(6) the converse of (1-3), on N.

LEMMA 6.17. *If $\mathcal{X}$ is an environmental bisimulation up to restriction, environment and $=_{\mathrm{str}}$ then $\mathcal{X} \subseteq \approx$.*

PROOF. Suppose $\mathcal{X}$ is such a relation. Define $\mathcal{Y}$ as the set of all elements $(r\,;\mathcal{E}\,;P\,;Q)$ such that there are $\widetilde{x}, \widetilde{y}, P', Q', \mathcal{E}', r'$ with

— $P \equiv \nu\widetilde{x}\ P'$ and $Q \equiv \nu\widetilde{y}\ Q'$, for $(\widetilde{x} \cup \widetilde{y}) \cap r = \emptyset$;
— $\mathcal{E} \subseteq \mathcal{E}'$ and $r \subseteq r'$;
— $P'\ \mathcal{X}_{\mathcal{E}';r'}\ Q'$.

Then $\mathcal{Y}$ is an environmental bisimulation. We omit the details, which are similar to (in fact much simpler than) those for the soundness of the technique in Definition 6.25. □

The technique preceding is essentially a special case of the *bisimulation up to environment, restriction, and* $\simeq$ that is discussed later (Definition 6.25; it is not quite a special case of it, but in this respect the differences are minor). Still, the technique above is useful because it is simpler to use; moreover, we need the simple technique to establish results, such as Theorem 6.18 and Corollary 6.19, that will then allow us to prove the more complex technique.

6.4 Compositionality

To establish congruence, we first prove that environmental bisimilarity is preserved by parallel composition (this is conceptually similar, but technically more complex, to the results of congruence with respect to evaluation contexts in the λ-calculi of previous sections). We then use this result, together with combinations of up-to techniques, to derive congruence results for arbitrary contexts.

6.4.1 *Congruence for Parallel Composition.*

THEOREM 6.18. *$P\ \approx_{\mathcal{E};r}\ Q$ implies $P \mid M\ \approx_{\mathcal{E};r}\ Q \mid N$ for $(M, N) \in (r\,;\mathcal{E})^\star$.*

PROOF. Suppose $\mathcal{Y}$ is a bisimulation. Define $\mathcal{X}$ as the set of all tuples

$$(r\,;(r\,;\mathcal{E})^\star\,;\ P \mid G\{\widetilde{M}/\widetilde{x}\}\,;\ Q \mid G\{\widetilde{N}/\widetilde{x}\})$$

such that G is guarded, $(r\,;\mathcal{E}\,;P\,;Q) \in \mathcal{Y}$, and $(G\{\widetilde{M}/\widetilde{x}\}, G\{\widetilde{N}/\widetilde{x}\}) \in (r\,;\mathcal{E})^\star$ with $(\widetilde{M}, \widetilde{N}) \in \mathcal{E}$ (and the appropriate side conditions on the names of G).

We prove that this is a bisimulation up-to $=_{\mathrm{str}}$, restriction, and environment, using a case analysis on the possible forms of action, in Section D.

Although G above is guarded, this is sufficient to prove the theorem, in virtue of Lemma 6.6 and clause (4) of bisimulation (see the reasoning that is made in Section D after each transition to rewrite the derivatives so to bring out a guarded context). □

Next, we establish some further properties of environmental bisimilarity, which will be needed to obtain the remaining congruence results.

6.4.2 *Further Properties of Environmental Bisimilarity.* In this section, we establish properties of environmental bisimilarity that mainly concern manipulation of environments, possibly in connection with transitivity issues.

The lemma below is the counterpart of Lemma 3.17.

COROLLARY 6.19. *$P \approx_{\mathcal{E};r} Q$ and $\mathcal{E}' \subseteq (r\,;\mathcal{E})^\star$ imply $P \approx_{\mathcal{E}';r} Q$.*

PROOF. By exhibiting the appropriate bisimulation. It is essentially a consequence of Theorem 6.18, since the hard part in the proof is clause (4) of bisimulation and this is handled using the theorem. □

Corollary 6.20 is a stronger form of transitivity, counterpart in HOπ of Lemma 3.19. It is derived from Lemma 6.9 and Corollary 6.19.

COROLLARY 6.20. *If $P \approx_{\mathcal{E}_1;r} Q$ and $Q \approx_{\mathcal{E}_2;r} L$, then $P \approx_{(r\,;\mathcal{E}_1)^\star\,(r\,;\mathcal{E}_2)^\star;r} L$.*

We report a few more results about forms of composition for environmental bisimilarity; these results have some interest on their own and will be useful in the following sections. The following two corollaries are derived from Corollary 6.20 and some simple manipulations of environments and names.

COROLLARY 6.21. *If $P \approx_{\mathcal{E}_1;r_1} Q$ and $Q \approx_{\mathcal{E}_2;r_2} L$, with $\mathsf{fn}(P, Q, \mathcal{E}_1) \subseteq r_1$ and $\mathsf{fn}(Q, L, \mathcal{E}_2) \subseteq r_2$, then $P \approx_{\mathcal{E}_1 \cup \mathcal{E}_2;r_1 \cup r_2} L$.*

PROOF. First, we obtain $P \approx_{\mathcal{E}_1;r_1,r_2} Q$ and $Q \approx_{\mathcal{E}_2;r_1,r_2} L$ by Lemma 6.12; then we apply Corollary 6.20 and derive $P \approx_{(r_1,r_2\,;\mathcal{E}_1)^\star\,(r_1,r_2\,;\mathcal{E}_2)^\star;L}$; finally we conclude with Lemma 6.10. □

COROLLARY 6.22. *$P \approx_{\mathcal{E};r} Q \simeq Q'$ implies $P \approx_{\mathcal{E};r} Q'$.*

PROOF. Let $s = \mathsf{fn}(\mathcal{E}, Q, Q') \cup r$. Then we have $Q \approx_{\emptyset;s} Q'$ (Lemma 6.13). By Corollary 6.19, $Q \approx_{\mathrm{Id}_s;s} Q'$, where Id_s is the identity relation on the processes with free names in s. By Lemma 6.11 also $Q \approx_{\mathrm{Id}_s;r} Q'$. Finally, by transitivity (Lemma 6.9), from $P \approx_{\mathcal{E};r} Q$ and $Q \approx_{\mathrm{Id}_s;r} Q'$ we derive $P \approx_{\mathcal{E};r} Q'$. □

We write $\approx_{\mathcal{E},(M,N);r}$ as an abbreviation for $\approx_{\mathcal{E}\cup\{(M,N)\};r}$, and $\approx_{(N,L);r}$ in place of $\approx_{\{(N,L)\};r}$.

The following theorem has also to do with transitivity. It allows us to modify, in an assertion $P \approx_{\mathcal{E},(M,N);r} Q$, the process Q and the associated environment entry N. We will use the theorem several times, mostly in applications of up-to techniques.

THEOREM 6.23. *$P \approx_{\mathcal{E},(M,N);r} Q$ and $Q \approx_{(N,L);s} Q'$ with $\mathsf{fn}(\mathcal{E}, r) \subseteq s$ imply $P \approx_{\mathcal{E},(M,L);r} Q'$.*

PROOF. $Q \approx_{(N,L);s} Q'$ implies (Corollary 6.19) $Q \approx_{(s\,;(N,L))^\star;s} Q'$. This implies (Lemma 6.11) $Q \approx_{(s\,;(N,L))^\star;r} Q'$. Hence, by transitivity (Lemma 6.9) and weakening on environments (Lemma 6.10) we deduce $P \approx_{\mathcal{E},(M,L);r} Q'$. □

We will also use the following lemma; it is the HOπ version of a standard result in process calculi, stating that if a process P is bisimilar with a derivative of a process Q, and conversely so, then the two initial processes P and Q are themselves bisimilar.

LEMMA 6.24. *Suppose* $P \Longrightarrow P'$ *and* $Q \Longrightarrow Q'$, *with* $P \approx_{\mathcal{E};r} Q'$ *and* $P' \approx_{\mathcal{E};r} Q$. *Then also* $P \approx_{\mathcal{E};r} Q$.

6.4.3 *Another Up-To Technique.* We can now introduce a further up-to technique—a stronger version of the technique in Definition 6.16 (note that we have used the technique in Definition 6.16 to obtain results in Section 6.4.1 that are used, in turn, to prove the soundness of the new technique).

We write $M =_{\mathrm{str}} \mathcal{X}_{\mathcal{E};r} \simeq N$ if there are M', N' with

— $M =_{\mathrm{str}} M'$ and $N \simeq N'$
— $M' \ \mathcal{X}_{\mathcal{E};r} \ N'$.

Definition 6.25. An environment relation $\mathcal{X}$ is a *bisimulation up to environment, restriction, and* $\simeq$ if $M \, \mathcal{X}_{\mathcal{E};r} \, N$ implies:

(1) if $M \xrightarrow{\tau} M'$ then $N \Longrightarrow N'$ and $M' =_{\mathrm{str}} \mathcal{X}_{\mathcal{E}';r} \simeq N'$, with $\mathcal{E} \subseteq \mathcal{E}'$;
(2) if $M \xrightarrow{aP} M'$ with $a \in r$, and $(P, Q) \in (r ; \mathcal{E})^\star$, then $N \stackrel{aQ}{\Longrightarrow} N'$ and $M' =_{\mathrm{str}} \mathcal{X}_{\mathcal{E}';r} \simeq N'$, with $\mathcal{E} \subseteq \mathcal{E}'$;
(3) if $M \xrightarrow{(\nu \widetilde{b})\overline{a}P} M'$ with $a \in r$ and $\widetilde{b}$ fresh, then $N \stackrel{(\nu \widetilde{c})\overline{a}Q}{\Longrightarrow} N'$, with $\widetilde{c}$ fresh, and $M' =_{\mathrm{str}} \mathcal{X}_{\mathcal{E}',(P,Q);r,r'} \simeq N'$, with $\mathcal{E} \subseteq \mathcal{E}'$;
(4) for all $(P, Q) \in \mathcal{E}$ we have $M \mid P \simeq \mathcal{X}_{\mathcal{E}';r} \simeq N \mid Q$, with $\mathcal{E} \subseteq \mathcal{E}'$;
(5) for all r' fresh (ie, not in $\mathsf{fn}(\mathcal{E}, M, N)$) we have $M \, \mathcal{X}_{\mathcal{E};r,r'} \, N$;
(6) the converse of (1-3), on the actions from N.

Note that the up-to restriction (use of r') is only used in the output clause; elsewhere it would be less useful and would complicate the soundness proof.

The soundness of the technique is established in Lemma 6.28; for its proof, we use the two following auxiliary results.

LEMMA 6.26. *Suppose* $\mathcal{X}$ *is an environmental bisimulation up to* $\simeq$, *restriction, environment, and* $P \ \mathcal{X}_{\mathcal{E};r} \ Q$, *and* $P \Longrightarrow P'$. *Then there is* Q' *and* $\mathcal{E}'$ *with* $\mathcal{E} \subseteq \mathcal{E}'$ *such that* $Q \Longrightarrow Q'$ *and* $P' =_{\mathrm{str}} \mathcal{X}_{\mathcal{E}';r} \simeq Q'$.

PROOF. Induction on the length of $P \Longrightarrow P'$. For the basic case, use the definition of bisimulation up-to. For the inductive case, use induction, transitivity of $=_{\mathrm{str}}$ and $\simeq$ (and some simple facts about weak transitions emanating from $\simeq$-related processes). □

LEMMA 6.27. *Suppose* $\mathcal{X}$ *is an environmental bisimulation up to* $\simeq$, *restriction, environment, and* $P \ \mathcal{X}_{\mathcal{E};r} \ Q$, *and* $P \stackrel{(\nu \widetilde{b})\overline{a}M}{\Longrightarrow} A$. *Then for all* t *there are* $B, N', N, s, \widetilde{c}$ *and* $\mathcal{F}$ *with* $\mathcal{E} \subseteq \mathcal{F}$ *and* $\widetilde{c} \cap t = \emptyset$, *such that* $Q \stackrel{(\nu \widetilde{c})\overline{a}N}{\Longrightarrow} B$ *and* $A =_{\mathrm{str}} \mathcal{X}_{\mathcal{F},(M,N');r,s} \approx_{(N',N);r,t} B$.

PROOF. If $P \stackrel{(\nu \widetilde{b})\overline{a}M}{\Longrightarrow} A$, then there are A_1, A_2 such that $P \Longrightarrow A_1 \xrightarrow{(\nu \widetilde{b})\overline{a}M} A_2 \Longrightarrow A$. First, we apply Lemma 6.26 and deduce that there are $P_1, Q_1, B_1, \mathcal{E}'$ such that $Q \Longrightarrow B_1$ and

$$A_1 =_{\mathrm{str}} P_1 \ \mathcal{X}_{\mathcal{E}';r} \ Q_1 \simeq B_1$$

with $\mathcal{E} \subseteq \mathcal{E}'$. Further, also

$$Q_1 \approx_{\emptyset;r,t} B_1$$

(Lemma 6.13).

Now, starting from $A_1 \xrightarrow{(\nu\widetilde{b})\overline{a}M} A_2$ we use Lemma 6.3, definition of bisimulation up-to, and simple properties of $\simeq$ (bisimulation properties on weak transitions) to deduce that there are P_1', P_2, s, N_1, N, Q_2, $Q_1'\widetilde{c}$, B_2, $\mathcal{E}''$ s.t. $B_1 \stackrel{(\nu\widetilde{c})\overline{a}N}{\Longrightarrow} B_2$, $\mathcal{E}' \subseteq \mathcal{E}''$ and $Q_1 \stackrel{(\nu\widetilde{c'})\overline{a}N_1}{\Longrightarrow} Q_1'$, with $\widetilde{c}, \widetilde{c'}$ fresh, with

$$A_2 =_{\mathrm{str}} P_1' =_{\mathrm{str}} P_2 \ \mathcal{X}_{\mathcal{E}'',(M,N_1);r,s} \ Q_2 \simeq Q_1' \approx_{(N_1,N);r,t} B_2.$$

Finally, proceeding similarly on the reduction $A_2 \Longrightarrow A$ we deduce that there are $B, \mathcal{E}'''$ and other processes s.t. $B_2 \Longrightarrow B$ and

$$A_2 =_{\mathrm{str}} P_1'' =_{\mathrm{str}} P_2' =_{\mathrm{str}} \mathcal{X}_{\mathcal{E}''',(M,N_1);r,s} \simeq Q_2' \simeq Q_1'' \approx_{(N_1,N);r,t} B$$

with $\mathcal{E}'' \subseteq \mathcal{E}'''$. By composing relations and environments, we get the final result. We get

$$A_2 =_{\mathrm{str}} \mathcal{X}_{\mathcal{E}''',(M,N_1);r,s} \approx_{(N_1,N);r,t} B$$

using Lemma 6.13, Corollary 6.20, and the weakening lemmas 6.10 and 6.11. □

LEMMA 6.28. *Suppose $\mathcal{Y}$ is an environmental bisimulation up to restriction, environment, and $\simeq$. Then $\mathcal{Y} \subseteq \approx$.*

PROOF. Suppose $\mathcal{Y}$ is such a relation. Define $\mathcal{X}$ as the set of all elements of the form $(r\,;\,\mathcal{E}\,;\,P\,;\,Q)$ such that there are $r_1, \mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, P_1, Q_1$ with

— $P \simeq_{\mathcal{E}_1;r} P_1$,
— $P_1 \,\mathcal{Y}_{\mathcal{E}_2;r,r_1}\, Q_1$,
— $Q_1 \simeq_{\mathcal{E}_3;r} Q$,
— $\mathcal{E} \subseteq \mathcal{E}_1\mathcal{E}_2\mathcal{E}_3$.

We prove that this is a bisimulation up to environment. r

The interesting case is that of output actions. Thus suppose $P \xrightarrow{(\nu\widetilde{b})\overline{a}M} P'$. We have: $P_1 \stackrel{(\nu\widetilde{b_1})\overline{a}M_1}{\Longrightarrow} P_1'$ with

$$P' \simeq_{\mathcal{E}_1,(M,M_1);r} P_1'.$$

We also have (Lemma 6.27, for $t = \mathsf{fn}(\mathcal{E}_3)$) $Q_1 \stackrel{(\nu\widetilde{c_1})\overline{a}N_1}{\Longrightarrow} Q_1'$ with, for some P_2, Q_2, N_1N_2, s,

$$P_1' =_{\mathrm{str}} P_2 \ \mathcal{Y}_{\mathcal{E}_2,(M_1,N_2);r,r_1,s} \ Q_2 \approx_{(N_2,N_1);r,t} Q_1'.$$

Finally we have $Q \stackrel{(\nu\widetilde{c})\overline{a}N}{\Longrightarrow} Q'$ with

$$Q_1' \simeq_{\mathcal{E}_3,(N_1,N);r} Q'.$$

Using Lemma 6.14 we deduce

$$P' \approx_{\mathcal{E}_1,(M,M_1);r} P_2$$

and similarly, using Theorem 6.23

$$Q_2 \approx_{\mathcal{E}_3,(N_2,N);r} Q'.$$

Setting $\mathcal{E}_1' \stackrel{\text{def}}{=} \mathcal{E}_1, (M, M_1)$, $\mathcal{E}_2' \stackrel{\text{def}}{=} \mathcal{E}_2, (M_1, N_2)$, $\mathcal{E}_3' \stackrel{\text{def}}{=} \mathcal{E}_3, (N_2, N)$, we have $\mathcal{E}, (M, N) \subseteq \mathcal{E}_1'\mathcal{E}_2'\mathcal{E}_3'$. Thus we can conclude

$$P' \; \mathcal{X}_{\mathcal{E}_1'\mathcal{E}_2'\mathcal{E}_3';r} \; Q'$$

and we are done. □

6.4.4 *Further Compositionality Properties.* So far we have shown only a congruence result of environmental bisimilarity with respect to parallel composition. We are now in the position to prove more general results.

LEMMA 6.29. *If* $P \approx_{\mathcal{E};r} Q$ *and* $b \notin \mathsf{fn}(\mathcal{E})$, *then* $\nu b\, P \approx_{\mathcal{E};r-b} \nu b\, Q$.

The first result is a substitutivity property for values (i.e., processes related in the environment). We must however ensure that the environment entries are indistinguishable when played in a bisimulation game; for this, proving it on the empty game (process $\mathbf{0}$ on both sides) is sufficient.

THEOREM 6.30. *If* $\mathbf{0} \approx_{\mathcal{E};r} \mathbf{0}$, *with* $(P, Q) \in \mathcal{E}$, *and* $\mathsf{fn}(G) \subseteq r$, *then* $G\{^P/_x\} \approx_{\mathcal{E};r} G\{^Q/_x\}$.

PROOF. From $\mathbf{0} \approx_{\mathcal{E};r} \mathbf{0}$ and Corollary 6.19 we deduce $\mathbf{0} \approx_{(r;\mathcal{E})^\star;r} \mathbf{0}$. Using clause (4) of bisimulation, and picking a name $a \in r$ (note that if r is $\emptyset$ then the result of the theorem is trivial), as $(a(x).G, a(x).G) \in (r\,;\,\mathcal{E})^\star$, we infer $a(x).G \approx_{(r;\mathcal{E})^\star;r} a(x).G$ (note that $\mathsf{fn}(a(x).G) \subseteq r$).

By clause (2) of bisimulation, there is B with $G\{^Q/_x\} \Longrightarrow B$, and A with $G\{^P/_x\} \Longrightarrow A$ s.t. $G\{^P/_x\} \approx_{(r;\mathcal{E})^\star;r} B$ and $A \approx_{(r;\mathcal{E})^\star;r} G\{^Q/_x\}$. Hence, using Lemma 6.24, $G\{^P/_x\} \approx_{(r;\mathcal{E})^\star;r} G\{^Q/_x\}$.

Finally by weakening the environment (Lemma 6.10), $G\{^P/_x\} \approx_{\mathcal{E};r} G\{^Q/_x\}$. □

COROLLARY 6.31. $\mathbf{0} \approx_{\mathcal{E};r} \mathbf{0}$, *with* $\mathsf{fn}(\mathcal{E}) \subseteq r$ *implies* $\mathcal{E} \subseteq \simeq$.

PROOF. Use Theorem 6.30 plus weakening. □

The next result, Theorem 6.32, is a substitutivity property for bisimilar processes, under the empty environment (which is the interesting case). It immediately gives us the congruence of $\simeq$. Theorem 6.32 looks somehow like the dual of Theorem 6.30: in the latter case, we substitute values of the environment, under the hypothesis of an "empty game"; in the former case we substitute processes of the game, with the hypothesis of an empty environment.

THEOREM 6.32. *If* $P \approx_{\emptyset;r} Q$, *and* $\mathsf{fn}(G, P, Q) \subseteq r$, *and* $\mathsf{fv}(G) \subseteq \{x\}$ *then also* $G\{^P/_x\} \approx_{\emptyset;r} G\{^Q/_x\}$.

PROOF. Appealing to Lemma 6.6 and Theorem 6.18, it is sufficient to prove the theorem for G guarded. In the proof, we write $\mathcal{E}_s$ as an abbreviation for $(s\,;\,(P, Q))^\star$. Let $\mathcal{X}$ be the set of all elements of the form

$$(r\,;\,\emptyset\,;\,P\,;\,Q)$$

or

$$(r\,;\,\mathcal{E}_r\,;\,G\{^P/_x\}\,;\,G\{^Q/_x\})$$

(where, we recall $\mathcal{E}_r = (r\,;\,(P,Q))^\star$), with $P \approx_{\emptyset;r} Q$, $\mathsf{fn}(G,P,Q) \subseteq r$ and G guarded.

We show that $\mathcal{X}$ is a bisimulation up to environment, restriction, and $\simeq$. We consider the clauses of bisimulation for elements $(r\,;\,\mathcal{E}_r\,;\,G\{^P/_x\}\,;\,G\{^Q/_x\})$. The case for the elements of the other kind, and clauses (4) and (5) of bisimulation, are simpler.

By Lemma 6.5, any action from $G\{^P/_x\}$ originates from an action from G.

—Output action $G \xrightarrow{(\nu\widetilde{b})\overline{a}G_1} G_2$.

We have $G\{^P/_x\} \xrightarrow{(\nu\widetilde{b})\overline{a}G_1\{^P/_x\}} G_2\{^P/_x\}$ and $G\{^Q/_x\} \xrightarrow{(\nu\widetilde{b})\overline{a}G_1\{^Q/_x\}} G_2\{^Q/_x\}$. We also have $\widetilde{b} \cap \mathsf{fn}(P,Q) = \emptyset$ and $(G_1\{^P/_x\}, G_1\{^Q/_x\}) \in \mathcal{E}_{r,\widetilde{b}}$. Further, from Lemma 6.6, $G_2 =_{\mathrm{str}} (x\ |)^n \mid G_2'$ for some $n \geq 0$ and G_2' guarded. Hence $G\{^P/_x\} =_{\mathrm{str}} (P|)^n \mid G_2'\{^P/_x\}$ and $G\{^Q/_x\} =_{\mathrm{str}} (Q\ |)^n \mid G_2'\{^Q/_x\}$. We also have, using clause (5) of bisimulation on $P \approx_{\emptyset;r} Q$,

$$P \approx_{\emptyset;r,\widetilde{b}} Q$$

We can therefore apply Theorem 6.18 (n times, plus we use the property $=_{\mathrm{str}} \subseteq \simeq$ and transitivity of $\simeq$) to deduce

$$(P\ |)^n \mid G_2'\{^Q/_x\} \simeq (Q\ |)^n \mid G_2'\{^Q/_x\}$$

(note that $\mathsf{fn}(P, Q, G_2'\{^Q/_x\}) \subseteq r, \widetilde{b}$). We now have

$$(P\ |)^n \mid G_2'\{^P/_x\}\ \mathcal{X}_{\mathcal{E}_{r,\widetilde{b}};r,\widetilde{b}}\ (P\ |)^n \mid G_2'\{^Q/_x\}$$

(note that $(P\ |)^n \mid G_2'$ is guarded). This closes the bisimulation, up to $\simeq$, restriction, and environment. (Note that $\mathcal{E}_r, (G_1\{^P/_x\}, G_1\{^Q/_x\}) \subseteq \mathcal{E}_{r,\widetilde{b}}$, and that up-to restriction is used on the names $\widetilde{b}$.)

—Input action $G \xrightarrow{a(y)} G_1$.

We have $G\{^P/_x\} \xrightarrow{a(y)} G_1\{^P/_x\}$ and $G\{^Q/_x\} \xrightarrow{a(y)} G_1\{^Q/_x\}$.

Take $(H\{^P/_z\}, H\{^Q/_z\}) \in \mathcal{E}_r$. If we set $G' \stackrel{\mathrm{def}}{=} G_1\{^{H\{x/z\}}/_y\}$, then we have

$$G_1\{^P/_x\}\{^{H\{P/z\}}/_y\} = G'\{^P/_x\} \text{ and } G_1\{^Q/_x\}\{^{H\{Q/z\}}/_y\} = G'\{^Q/_x\}.$$

We can now proceed as in the previous case, by separating G' into a guarded component and a composition of x. We can then rearrange $G_1\{^P/_x\}$ and $G_1\{^Q/_x\}$ by means of $\simeq$ and $=_{\mathrm{str}}$ so to close the bisimulation.

—Internal action $G \xrightarrow{\tau} G_1$. This is similar to the previous cases (in fact simpler). □

COROLLARY 6.33. *Relation $\simeq$ is a congruence relation.*

PROOF. It follows from Theorem 6.32, Lemma 6.29, and Lemma 6.13. □

Thus, if $P \simeq Q$, then for any G with $\mathsf{fv}(G) \subseteq \{x\}$ we have $G\{^P/_x\} \simeq G\{^Q/_x\}$.

6.5 Contextual Equivalence

The congruence result can be used to establish the correspondence between environmental bisimilarity and contextual equivalence. In concurrency, due to the presence of non-determinism, confluence usually fails, and the branching structure in the reduction tree of a term becomes important. The definition of contextual equivalence has to be refined, adding a bisimulation clause on reductions. The resulting relation is called *barbed congruence*. We consider here the reduction-closed version of barbed congruence [Honda and Yoshida 1995; Sangiorgi and Walker 2001].

Definition 6.34 (*barbed congruence*). *Reduction-closed barbed congruence*, $\equiv_{\mathrm{b}}$, is the largest relation that is symmetric, reduction-closed (i.e., if $M \equiv_{\mathrm{b}} N$, for M, N closed, and $M \stackrel{\tau}{\rightarrow} M'$, then there is N' s.t. $N \Longrightarrow N'$ and $M' \equiv_{\mathrm{b}} N'$), context-closed (i.e., $M \equiv_{\mathrm{b}} N$ implies $C[M] \equiv_{\mathrm{b}} C[N]$, for all contexts), and barb preserving (i.e., if $M \equiv_{\mathrm{b}} N$, for M, N closed, and $M \Downarrow_a$, then also $N \Downarrow_a$).

Although less robust than ordinary barbed congruence, reduction-closed barbed congruence allows us a simpler proof of the correspondence with a labelled bisimilarity.

THEOREM 6.35. *Relations* $\equiv_{\mathrm{b}}$ *and* $\simeq$ *coincide.*

PROOF. See Section E. □

6.6 Up To Context

We consider here a sophisticated up-to technique, involving an up to context of the kind discussed in λ-calculi of earlier sections: *environmental bisimulation up to context and* $=_{\mathrm{str}}$.

A context C, possibly containing several holes, is an *evaluation context* (or also a *redex context*) if the first hole (by convention the hole $[\cdot]_1$) appears exactly once and at the top level (i.e., not underneath a prefix); that is, $C =_{\mathrm{str}} \nu\tilde{x}([\cdot]_1 \mid C')$ where C' does not contain hole $[\cdot]_1$. And, if C is an evaluation context, then $\mathsf{bn}_1(C)$ are the bound names of C whose scope include the hole $[\cdot]_1$. We write $P \ \mathcal{X}^{\mathrm{c}}_{\mathcal{E};r} \ Q$ if there is a redex context C and processes $\widetilde{M}, \widetilde{N}$ such that

—$P = C[P', \widetilde{M}]$, $Q = C[Q', \widetilde{N}]$;
—$\mathsf{bn}(C) \cap \mathsf{fn}(\mathcal{E}, r) = \emptyset$ and $\mathsf{fn}(C) \subseteq r$;
—if $s = \mathsf{bn}_1(C)$ then $P' \ \mathcal{X}_{\mathcal{E};r,s} \ Q'$;
—$\widetilde{M} \mathcal{E} \widetilde{N}$.

The condition $\mathsf{bn}(C) \cap \mathsf{fn}(\mathcal{E}, r) = \emptyset$ ensures us that there is no accidental clash between the bound names introduced by the context and existing names in the environment. (By appropriate renaming of the bound names in P and Q, the condition can in fact always be guaranteed.) Moreover, the condition ensures us that the free names in the context are names already known by the observer. The third condition, shows that the context can bind names that appear free in the processes P', Q' (names s); but then, these names should be taken into account in the bisimulation game between P' and Q'; further, by the

second condition, these names cannot appear free in $\mathcal{E}$ and cannot be in r. If, in addition, the context has only the hole $[\cdot]_1$, then we write $P\ \mathcal{X}^{\mathrm{c1}}_{\mathcal{E};r}\ Q$. Thus, $P\ \mathcal{X}^{\mathrm{c1}}_{\mathcal{E};r}\ Q$ holds if there is a redex context C such that

—$P = C[P']$, $Q = C[Q']$,
—$\mathsf{bn}(C) \cap \mathsf{fn}(\mathcal{E}, r) = \emptyset$ and $\mathsf{fn}(C) \subseteq r$;
—if $s = \mathsf{bn}_1(C)$ then $P'\ \mathcal{X}_{\mathcal{E};r,s}\ Q'$

Definition 6.36. An environmental relation $\mathcal{X}$ is an *environmental bisimulation up to context and* $=_{\mathrm{str}}$ if $M\ \mathcal{X}_{\mathcal{E};r}\ N$ implies:

(1) if $M \xrightarrow{\tau} M'$ then $N \Longrightarrow N'$ and $M' =_{\mathrm{str}} \mathcal{X}^{\mathrm{c}}_{\mathcal{E};r} =_{\mathrm{str}} N'$;
(2) if $M \xrightarrow{aP} M'$ with $a \in r$, and $(P, Q) \in (r\,;\,\mathcal{E})^\star$, then $N \overset{aQ}{\Longrightarrow} N'$ and $M' =_{\mathrm{str}} \mathcal{X}^{\mathrm{c}}_{\mathcal{E};r} =_{\mathrm{str}} N'$;
(3) if $M \xrightarrow{(\nu\widetilde{b})\overline{a}P} M'$ with $a \in r$ and $\widetilde{b}$ fresh, then $N \overset{(\nu\widetilde{c})\overline{a}Q}{\Longrightarrow} N'$, with $\widetilde{c}$ fresh, and $M' =_{\mathrm{str}} \mathcal{X}^{\mathrm{c}}_{\mathcal{E},(P,Q);r} =_{\mathrm{str}} N'$;
(4) for all $(P, Q) \in \mathcal{E}$ we have $P \mid M =_{\mathrm{str}} \mathcal{X}^{\mathrm{c1}}_{\mathcal{E};r} =_{\mathrm{str}} Q \mid N$;
(5) for all r' fresh (ie, not in $\mathsf{fn}(\mathcal{E}, M, N)$) we have $M\ \mathcal{X}_{\mathcal{E};r,r'}\ N$;
(6) the converse of (1-3), on N.

Note the restricted form of up-to context in condition (4); this is necessary to guarantee soundness of the technique. As a counterexample, take $P \overset{\mathrm{def}}{=} \overline{a}(b.\mathbf{0}).\mathbf{0}$ and $Q \overset{\mathrm{def}}{=} \overline{a}(c.\mathbf{0}).\mathbf{0}$. When both b and c are observable, P and Q are not bisimilar, as the values emitted, namely $b.\mathbf{0}$ and $c.\mathbf{0}$, can be played back producing a difference. Without the constraint in condition (4), however, the environmental relations with elements $(r\,;\,\emptyset\,;\,P\,;\,Q)$ and $(r\,;\,(b.\mathbf{0}, c.\mathbf{0})\,;\,\mathbf{0}\,;\,\mathbf{0})$, where $r = \{a, b, c\}$, would be an environmental bisimulation up to context and $=_{\mathrm{str}}$. Indeed, condition (4) would become void if we replaced $P \mid M =_{\mathrm{str}} \mathcal{X}^{\mathrm{c1}}_{\mathcal{E};r} =_{\mathrm{str}} Q \mid N$ with $P \mid M =_{\mathrm{str}} \mathcal{X}^{\mathrm{c}}_{\mathcal{E};r} =_{\mathrm{str}} Q \mid N$, as $M\ \mathcal{X}_{\mathcal{E};r}\ N$ implies $P \mid M =_{\mathrm{str}} \mathcal{X}^{\mathrm{c}}_{\mathcal{E};r} =_{\mathrm{str}} Q \mid N$, for all $(P, Q) \in \mathcal{E}$, using the context $[\cdot]_2 \mid [\cdot]_1$.

LEMMA 6.37. *If $\mathcal{X}$ is an environmental bisimulation up to context and $=_{\mathrm{str}}$, then $\mathcal{X} \subseteq \approx$.*

PROOF. Let $\mathcal{Y}$ be a bisimulation up to context and $=_{\mathrm{str}}$. We consider the relation $\mathcal{X}$ composed by all elements of the form

$$(r\,;\,(r\,;\,\mathcal{E})^\star\,;\,C[P, \widetilde{M}]\,;\,C[Q, \widetilde{N}])$$

such that

—$\mathsf{bn}(C) \cap \mathsf{fn}(\mathcal{E}, r) = \emptyset$ and $\mathsf{fn}(C) \subseteq r$,
—if $\mathsf{bn}_1(C) = s$, then $P\ \mathcal{Y}_{\mathcal{E};r,s}\ Q$,
—$\widetilde{M}\,\mathcal{E}\,\widetilde{N}$,
—all holes except $[\cdot]_1$ are guarded in C and C is a redex context.

We prove that $\mathcal{X}$ is a bisimulation up to restriction, $=_{\mathrm{str}}$, and environment.

Consider an element $(r\,;\,(r\,;\,\mathcal{E})^\star\,;\,C[P,\widetilde{M}]\,;\,C[Q,\widetilde{N}])$ of the relation. We have, for some s, C', $C[P,\widetilde{M}] =_{\rm str} \nu s\,(P \mid C'[\widetilde{M}])$, $C[Q,\widetilde{N}] =_{\rm str} \nu s\,(Q \mid C'[\widetilde{N}])$, with $(s \cup \mathsf{bn}(C')) \cap \mathsf{fn}(\mathcal{E}, r) = \emptyset$, $\mathsf{fn}(C') \subseteq s, r$, C' guarded, and $P\ \mathcal{Y}_{\mathcal{E}\,;r,s}\ Q$.

We consider the possible actions from $C[P,\widetilde{M}]$. There are 3 possibilities: the action comes from P alone, from $C'[\widetilde{M}]$ alone, or from an interaction between these 2 components. First the actions from P alone. (In all the proof we often use implicitly, without mentioning it, Lemma 6.3.)

—τ action, $P \stackrel{\tau}{\to} P'$. Thus $C[P,\widetilde{M}] \stackrel{\tau}{\to} M =_{\rm str} \nu s\,(P' \mid C'[\widetilde{M}])$.
Since $P\ \mathcal{Y}_{\mathcal{E}\,;r,s}\ Q$ we have $Q \Longrightarrow Q'$ and $P'\ \mathcal{Y}^{\rm c}_{\mathcal{E}\,;r,s}\ Q'$.
This means that there are $D, \widetilde{A}, \widetilde{B}$ such that
—$P' =_{\rm str} D[P'', \widetilde{A}]$, $Q' =_{\rm str} D[Q'', \widetilde{B}]$,
—$\mathsf{bn}(D) \cap \mathsf{fn}(\mathcal{E}, r, s) = \emptyset$ and $\mathsf{fn}(D) \subseteq r, s$,
—if $t = \mathsf{bn}_1(D)$ then $P''\ \mathcal{Y}_{\mathcal{E}\,;r,s,t}\ Q''$,
—$\widetilde{A}\mathcal{E}\widetilde{B}$.
Let D' be the context obtained by composing C and D in the obvious way. Then we have
$$M =_{\rm str} D'[P'', \widetilde{A}, \widetilde{M}], \text{ and } C[Q,\widetilde{N}] \Longrightarrow =_{\rm str} D'[Q'', \widetilde{B}, \widetilde{N}].$$
We also have:
—$\mathsf{bn}(D') \cap \mathsf{fn}(\mathcal{E}, r) = \emptyset$, and $\mathsf{fn}(D') \subseteq (\mathsf{fn}(C) \cup (\mathsf{fn}(D) - s)) \subseteq (r \cup r) = r$;
—$\mathsf{bn}_1(D') = s \cup t$, and $P''\ \mathcal{Y}_{\mathcal{E}\,;r,s,t}\ Q''$
—$\widetilde{A}, \widetilde{M}\mathcal{E}\widetilde{B}, \widetilde{N}$
—all holes except $[\cdot]_1$ are guarded in D' and D' is a redex context.
We can therefore conclude that $D'[P'', \widetilde{A}, \widetilde{M}]\ \mathcal{X}_{(r\,;\mathcal{E})^\star\,;r}\ D'[Q'', \widetilde{B}, \widetilde{N}]$ and we are done.

—Input action. This is similar (note that any input from $(r\,;\,\mathcal{E})^\star$ is also a legal input in the bisimulation game for $P\ \mathcal{Y}_{\mathcal{E}\,;r,s}\ Q$).

—Output action. Suppose $P \xrightarrow{(\nu \widetilde{b})\overline{a}P_1} P'$, with $a \in r$. Since $P\ \mathcal{Y}_{\mathcal{E}\,;r,s}\ Q$, we have
$$Q \stackrel{(\nu \widetilde{b})\overline{a}Q_1}{\Longrightarrow} Q'$$
with $\widetilde{b}, \widetilde{c}$ fresh and $P' =_{\rm str} \mathcal{X}^{\rm c}_{\mathcal{E},(P_1,Q_1)\,;r,s} =_{\rm str} Q'$.
This means that there are $D, \widetilde{A}, \widetilde{B}$ such that
—$P' = D[P'', \widetilde{A}]$, $Q' = D[Q'', \widetilde{B}]$,
—$\mathsf{bn}(D) \cap \mathsf{fn}(\mathcal{E}, P_1, Q_1, r, s) = \emptyset$, and $\mathsf{fn}(D) \subseteq r, s$,
—if $t = \mathsf{bn}_1(D)$ then $P''\ \mathcal{Y}_{\mathcal{E},(P_1,Q_1)\,;r,s,t}\ Q''$,
—$\widetilde{A}\mathcal{E}\widetilde{B}$.
We also have, for $s'_P = s \cap \mathsf{fn}(P_1)$, and $s''_P = s - s'_P$, and for $s'_Q = s \cap \mathsf{fn}(Q_1)$, and $s''_Q = s - s'_Q$:
$$\begin{array}{rl} C[P,\widetilde{M}] \xrightarrow{(\nu \widetilde{b}, s'_P)\overline{a}P_1} & =_{\rm str} \nu s''_P\,(P' \mid C'[\widetilde{M}]) =_{\rm str} \nu s''_P\,D'[P'', \widetilde{A}, \widetilde{M}] \\ C[Q,\widetilde{N}] \stackrel{(\nu \widetilde{b}, s'_Q)\overline{a}Q_1}{\Longrightarrow} & =_{\rm str} \nu s''_Q\,(Q' \mid C'[\widetilde{N}]) =_{\rm str} \nu s''_Q\,D'[Q'', \widetilde{B}, \widetilde{N}] \end{array}$$
where D' is the context obtained by composing C' and D. Making use of the up to restriction and up to environment, it is sufficient to show that
$$D'[P'', \widetilde{A}, \widetilde{M}]\ \mathcal{X}_{(r,s\,;\,\mathcal{E},(P_1,Q_1))^\star\,;r,s}\ D'[Q'', \widetilde{B}, \widetilde{N}]$$

This holds because
— $\mathsf{bn}(D') \cap (r, s) = \emptyset$ and $\mathsf{fn}(D') \subseteq r, s$,
— $\mathsf{bn}_1(D') = t$ and $P''\ \mathcal{Y}_{\mathcal{E},(P_1,Q_1);r,s,t}\ Q''$,
— $\widetilde{A}\mathcal{E}\widetilde{B}$ and $\widetilde{M}\mathcal{E}\widetilde{N}$.

Now we consider the actions coming from $C'[\widetilde{M}]$ alone.

— τ action. Let $C''[\widetilde{M}]$ and $C''[\widetilde{N}]$ be the resulting terms (here we are appealing to Lemma 6.5). We have (doing as in Lemma 6.6), for some D guarded and a multiset J

$$C'' =_{\mathrm{str}} \prod_{j\in J}[\cdot]_j \Big| D.$$

Hence

$$\nu s\,(P' \mid C''[\widetilde{M}]) =_{\mathrm{str}} \nu s\left(P' \Big| \prod_j M_j \Big| D[\widetilde{M}]\right)$$

and similarly

$$\nu s\,(Q' \mid C''[\widetilde{N}]) =_{\mathrm{str}} \nu s\left(Q' \Big| \prod_j N_j \Big| D[\widetilde{N}]\right).$$

Using clause (4) of bisimulation up-to, as $\widetilde{M}\mathcal{E}\widetilde{N}$, we have

$$P' \Big| \prod_j M_j =_{\mathrm{str}} D'[P'']$$

$$Q' \Big| \prod_j N_j =_{\mathrm{str}} D'[Q'']$$

where D' is a redex context, and $P''\ \mathcal{Y}_{\mathcal{E};r,s,t}\ Q''$, where $t = \mathsf{bn}_1(D')$.
In conclusion, we end up, modulo $=_{\mathrm{str}}$ with derivatives

$$\nu s\,(D'[P''] \mid D[\widetilde{M}])$$

and

$$\nu s\,(D'[Q''] \mid D[\widetilde{N}])$$

which satisfy all conditions of $\mathcal{X}$.

— Input actions: similar

— Output actions. This also is similar. All names extruded must be fresh, since $(\mathsf{bn}_1(C'), s) \cap \mathsf{fn}(\mathcal{E}, r) = \emptyset$. Call t such extruded names. Then we have also $P\ \mathcal{X}_{\mathcal{E};r,s,t}\ Q$ (clause (5) of bisimulation), and the reasoning is as above.

Now the case of an interaction between P and $C[\widetilde{M}]$. This interaction is given by an output action from one of the components, and an input action from the other. Both such actions can be dealt with similarly to above. (A possible difference is that the name along which the interaction occurs can be a name in s, but this does not give a problem)

Finally, clause (4) of bisimulation can be accommodated similarly to above (see action from $C[\widetilde{M}]$, where we used such clause) □

This technique can be further strengthened by combining it with the up-to environment, or by using *strong environmental bisimilarity* in place of $=_{\mathrm{str}}$. In strong bisimilarity only strong transitions are used in the bisimulation game; the basic properties, including congruence, are established in the same manner as for the weak bisimilarity. We believe it would also be possible to use the *expansion* relation, in place of structural equivalence, along the lines of the techniques in Section 3.6.

6.7 An Example

We consider two ways of modelling the replication operator in HOπ. That is, given a process P we want a term that makes P available an unbounded number of times. The two terms that achieve this are A_P and B_P, defined as follows, where $a \notin \mathsf{fn}(P)$.

$$A_P \stackrel{\mathrm{def}}{=} \nu a\,(K_P \mid \overline{a}K_P) \qquad \text{where } K_P \stackrel{\mathrm{def}}{=} a(x).(P \mid x \mid \overline{a}x)$$

and we have

$$A_P \stackrel{\tau}{\rightarrow} =_{\mathrm{str}} P \mid A_P \stackrel{\tau}{\rightarrow} =_{\mathrm{str}} P \mid P \mid A_P \stackrel{\tau}{\rightarrow} \cdots$$

We set

$$B_P \stackrel{\mathrm{def}}{=} \nu a\,(H \mid \overline{a}(P \mid H)) \qquad \text{where } H \stackrel{\mathrm{def}}{=} a(x).(x \mid \overline{a}x)$$

and we have

$$B_P \stackrel{\tau}{\rightarrow} =_{\mathrm{str}} P \mid B_P \stackrel{\tau}{\rightarrow} =_{\mathrm{str}} P \mid P \mid B_P \stackrel{\tau}{\rightarrow} \cdots$$

The internal structure of A_P and B_P is however different. We can prove $A_P \simeq B_P$ using the environmental relation that has just the tuples

$$(r\,;\,\emptyset\,;\,A_P\,;\,B_P),$$

for $\mathsf{fn}(P) \subseteq r$ and $a \notin r$, and proving that this relation is an environmental bisimulation up to context and $=_{\mathrm{str}}$.

The only action that A_P can perform is, up to $=_{\mathrm{str}}$,

$$A_P \stackrel{\tau}{\rightarrow} =_{\mathrm{str}} P \mid A_P$$

and similarly for B_P it is

$$B_P \stackrel{\tau}{\rightarrow} =_{\mathrm{str}} P \mid B_P.$$

Now we are done, because $P \mid B_P \;\; \mathcal{X}^{\mathrm{c}}_{\emptyset;r} \;\; P \mid A_P$.

The use of up-to context allows us to work with a very simple relation—leaving the parameterised set of free names aside, it is a singleton relation—and to have an empty environment. A full bisimulation would have had to take into account, for instance, all possible actions and derivatives from P.

A variant of the above example is the proof of the equality between the terms $\overline{c}A$ and $\overline{c}B$, where

$$A \stackrel{\mathrm{def}}{=} b(y).(\nu a\,(K \mid \overline{a}K)) \qquad \text{for } K \stackrel{\mathrm{def}}{=} a(x).(y \mid x \mid \overline{a}x)$$

and

$$B \stackrel{\mathrm{def}}{=} b(y).(\nu a\,(H \mid \overline{a}(y \mid H)))$$

and H is as above. Terms $\overline{c}A$ and $\overline{c}B$ send on c terms that can receive a process at b and then replicate this process.

An environmental bisimulation up to context and $=_{\mathrm{str}}$ that proves $\overline{c}A \simeq \overline{c}B$ contains the tuples of the form

$$(r\,;\,\emptyset\,;\,\overline{c}A\,;\,\overline{c}B)$$

with $a \notin r$ and $b, c \in r$, and

$$\left(r; \{(A, B)\}; \prod_{i=1}^{n} A; \prod_{i=1}^{n} B\right)$$

with $a \notin r$ and $b \in r$ and $n \geq 0$, and

$$\left(r; \{(A, B)\}; A_P \middle| \prod_{i=1}^{n} A; B_Q \middle| \prod_{i=1}^{n} B\right) \tag{6}$$

with $a \notin r$ and $b \in r$ and $n \geq 0$ and $(P, Q) \in (r\,;\,\{A, B\})^\star$, and with A_R and B_R, for an arbitrary process R, defined as in the previous example. (The need of the components $\prod_{i=1}^{n} A$ and $\prod_{i=1}^{n} B$ comes from clause (4) of the bisimulation up-to.) The reasoning needed in the proof is similar to that of the previous example.

Once more, the use of up-to techniques simplifies the proof substantially. Defining a plain environmental bisimulation would be rather hard. For instance, we would have to take into account actions performed by processes such as P and Q that appear in (6): these processes are in $(r\,;\,\{A, B\})^\star$, but their derivatives might not be so. This may occur when P and Q perform an input action; as the environment is non-empty, the terms that they receive in the input are in $(r\,;\,\{A, B\})^\star$ but need not be syntactical equal.

6.8 Extensions

We have examined a few extensions of the HOπ language, and variants of the bisimulation, and checked that they can be accommodated with minor modifications to the definitions and results presented in this section. They include: *strong bisimilarity* (essentially this amounts to changing all weak transitions into strong ones); *late bisimilarity* (as opposed to the early bisimilarity we have considered); communication of process abstractions (in the CHOCS-like language of this section, only processes can be passed around; in HOπ, besides processes, abstractions, that is, parameterised processes, can be passed too; moreover, these abstractions can be of arbitrary high order; this requires adding constructs for abstraction and application to the syntax as in full HOπ [Sangiorgi 1992; Sangiorgi and Walker 2001]).

7. RELATED WORK

λ-calculi. Forms of bisimulation with an environment have been used (under no explicit name) by Sumii and Pierce for λ-calculi with perfect encryption [Sumii and Pierce 2007a] and data abstraction [Sumii and Pierce 2007b], inspired by bisimulations for typed π-calculus [Boreale and Sangiorgi 1998], polymorphic π-calculus [Pierce and Sangiorgi 2000] and spi-calculus [Abadi

and Gordon 1998]. However, their bisimulations were not able to handle higher-order functions. To address this issue, Sumii and Pierce [Sumii and Pierce 2007b, Section 7] proposed a rather complex variant of their bisimulations with induction on the tree height of evaluation derivation, and an up-to-context technique built into the definition of bisimulations. Koutavas and Wand (KW in short) later gave a clearer account of this approach, in the λ-calculus with store of Section 5 [Koutavas and Wand 2006b], and in a language with objects [Koutavas and Wand 2006a].

KW has an induction on the evaluation of terms to values and an up-to context technique which are nicely hardwired into the definition of the bisimulation. However, KW relies on big-step semantics, which does not scale to languages with nondeterminism or concurrency. Further KW breaks the monotonicity of the generating functional of the bisimulation (they have not only $\mathcal{E}$ but also $\mathcal{X}$ in a negative position because of the induction), and therefore requires extra proofs to guarantee that bisimilarity exists (and is transitive). Sumii and Pierce, as well as Koutavas and Wand, gave only indirect proofs of these properties, via the correspondence with contextual equivalence. This strategy may not always be possible in other languages (non-deterministic ones, for example). For similar reasons, it could be difficult to enhance KW bisimulation method via up-to techniques (for instance, in Section 3.8 we made a nontrivial use of up-to expansion, which is not available in KW; hence a proof of this equivalence with KW requires an infinite relation, rather than a singleton relation as in our proof).

In general, as the examples we have analyzed show (such as the example in Section 5.6, which is a minor modification of Koutavas and Wand's example [Koutavas and Wand 2006b], Section 6.2), our use of expansion (or the related up-to reduction) can play the role of the induction in KW. Intuitively, KW induction allows one to reason on terms which reduce in fewer steps to values. Up-to expansion gives a similar possibility.

The bisimulation clause on functions of environmental bisimulation is reminiscent of logical relations; see, for example, Mitchell [1996, Chapter 8] and Pitts [2005]. (The analogy is stronger for the BA-bisimulations discussed in the Introduction; we recall that in logical relations two functions are related if they map related arguments to related results.) However, logical relations represent a type-directed technique and as such remain quite different from bisimulations, which can be untyped. Logical relations work very well in pure simply-typed or polymorphic λ-calculus [Reynolds 1983], but they become complex—requiring domain theory or step indices [Appel and McAllester 2001]—and/or incomplete in richer languages with recursive types [Birkedal and Harper 1999; Crary and Harper 2000; Appel and McAllester 2001; Ahmed 2006], existential types [Pitts 2005; Crary and Harper 2000; Ahmed et al. 2003; Ahmed 2006], store [Ahmed et al. 2003], or encryption [Sumii and Pierce 2003], to give just a few examples. A main advantage of environmental bisimulations over logical relations is that it is relatively straightforward to achieve completeness in a variety of higher-order languages. For instance, as recently demonstrated by Sumii [Sumii 2009], our environmental bisimulations can be adapted to λ-calculus with existential types and references to obtain a complete

characterization of contextual equivalence, while known logical relations for the same language are incomplete [Ahmed et al. 2009]. A possible disadvantage of environmental bisimulations is that equivalence proofs can be complex, but as demonstrated in this article (see also Sumii [2009], where all the examples in [Ahmed et al. 2009] are proved by environmental bisimulations), we can often mitigate this problem by appropriate up-to techniques. We may regard an exception to this Koutavas and Wand's orginal example in [Koutavas and Wand 2006b], Section 6.2: as pointed out at the end of Section 5.6, to prove it we have first to β-expand some subterms and then appeal to the transitivity of bisimilarity. We leave it for future work whether there is some more powerful up-to technique that allows us to avoid such β-expansions. (For concrete proofs, however, simple "massaging" of the terms, such as β-expansions, are quite acceptable.)

Concurrent languages. There are only a few concurrent higher-order languages for which bisimulation techniques have been given; usually the bisimilarity is either a form of higher-order bisimulation and Howe's technique [Howe 1996] is used to prove congruence [Ferreira et al. 1998; Baldamus and Frauenstein 1995; Godskesen and Hildebrandt 2005], or it is a form of context bisimulation or normal bisimulation (e.g., [Sangiorgi 1992, 1996; Merro and Hennessy 2002; Merro and Nardelli 2005; Jeffrey and Rathke 2004, 2005]). Howe's technique appears to have limitations in concurrency. It seems to be sensitive to the choice of the bisimilarity; in particular it is often troublesome if the bisimilarity is not both in the "delay" and in the "late" style (recent advances here have been made by Godskesen and Hildebrandt [2005] and Lenglet et al. [2009]). The technique also seems to be sensitive to the structure of terms that are allowed to interact. For instance, non-functional patterns of rules, in which the composition construct is persistent, may give problems (see Jeffrey and Rathke [2004] for a discussion).

In *context bisimulation* [Sangiorgi 1996; Merro and Hennessy 2002; Merro and Nardelli 2005; Godskesen and Hildebrandt 2005], whenever an interaction is produced between the tested processes and the observer, all possible contexts that have originated the action from the observer are taken into account. For instance, clause (3) for output actions would become:

—if $M \xrightarrow{(\nu\tilde{b})\overline{a}P} M'$ then there are $Q, N', \tilde{c}$ such that $N \overset{(\nu\tilde{c})\overline{a}Q}{\Longrightarrow} N'$,
and for all G with $\mathsf{fv}(G) \subseteq X$, $\nu\tilde{b}\,(G\{^{P}/_{X}\} \mid M')\ \mathcal{R}\ \nu\tilde{c}\,(G\{^{Q}/_{X}\} \mid N')$.

The quantification over all recipients G is a heavy demand, close to explicitly requiring that the bisimulation is preserved by parallel composition.

To simplify context bisimulation, *normal bisimulation* has been proposed. The technique appears in Sangiorgi [1992]; Jeffrey and Rathke [2005] have then improved it, using an elegant symbolic treatment of triggers; this has allowed them to overcome certain limitations of the original technique (for instance, problems with recursive types). Normal bisimulation replaces the exchange of processes with the exchange of special processes called *triggers* and essentially implements a dynamic translation of higher-order communication into first-order, for a trigger that is communicated acts like a name-pointer to

a process. Normal bisimulation does not explicitly use environments; these are hardwired inside the tested processes, by means of triggers. The main difference with environmental bisimulation is in the input clause: normal bisimulation does not use universal quantifications, and is therefore simpler. It is possible that for environmental bisimulation the same simplification could be derived as a further form of up-to technique; we leave the details for future work.

Possible advantages of environmental bisimulation are the following. First, it is straightforward to adapt environmental bisimulation to the strong case (where all transitions, including internal steps, are treated equally): all definitions and results we have presented can be adapted to strong bisimilarity (for closed terms, this essentially amounts to changing all weak transitions into strong ones; open terms require more care, but the proofs remain similar). Adapting normal bisimulation to the strong case is sometimes delicate [Lenglet 2010] because the transformation of process communications into trigger communications is valid for weak, but not for strong, equivalences [Cao 2006]. Second, the proof of congruence of environmental bisimulation is simpler and more direct. Related to this is the development of up-to techniques, in particular up-to contexts. Finally, normal bisimulation relies on the possibility of encoding higher-order communications using triggers; this might not be possible, or might be difficult to achieve, in extensions of HOπ or in calculi different from HOπ. For instance, the encoding does not work in presence of dynamic operators such as (unguarded) choice, or calculi with constructs of location or for distribution.

To the best of our knowledge, the only higher-order concurrent calculus for which up-to-context techniques had been derived is the Ambient calculus [Merro and Nardelli 2005], for a form of context bisimulation. Ambients however represent a special case of higher-order calculus, for processes can move but cannot be communicated. Moving a process is quite different from communicating it as in HOπ: in the former case the process will always be run, immediately and exactly once; in the latter case, in contrast, the process may be copied, and it is the recipient of the process that decides when and where to run each copy. Thus the problems of congruence of bisimulation, and the related problems for up-to contexts, only show up in a limited form in Ambients.

We show in Sangiorgi et al. [2007] that, in the sequential languages without references, environmental bisimulations can be somewhat simplified, essentially merging the environment with the plain terms. In the resulting bisimulation, called logical bisimulation, the generating functional is however nonmonotone, but has nevertheless a greatest fixed-point that coincides with contextual equivalence (and hence also with environmental bisimilarity). The possibility of using this technique seems however limited to the pure sequential languages.

8. CONCLUSIONS AND FUTURE WORK

In this article we have developed the basic theory of environmental bisimulations. In summary, with environmental bisimulations we aim at (1) maintaining the definition of the bisimulation as simple as possible, so to facilitate proofs of its basic properties (in particular congruence and up-to-context techniques,

which are notoriously hard in higher-order languages); and (2) separately developing enhancements of the bisimulation method, so as to have simple bisimilarity proofs between terms.

We have examined some basic up-to techniques—those that appeared to us as the most useful—in small-step and big-step versions, and combinations of them, including up-to context techniques because they are important but notoriously hard to derive in higher-order languages. Further up-to techniques could be added. For instance, in the imperative language of Section 5, an up-to technique that manipulates the store could further simplify proofs such as that of the example in Section 5.6.

Bisimulation and coinductive techniques are known to represent a hard problem in higher-order languages. While we certainly would not claim that environmental bisimulations are definitely better than applicative bisimulations or other co-inductive techniques in the literature (indeed, probably a single *best* bisimulation for this does not exist), we believe it is important to explore different approaches and understand their relative merits. In this respect it is encouraging that environmental bisimulation work on a variety of calculi, and the proof of its basic properties, and associated up-to techniques, are fairly easy to transport. To the best of our knowledge, none of the previous techniques can handle such a variety of languages. In the future we plan to consider more sophisticated concurrent languages. For instance, the passivation construct of the Kell Calculus [Schmitt and Stefani 2004] appears challenging.

Although the basic idea of environmental bisimulation is the same on all calculi considered in the paper, the bisimilarity clauses can differ depending on the features of the calculus (e.g., the calculi with concurrency and references). It would be interesting to formulate environmental bisimulation in a more abstract manner, and to derive the concrete definitions presented in this paper as special instances of it. For this, Milner's bigraphs [Milner 2006] would be a candidate framework.

APPENDIX

A. PROOFS FROM SECTION 3

FACT A.1. *Let $\mathcal{E}$ be a relation on closed terms, and suppose $\widetilde{V}\ \mathcal{E}\ \widetilde{W}$. For any context C, if $C[\widetilde{V}]$ is a value, then also $C[\widetilde{W}]$ is a value, and $C[\widetilde{V}]\ \mathcal{E}^\star\ C[\widetilde{W}]$.*

If C is a context with multiple holes, then $C[\cdot][\widetilde{V}]$ is the context obtained by replacing the first hole, $[\cdot]_1$, with $[\cdot]$ and filling the remaining holes with the values $\widetilde{V}$ in the usual way.

LEMMA A.2. *Suppose $C[\cdot][\widetilde{V}]$ is an evaluation context, and M is not a value. We have:*

(1) If $M \longrightarrow M'$ then $C[M, \widetilde{V}] \longrightarrow C[M', \widetilde{V}]$;
(2) If $C[M, \widetilde{V}] \longrightarrow P$ then $P = C[M', \widetilde{V}]$ for some M' such that $M \longrightarrow M'$.

PROOF. By induction on C. All cases are easy. □

Lemmas 3.15 and 3.16 are proved simultaneously, from the following lemma.

LEMMA A.3. *(1) $V \approx_{\mathcal{E}} W$ implies $C[V] \approx_{\mathcal{E}} C[W]$, for all C.*
(2) $M \approx_{\mathcal{E}} N$ implies $C[M] \approx_{\mathcal{E}} C[N]$, for all evaluation contexts C.

PROOF. Suppose $\mathcal{Y}$ is a bisimulation, and take

$$\begin{array}{rl}\mathcal{X} \stackrel{\text{def}}{=} & \{(\widehat{\mathcal{E}^\star}, C[M, \widetilde{V}], C[N, \widetilde{W}]) \ \text{s.t.} \\ & M \ \mathcal{Y}_{\mathcal{E}} \ N, \\ & C \text{ is an evaluation context on the first hole,} \\ & V_i \ \mathcal{E} \ W_i\} \\ & \cup \{(\widehat{\mathcal{E}^\star}, C[\widetilde{V}], C[\widetilde{W}]) \ \text{s.t.} \ \ \mathcal{E} \in \mathcal{Y} \text{ and } V_i \ \mathcal{E} \ W_i\} \\ & \cup \{\widehat{\mathcal{E}^\star} \ \text{s.t.} \ \ \mathcal{E} \in \mathcal{Y} \ \}\end{array}$$

We show that this is a bisimulation up to environment. We use a few times the following result (we exploit here the peculiarity of clause (2) of environmental bisimulations on abstractions).

FACT A.4. *If $\lambda x.P \ \mathcal{E}^\star \ \lambda x.Q$, and $M \ \mathcal{E}^\star \ N$, and $\mathcal{E} \in \mathcal{Y}$ (where $\mathcal{Y}$ is the bisimulation used in the definition of $\mathcal{X}$), then $(P\{^M/_x\}, Q\{^N/_x\}) \in \mathcal{X}_{\mathcal{E}^\star}$.*

PROOF. We distinguish two cases: $\lambda x.P \ \mathcal{E} \ \lambda x.Q$, and the case $P = D[\widetilde{V'}]$ and $Q = D[\widetilde{W'}]$ for $\widetilde{V'} \ \mathcal{E} \ \widetilde{W'}$. In the first case, since $\mathcal{Y}$ is a bisimulation, we get $(P\{^M/_x\}, Q\{^N/_x\}) \in \mathcal{Y}_{\mathcal{E}}$, which implies $(P\{^M/_x\}, Q\{^N/_x\}) \in \mathcal{X}_{\mathcal{E}^\star}$ (using the first set in the definition of $\mathcal{X}$). In the second case, we have $(D[\widetilde{V'}]\{^M/_x\}, D[\widetilde{W'}]\{^N/_x\}) \in \mathcal{X}_{\mathcal{E}^\star}$ (using the second set of $\mathcal{X}$). □

We first prove the bisimulation for elements of the form

$$(\widehat{\mathcal{E}^\star}, C[\widetilde{V}], C[\widetilde{W}]). \tag{7}$$

On these elements, we do not make use of the "up-to environment." Clause (1.b) is immediate: if $C[\widetilde{V}]$ is a value, then also $C[\widetilde{W}]$ is a value and they are in $\widehat{\mathcal{E}^\star}$ (Fact A.1). This is sufficient, because $\widehat{\mathcal{E}^\star} \in \mathcal{X}$.

Clause (1.a) is proved by an induction on C. The details are easy: the only possible case is $C = C_1 C_2$.

— Suppose $C_1[\widetilde{V}] \longrightarrow P$. Then by induction $C_1[\widetilde{W}] \Longrightarrow Q$ with $(P, Q) \in \mathcal{X}_{\mathcal{E}^\star}$. As $(C_2[\widetilde{V}], C_2[\widetilde{W}]) \in \mathcal{E}^\star$, also $(PC_2[\widetilde{V}], QC_2[\widetilde{W}]) \in \mathcal{X}_{\mathcal{E}^\star}$.
— Suppose $C_1[\widetilde{V}]$ is a value, say $\lambda x.P$, and $C_1[\widetilde{V}]C_2[\widetilde{V}] \longrightarrow P\{^{C_2[\widetilde{V}]}/_x\}$. Then also $C_1[\widetilde{W}]$ is a value (Fact A.1), say $\lambda x.Q$, and $C_1[\widetilde{W}]C_2[\widetilde{W}] \longrightarrow Q\{^{C_2[\widetilde{W}]}/_x\}$. We can now conclude $(P\{^{C_2[\widetilde{V}]}/_x\}, Q\{^{C_2[\widetilde{W}]}/_x\}) \in \mathcal{X}_{\mathcal{E}^\star}$ using Fact A.4.

Now elements of the form

$$(\widehat{\mathcal{E}^\star}, C[M, \widetilde{V}], C[N, \widetilde{W}]). \tag{8}$$

Suppose M is a value. In this case, $N \Longrightarrow W$, for some W, and we have $\mathcal{E}' \in \mathcal{Y}$ for $\mathcal{E}' = \mathcal{E} \cup \{(M, W)\}$. By Lemma A.2(1), also $C[N, \widetilde{W}] \Longrightarrow C[W, \widetilde{W}]$. Now

$$(\widehat{\mathcal{E}'^\star}, C[M, \widetilde{V}], C[W, \widetilde{W}])$$

is an element of the form (7). Using this property, and the fact that the bisimulation clauses on elements of the form (7) hold (as we just proved) it is immediate to get that the bisimulation clauses also hold on elements of the form (8), up to environment. (Here we need that $\mathcal{X}$ is an environmental bisimulation up to environment, for we are referring to elements of the form (7) but with $\mathcal{E}'$ in place of $\mathcal{E}$, and $\mathcal{E} \subseteq \mathcal{E}'$ hence also $\mathcal{E}^{\widehat{\star}} \subseteq \mathcal{E}'^{\widehat{\star}}$.)

On the other hand, if M is not a value, then neither is $C[M, \widetilde{V}]$. Therefore $C[M, \widetilde{V}] \longrightarrow P$, for some P, and only clause (1.b) of the definition of bisimulation applies. By Lemma A.2(2), $P = C[M', \widetilde{V}]$ with $M \longrightarrow M'$. By definition of bisimulation, $N \Longrightarrow N'$ and $M' \; \mathcal{Y}_{\mathcal{E}} \; N'$, and by Lemma A.2(1), $C[N, \widetilde{W}] \Longrightarrow C[N', \widetilde{W}]$. We are done, since

$$(\mathcal{E}^{\widehat{\star}}, C[M', \widetilde{V}], C[N', \widetilde{W}]) \in \mathcal{X}.$$

Finally, we have to check elements of $\mathcal{X}$ of the form $\mathcal{E}^{\widehat{\star}}$. Take $\lambda x.\, P \; \mathcal{E}^{\widehat{\star}} \; \lambda x.\, Q$ and $M \; \mathcal{E}^{\star} \; N$. We conclude $(P\{^M/_x\}, Q\{^N/_x\}) \in \mathcal{X}_{\mathcal{E}^{\widehat{\star}}}$ using Fact A.4. □

COROLLARY A.5. *$M \approx_{\mathcal{E}} N$ implies $ML \approx_{\mathcal{E}} NL$, for all $L \in \Lambda^{\bullet}$.*

Now we turn to prove Lemma 3.22. We prepare for this a few lemmas.

LEMMA A.6. *If $\lambda x.\, M \simeq \lambda x.\, N$ then for all P we have $M\{^P/_x\} \simeq N\{^P/_x\}$.*

PROOF. We use definitions of the bisimulations and weakening on the environments. □

LEMMA A.7. *If $\lambda x.\, M \simeq^{\star} \lambda x.\, N'$ and $\lambda x.\, N' \simeq \lambda x.\, N$, then for all $M_1 \simeq^{\star} N_1$ we have*

$$M\{^{M_1}/_x\} \;\simeq^{\star}\simeq\; N\{^{N_1}/_x\}.$$

PROOF. We have either $\lambda x.\, M \simeq \lambda x.\, N'$, or $\lambda x.\, M$ and $\lambda x.\, N'$ have a common nonempty context. In the first case, we have

$$M\{^{M_1}/_x\} \simeq^{\star} M\{^{N_1}/_x\}$$

and, since $\lambda x.\, M \simeq \lambda x.\, N'$ and $\simeq$ is a bisimulation, using Lemma A.6 we have:

$$\simeq N'\{^{N_1}/_x\}$$

and then, from $\lambda x.\, N' \simeq \lambda x.\, N$ and again Lemma A.6,

$$\simeq N\{^{N_1}/_x\}$$

and the result follows from the transitivity of $\simeq$. In the second case, we have:

$$M\{^{M_1}/_x\} \simeq^{\star} N'\{^{N_1}/_x\}$$

and then we can proceed as in the previous case. □

LEMMA A.8. *If $M \simeq N$ and $M \Longrightarrow \lambda x.\, M'$ then there is N' s.t. $N \Longrightarrow \lambda x.\, N'$ and $\lambda x.\, M' \simeq \lambda x.\, N'$.*

PROOF. The interesting case is when M is already an abstraction. Let $\mathcal{X}$ be a bisimulation that includes the element $(\emptyset, M, N)$. There must be N' s.t. $N \Longrightarrow \lambda x.\, N'$ and $\{(M, \lambda x.\, N')\} \in \mathcal{X}$. Then also $\mathcal{X} \cup \{(\emptyset, M, \lambda x.\, N')\}$ is a bisimulation. □

LEMMA A.9. *For any context C and closed terms $\lambda x.M$ and N, it holds that*

$$C[(\lambda x.M)N] \simeq C[M\{^N/_x\}].$$

PROOF. In this proof, we write $P >_\beta Q$ if Q is obtained from P by β-converting some of its closed subterms; formally, there is a multihole context C and closed terms $\lambda x.P_i$, and Q_i (for $0 \leq i \leq n$) such that $P = C[(\lambda x.P_1)Q_1]\ldots[(\lambda x.P_n)Q_n]$ and $Q = C[P_1\{^{Q_1}/_x\}]\ldots[P_n\{^{Q_n}/_x\}]$.

Now, for the proof of the lemma, we take the environmental relation consisting of all triples

$$(\mathcal{E}, M, N)$$

with $M >_\beta N$ and $\mathcal{E} \subseteq >_\beta$, and all sets $\mathcal{E}$ with $\mathcal{E} \subseteq >_\beta$. The proof that this relation is an environmental bisimulation are tedious but straightforward: essentially, since $>_\beta$ operates on closed terms only, if $M >_\beta N$ and $M \longrightarrow M'$, then this step is either produced by a β-reduction that in N had already been performed, in which case N need not move, or the step has consumed a β-redex that appears also in N (though it is not identical because there might be β-converted subterms), in which case N too can consume this β-redex and the resulting term is still in the relation $>_\beta$ with M'. □

LEMMA A.10 (LEMMA 3.22 IN THE MAIN TEXT). *$M \simeq N$ implies, for all contexts C, $C[M] \approx_{\widehat{\simeq}} C[N]$.*

PROOF. The following relation is a bisimulation up-to bisimilarity:

$$\mathcal{X} \stackrel{\text{def}}{=} \{(\widehat{\simeq^\star}, C[\widetilde{M}], C[\widetilde{N}]) \text{ s.t. } \widetilde{M} \simeq \widetilde{N}\} \cup \{\widehat{\simeq^\star}\}$$

Consider $(\widehat{\simeq^\star}, C[\widetilde{M}], C[\widetilde{N}])$. We check clauses (1.a) and (1.b) by induction on the size of the context (note that for (1.b) we thus have to prove that if $C[\widetilde{M}]$ is a value then there are W, W' s.t. $C[\widetilde{N}] \Longrightarrow W$, with $C[\widetilde{M}] \simeq^\star W' \simeq W$. We write $M\mathcal{X}N$ if $(\simeq^\star, M, N) \in \mathcal{X}$ (that is, it must be $M \simeq^\star N$).

Case $C = [\cdot]_i$: easy (in both clauses, the derivatives are in $\simeq$, for clause (1.b) we use Lemma A.8).

Case $C = \lambda x.C'$: easy.

Case $C = C_1C_2$.

—Suppose $C_1[\widetilde{M}] \longrightarrow P$, and therefore $C[\widetilde{M}] \longrightarrow PC_2[\widetilde{M}]$.

By induction, $C_1[\widetilde{N}] \Longrightarrow Q$ and there is Q' with $P\ \mathcal{X}\ Q'$ and $Q \simeq Q'$. Hence

$$C[\widetilde{N}] \Longrightarrow QC_2[\widetilde{N}]$$

and

$$PC_2[\widetilde{M}]\ \mathcal{X}\ Q'C_2[\widetilde{N}]\,.$$

We can conclude

$$Q'C_2[\widetilde{N}] \simeq QC_2[\widetilde{N}]$$

using Corollary A.5.

—Suppose $C_1[\widetilde{M}] = \lambda x.P$, and $C[\widetilde{M}] \longrightarrow P\{^{C_2[\widetilde{M}]}/_x\}$.

Then by induction, $C_1[\widetilde{N}] \Longrightarrow \lambda x.Q$ with $\lambda x.P \simeq^\star \lambda x.Q'$ and $\lambda x.Q \simeq \lambda x.Q'$, for some Q'. We can then conclude using Lemma A.7.

Finally, consider elements of $\mathcal{X}$ from $\simeq^{\widehat{\star}}$, say $(\lambda x.\, P, \lambda x.\, Q)$, and take $M \simeq^{\star} N$. The conclusion $P\{^{M}/_{x}\}\ \mathcal{X}{\simeq}\ Q\{^{N}/_{x}\}$ follows from reflexivity of $\simeq$ and Lemma A.7. □

B. PROOFS OF LEMMAS 5.3 AND 5.4

In this section, we prove Lemmas 5.3 and 5.4. As for the pure λ-calculi, in Section A, the two lemmas are proved simultaneously, and reasoning by induction on the size of the contexts. The additional feature of the imperative calculus make the proof longer and more tedious; therefore, although the structure of the proof is the same, we give all details.

We first mention the "up to environment" technique, which we will then use for the proof of the main result. Definition and proof of soundness of the techniques are as in pure λ-calculi. We only report the definition.

Definition B.1. An environment relation $\mathcal{X}$ is a *bisimulation up to environment* if

(1) $\langle s\,;\, M\rangle\ \mathcal{X}_{\mathcal{E}}\ \langle t\,;\, N\rangle$ implies:
  (a) if $\langle s\,;\, M\rangle \longrightarrow \langle s'\,;\, M'\rangle$ then $\langle s'\,;\, M'\rangle\ \mathcal{X}_{\mathcal{E}'}\ \langle t\,;\, N\rangle$ for some $\mathcal{E}'$ with $\mathcal{E} \subseteq \mathcal{E}'$;
  (b) if $M = V$ then $\langle t\,;\, N\rangle \Longrightarrow \langle t'\,;\, W\rangle$ and $V\,\mathcal{E}'\,W$ for some $\mathcal{E}'$ with $\mathcal{E} \subseteq \mathcal{E}'$ and $(\mathcal{E}', s, t') \in \mathcal{X}$;

(2) if $(\mathcal{E}, s, t) \in \mathcal{X}$ then for all $(V, W) \in \mathcal{E}$ we have:
  (a) $V = c$ iff $W = c$;
  (b) $V = (V_1, \ldots, V_n)$ implies $W = (W_1, \ldots, W_n)$ and for all i, we have $V_i\,\mathcal{E}'\,W_i$ for $\mathcal{E} \subseteq \mathcal{E}'$ and $(\mathcal{E}', s, t) \in \mathcal{X}$;
  (c) for all fresh l, l', we have $(\mathcal{E}', s[l = 0], t[l' = 0]) \in \mathcal{X}$, for $\mathcal{E} \subseteq \mathcal{E}'$ and $(l, l') \in \mathcal{E}'$;
  (d) if $V = l$ then $W = l'$, for some l', and moreover,
    i. $(s(l), t(l')) \in \mathcal{E}'$, for $(\mathcal{E}', s, t) \in \mathcal{X}$ and $\mathcal{E} \subseteq \mathcal{E}'$;
    ii. for all $(V_1, W_1) \in \mathcal{E}^{\widehat{\star}}$, we have $(\mathcal{E}', s[l = V_1], t[l' = W_1]) \in \mathcal{X}$ and $\mathcal{E} \subseteq \mathcal{E}'$;
  (e) if $V = \lambda x.\, P$ then $W = \lambda x.\, Q$ and for all $V_1, W_1 \in \mathcal{E}^{\widehat{\star}}$ it holds that $\langle s\,;\, P\{^{V_1}/_{x}\}\rangle\ \mathcal{X}_{\mathcal{E}'}\ \langle t\,;\, Q\{^{W_1}/_{x}\}\rangle$, for $\mathcal{E} \subseteq \mathcal{E}'$;
  (f) the converse of the above two conditions, on N.

Fact B.2. *Suppose $\widetilde{V}\ \mathcal{E}\ \widetilde{W}$. Then for any context C, if $C[\widetilde{V}]$ is a value then $C[\widetilde{V}]\ \mathcal{E}^{\widehat{\star}}\ C[\widetilde{W}]$ (i.e., $C[\widetilde{W}]$ is a value too).*

We reuse the notation for contexts in Section A (before Lemma A.2). In the remainder of the section, unless otherwise stated, all contexts are location-free, and all terms and contexts are closed.

Lemma B.3. *Suppose $C[\cdot][\widetilde{V}]$ is an evaluation context, and M is not a value. We have:*

(1) If $M \longrightarrow M'$ then $C[M, \widetilde{V}] \longrightarrow C[M', \widetilde{V}]$;
(2) If $C[M, \widetilde{V}] \longrightarrow P$ then $P = C[M', \widetilde{V}]$ for some M' such that $M \longrightarrow M'$.

We recall the assertions of the two lemmas to prove.

LEMMA B.4 (LEMMAS 5.3 AND 5.4 IN THE MAIN TEXT). *(1)* $\langle s\,;\,V\rangle \approx_{\mathcal{E}} \langle t\,;\,W\rangle$ *implies* $\langle s\,;\,C[V]\rangle \approx_{\mathcal{E}} \langle t\,;\,C[W]\rangle$*, for all* C.

(2) $\langle s\,;\,M\rangle \approx_{\mathcal{E}} \langle t\,;\,N\rangle$ *implies* $\langle s\,;\,C[M]\rangle \approx_{\mathcal{E}} \langle t\,;\,C[N]\rangle$*, if the hole of* C *is in redex position.*

PROOF. Suppose $\mathcal{Y}$ is a bisimulation, and take

$$\begin{aligned}\mathcal{X} \stackrel{\text{def}}{=}\ & \{(\widehat{\mathcal{E}^\star}, s, t, C[M, \widetilde{V}], C[N, \widetilde{W}]) \text{ s.t. } \langle s\,;\,M\rangle\ \mathcal{Y}_{\mathcal{E}}\ \langle t\,;\,N\rangle, \text{ and } [\cdot]_1 \text{ is in redex position in } C, \text{ and } V_i\ \mathcal{E}\ W_i\}\\ & \cup(\widehat{\mathcal{E}^\star}, s, t, C[\widetilde{V}], C[\widetilde{W}]) \text{ s.t. } (\mathcal{E}, s, t) \in \mathcal{Y} \text{ and } V_i\ \mathcal{E}\ W_i\}\\ & \cup\{(\widehat{\mathcal{E}^\star}, s, t) \text{ s.t. } (\mathcal{E}, s, t) \in \mathcal{Y}\}\end{aligned}$$

We show that this is a bisimulation up to environment. First, we introduce some notations for the proof.

We say that $\langle s\,;\,P\rangle \mathcal{X} \langle t\,;\,Q\rangle$ *with base* $\mathcal{E}$ if

—either $P = C[M, \widetilde{V}]$, $Q = C[N, \widetilde{W}]$, and $(\widehat{\mathcal{E}^\star}, s, t, C[M, \widetilde{V}], C[N, \widetilde{W}]) \in \mathcal{X}$, because $\langle s\,;\,M\rangle\ \mathcal{Y}_{\mathcal{E}}\ \langle t\,;\,N\rangle$, and $[\cdot]_1$ is in redex position in C and $V_i\ \mathcal{E}\ W_i$,

—or $P = C[\widetilde{V}]$, $Q = C[\widetilde{W}]$, and $(\widehat{\mathcal{E}^\star}, s, t, C[\widetilde{V}], C[\widetilde{W}]) \in \mathcal{X}$ because $(\mathcal{E}, s, t) \in \mathcal{Y}$ and $V_i\ \mathcal{E}\ W_i$.

FACT B.5. *Suppose* $\langle s\,;\,P\rangle \mathcal{X} \langle t\,;\,Q\rangle$ *with base* $\mathcal{E}'$*, and* $\mathcal{E} \subseteq \mathcal{E}'$*. We have:*

—$\langle s\,;\,\#_i P\rangle \mathcal{X} \langle t\,;\,\#_i Q\rangle$ *with base* $\mathcal{E}'$,

—$\langle s\,;\,!P\rangle \mathcal{X} \langle t\,;\,!Q\rangle$ *with base* $\mathcal{E}'$,

—*if* $P'\ \mathcal{E}^\star\ Q'$*, then* $\langle s\,;\,PP'\rangle \mathcal{X} \langle t\,;\,QQ'\rangle$ *with base* $\mathcal{E}'$,

—*if* $V\ \mathcal{E}^\star\ W$ *then also* $\langle s\,;\,VP\rangle \mathcal{X} \langle t\,;\,WQ\rangle$ *with base* $\mathcal{E}'$,

—*if* $\widetilde{P}\ \mathcal{E}^\star\ \widetilde{Q}$*, then* $\langle s\,;\,\mathrm{op}(\widetilde{c}, P, \widetilde{P})\rangle \mathcal{X} \langle t\,;\,\mathrm{op}(\widetilde{c}, Q, \widetilde{Q})\rangle$ *with base* $\mathcal{E}'$,

—*if* $P_i\ \mathcal{E}^\star\ Q_i$*, then* $\langle s\,;\,\mathtt{if}\,P\,\mathtt{then}\,P_1\,\mathtt{else}\,P_2\rangle \mathcal{X} \langle t\,;\,\mathtt{if}\,Q\,\mathtt{then}\,Q_1\,\mathtt{else}\,Q_2\rangle$ *with base* $\mathcal{E}'$,

—*if* $\widetilde{P}\ \mathcal{E}^\star\ \widetilde{Q}$ *and* $\widetilde{V}\ \mathcal{E}^\star\ \widetilde{W}$*, then* $\langle s\,;\,(\widetilde{V}, P, \widetilde{P})\rangle \mathcal{X} \langle t\,;\,(\widetilde{W}, Q, \widetilde{Q})\rangle$ *with base* $\mathcal{E}'$,

—*if* $P'\ \mathcal{E}^\star\ Q'$*, then* $\langle s\,;\,P := P'\rangle \mathcal{X} \langle t\,;\,Q := Q'\rangle$ *with base* $\mathcal{E}'$,

—*if* $(l, l') \in \mathcal{E}$ *then* $\langle s\,;\,l := P\rangle \mathcal{X} \langle t\,;\,l' := Q\rangle$ *with base* $\mathcal{E}'$.

FACT B.6. *Suppose* $\lambda x.P\ \widehat{\mathcal{E}^\star}\ \lambda x.Q$*, and* $V\ \widehat{\mathcal{E}^\star}\ W$*, and* $(\mathcal{E}, s, t) \in \mathcal{Y}$*, where* $\mathcal{Y}$ *is the bisimulation in the definition of* $\mathcal{X}$*. Then* $\langle s\,;\,P\{^V/_x\}\rangle \mathcal{X} \langle t\,;\,Q\{^W/_x\}\rangle$ *with base* $\mathcal{E}$.

PROOF. We distinguish two cases: $\lambda x.P\ \mathcal{E}\ \lambda x.Q$, and the case $P = D[\widetilde{V}']$ and $Q = D[\widetilde{W}']$ for $\widetilde{V}'\ \mathcal{E}\ \widetilde{W}'$. In the first case, since $\mathcal{Y}$ is a bisimulation, we get $(\mathcal{E}, s, t, P\{^V/_x\}, Q\{^W/_x\}) \in \mathcal{Y}$, which shows that $\langle s\,;\,P\{^V/_x\}\rangle \mathcal{X} \langle t\,;\,Q\{^W/_x\}\rangle$ with base $\mathcal{E}$. In the second case, we have $\langle s\,;\,D[\widetilde{V}']\{^V/_x\}\rangle \mathcal{X} \langle t\,;\,D[\widetilde{W}']\{^W/_x\}\rangle$ with base $\mathcal{E}$. □

We are now ready to prove that $\mathcal{X}$ is a bisimulation up to environment. We first consider elements of the form

$$(\widehat{\mathcal{E}^\star}, s, t, C[\widetilde{V}], C[\widetilde{W}]). \tag{9}$$

On these elements, clause (1.b) is immediate: By Fact B.2 if $C[\widetilde{V}]$ is a value, then also $C[\widetilde{W}]$ is a value and they are in $\mathcal{E}^{\widehat{\star}}$. This is sufficient, because $(\mathcal{E}^{\widehat{\star}}, s, t) \in \mathcal{X}$.

Consider now clause (1.a). We proceed by induction on C. First, the possible transitions for when the context C is of the form C_1C_2.

—Suppose $\langle s\,;\, C_1[\widetilde{V}]\rangle \longrightarrow \langle s'\,;\, P\rangle$. Then by induction $\langle t\,;\, C_1[\widetilde{W}]\rangle \Longrightarrow \langle t'\,;\, Q\rangle$ with $\langle s'\,;\, P\rangle \mathcal{X} \langle t'\,;\, Q\rangle$ with base $\mathcal{E}'$ for some $\mathcal{E}'$ with $\mathcal{E} \subseteq \mathcal{E}'$. Applying Fact B.5, also $\langle s'\,;\, PC_2[\widetilde{V}]\rangle \mathcal{X} \langle t'\,;\, QC_2[\widetilde{W}]\rangle$ with base $\mathcal{E}'$.

—Suppose $\langle s\,;\, C_2[\widetilde{V}]\rangle \longrightarrow \langle s'\,;\, P\rangle$ and $C_1[\widetilde{V}]$ is a value. By Fact B.2, $C_1[\widetilde{W}]$ is also a value.
By induction $\langle t\,;\, C_2[\widetilde{W}]\rangle \Longrightarrow \langle t'\,;\, Q\rangle$ with $\langle s'\,;\, P\rangle \mathcal{X} \langle t'\,;\, Q\rangle$ with base $\mathcal{E}'$ for some $\mathcal{E}'$ with $\mathcal{E} \subseteq \mathcal{E}'$. Applying Fact B.5, also $\langle s'\,;\, C_1[\widetilde{V}]P\rangle \mathcal{X} \langle t'\,;\, C_1[\widetilde{W}]Q\rangle$ with base $\mathcal{E}'$.

—Suppose $C_1[\widetilde{V}]$ and $C_2[\widetilde{V}]$ are values, say $\lambda x.\, P$ and V, and $\langle s\,;\, C_1[\widetilde{V}]C_2[\widetilde{V}]\rangle \longrightarrow \langle s\,;\, P\{^V/_x\}\rangle$. Then also $C_1[\widetilde{W}]$ and $C_2[\widetilde{W}]$ are values (Fact B.2), say $\lambda x.\, Q$ and W, and $\langle t\,;\, C_1[\widetilde{W}]C_2[\widetilde{W}]\rangle \longrightarrow \langle t\,;\, Q\{^W/_x\}\rangle$. We can now conclude using Fact B.6.

Now the remaining possibilities for the context.

—$C = (C_1, \ldots, C_n)$. Suppose $\langle s\,;\, C[\widetilde{V}]\rangle \longrightarrow \langle s'\,;\, (C_1[\widetilde{V}], \ldots, P, \ldots, C_n[\widetilde{V}])\rangle$ because $\langle s\,;\, C_i[\widetilde{V}]\rangle \longrightarrow \langle s'\,;\, P\rangle$. This means that $C_1[\widetilde{V}], \ldots, C_{i-1}[\widetilde{V}]$ are values. Hence also $C_1[\widetilde{W}], \ldots, C_{i-1}[\widetilde{W}]$ are values (Fact B.2). Moreover, by induction, $\langle t\,;\, C_i[\widetilde{W}]\rangle \Longrightarrow \langle t'\,;\, Q\rangle$ with $\langle s'\,;\, P\rangle \mathcal{X} \langle t'\,;\, Q\rangle$) with base $\mathcal{E}'$ for some $\mathcal{E}'$ with $\mathcal{E} \subseteq \mathcal{E}'$. We also have $\langle t\,;\, C[W]\rangle \Longrightarrow \langle t'\,;\, (C_1[\widetilde{W}], \ldots, Q, \ldots, C_n[\widetilde{W}])\rangle$. Applying Fact B.5,

$$\langle s'\,;\, (C_1[\widetilde{V}], \ldots, P, \ldots, C_n[\widetilde{V}])\rangle \mathcal{X} \langle t'\,;\, (C_1[\widetilde{W}], \ldots, Q, \ldots, C_n[\widetilde{W}])\rangle$$

with base $\mathcal{E}'$.

—$C = \mathtt{op}(\widetilde{C})$. Similar to the previous case.

—$C = \mathtt{if\ \ then}\ C_1\ \mathtt{else}\ C_2\, C_3$. There are 3 possibilities of reduction, corresponding to the 3 rules in the operational semantics of the `if-then-else` construct.
The case when $\langle s\,;\, C_1[\widetilde{V}]\rangle \longrightarrow \langle s'\,;\, P\rangle$ can be dealt with as previous cases. In the case when $C_1[\widetilde{V}] = c$, for $c \in \{\mathsf{true}, \mathsf{false}\}$ it must also be $C_1[\widetilde{W}] = c$, and then the rest is easy.

—$C = \#_i C_1$. The case when $C_1 = [\cdot]_j$, for some j, is easy (using the fact that $V_j \mathcal{E} W_j$ and $(\mathcal{E}, s, t) \in \mathcal{Y}$). Otherwise, there are 2 possibilities of reduction. The case when $\langle s\,;\, C_1[\widetilde{V}]\rangle \longrightarrow \langle s'\,;\, P\rangle$ can be dealt with as previous cases.
Now the case when $C_1 = (C_1', \ldots, C_n')$. We have $\langle s\,;\, \#_i(C_1'[\widetilde{V}], \ldots, C_n'[\widetilde{V}])\rangle \longrightarrow \langle s\,;\, C_i'[\widetilde{V}]\rangle$. Term $C[\widetilde{W}]$ has the same shape (Fact B.2), we have

$$\langle t\,;\, \#_i(C_1'[\widetilde{W}], \ldots, C_n'[\widetilde{W}])\rangle \longrightarrow \langle t\,;\, C_i'[\widetilde{W}]\rangle,$$

and we are done.

—$C = !C_1$. Here: either $C_1 = [\cdot]_i$, for some i and $V_i = l$, for some l; or $C_1[\widetilde{V}]$ is not a value. In the first case, it must be $W_i = l'$, for some l' with $(s(l), t(l')) \in \mathcal{E}'$, for $\mathcal{E}' = \mathcal{E} \cup \{(l, l')\}$ with $\mathcal{E}' \in \mathcal{Y}$.
We therefore have $\langle s\,;\, !l\rangle \longrightarrow \langle s\,;\, s(l)\rangle$ and $\langle t\,;\, !l'\rangle \longrightarrow \langle t\,;\, t(l')\rangle$, and we are done, as $(\mathcal{E}'^{\widehat{\star}}, s, t) \in \mathcal{X}$. (Note that here we make use of the up to environment,

for we appeal to $\mathcal{E}'$, which is larger than $\mathcal{E}$.)

The second case can be dealt with as previous cases.

—C is $C_1 := C_2$.

We have 3 possibilities of reduction for $C[\widetilde{V}]$, corresponding to the cases when $C_1[\widetilde{V}]$ reduces, $C_2[\widetilde{V}]$ reduces, and when the assignment is made. We consider the third one only. In this case, $C_1[\widetilde{V}] = l$, $C_2[\widetilde{V}]$ is a value, and we have

$$\langle s\,;\, l := C_2[\widetilde{V}]\rangle \longrightarrow \langle s[l = C_2[\widetilde{V}]]\,;\, \star\rangle.$$

Since contexts cannot contain (free) locations, we have $C_1[\widetilde{V}] = V_i = l$, for some i. By definition of bisimulation, it must be $C_1[\widetilde{W}] = W_i = l'$ for some l'. Further, we have $(C_2[\widetilde{V}], C_2[\widetilde{W}]) \in \mathcal{E}^{\widehat{\star}}$ and

$$\langle t\,;\, l' := C_2[\widetilde{W}]\rangle \longrightarrow \langle t[l' = C_2[\widetilde{W}]]\,;\, \star\rangle.$$

Since $(\mathcal{E}, s, t) \in \mathcal{Y}$ and $(l, l') \in \mathcal{E}$, by definition of bisimulation we have $(\mathcal{E}, s[l \leftarrow C_2[\widetilde{V}]], t[l' \leftarrow C_2[\widetilde{W}]]) \in \mathcal{Y}$. Hence also $(\mathcal{E}^{\widehat{\star}}, s[l = C_2[\widetilde{V}]], t[l' = C_2[\widetilde{W}]], \star) \in \mathcal{X}$, and we are done.

—$C = \nu x\, C_1$. We can pick l fresh. Let $|\ \widetilde{V}\ | = n$ (that is, C has at most holes $[\cdot]_1, \ldots, [\cdot]_n$). Define C_2 as the context obtained from C_1 by replacing all occurrences of x with $[\cdot]_{n+1}$. We have

$$\langle s\,;\, \nu x\, C_1[\widetilde{V}]\rangle \longrightarrow \langle s[l = 0]\,;\, C_2[\widetilde{V}][l]_{n+1}\rangle$$

(since $C_2[\widetilde{V}][l]_{n+1} = C_1[\widetilde{V}]\{{}^{l}/_{x}\}$) and, similarly,

$$\langle t\,;\, \nu x\, C_1[\widetilde{W}]\rangle \longrightarrow \langle t[l = 0]\,;\, C_2[\widetilde{W}][l]_{n+1}\rangle.$$

By clause (2.c) of the definition of bisimulation, we also have

$$(\mathcal{E}', s[l = 0], t[l = 0]) \in \mathcal{Y}$$

for $\mathcal{E}' = \mathcal{E} \cup \{(l, l)\}$. We can therefore conclude that

$$\langle s[l = 0]\,;\, C_2[\widetilde{V}][l]_{n+1}\rangle \mathcal{X} \langle t[l = 0]\,;\, C_2[\widetilde{W}][l]_{n+1}\rangle$$

with base $\mathcal{E}'$ (again, we use the up to environment).

Now elements of the form

$$(\mathcal{E}^{\widehat{\star}}, s, t, C[M, \widetilde{V}], C[N, \widetilde{W}]) \tag{10}$$

Suppose M is a value. In this case, $\langle t\,;\, N\rangle \Longrightarrow \langle t'\,;\, W\rangle$ and $(\mathcal{E}', s, t') \in \mathcal{Y}$ for some W with $\mathcal{E}' = \mathcal{E} \cup \{(M, W)\}$. By Lemma B.3(1), also $\langle t\,;\, C[N, \widetilde{W}]\rangle \Longrightarrow \langle t'\,;\, C[W, \widetilde{W}]\rangle$. Now we have two terms $C[M, \widetilde{V}]$ and $C[W, \widetilde{W}]$ and stores s, t' of the form (9), with base $\mathcal{E}'$. Using this property, and the fact that the bisimulation clauses on elements of the form (9) hold (as we just proved) it is immediate to get that the bisimulation clauses also hold on elements of the form (10) (up to environment).

On the other hand, if M is not a value, then also $C[M, \widetilde{V}]$ is not a value. Therefore $\langle s\,;\, C[M, \widetilde{V}]\rangle \longrightarrow \langle s'\,;\, P\rangle$, for some P, and only clause (1.b) of the definition of bisimulation applies. By Lemma B.3(2), $P = C[M', \widetilde{V}]$ with $\langle s\,;\, M\rangle \longrightarrow \langle s'\,;\, M'\rangle$. By definition of bisimulation, $\langle t\,;\, N\rangle \Longrightarrow \langle t'\,;\, N'\rangle$ and $\langle s'\,;\, M'\rangle\ \mathcal{Y}_{\mathcal{E}}\ \langle t'\,;\, N'\rangle$, and by Lemma B.3(1), $\langle t\,;\, C[N, \widetilde{W}]\rangle \Longrightarrow \langle t'\,;\, C[N', \widetilde{W}]\rangle$. We are done, since

$$(\mathcal{E}^{\widehat{\star}}, s', t', C[M', \widetilde{V}], C[N', \widetilde{W}]) \in \mathcal{X}.$$

Finally, we have to check elements of the form

$$(\widehat{\mathcal{E}^\star}, s, t)$$

We know $(\mathcal{E}, s, t) \in \mathcal{Y}$. First, we note that if $V \widehat{\mathcal{E}^\star} W$ then their outermost construct is the same. This holds because either V, W have a common context, or because they are in $\mathcal{E}$ and then it follows from the definition of bisimulation. Therefore there are the following 5 cases to consider, one for each case in clause (2) of the definition of bisimulation.

—$V = \lambda x.\, P$ and $W = \lambda x.\, Q$, and take $V' \; \widehat{\mathcal{E}^\star} \; W'$. We conclude for $P\{^{V'}/_x\}$ and $Q\{^{W'}/_x\}$ using Fact B.6.

—$V = (\widetilde{V})$ and $W = (\widetilde{W})$. If $V \mathcal{E} W$ then, by definition of bisimulation, for all i, we have $V_i \mathcal{E} W_i$, and we are done.
Otherwise, $V = (C_1[\widetilde{V}], \ldots, C_n[\widetilde{V}])$ and $W = (C_1[\widetilde{W}], \ldots, C_n[\widetilde{W}])$ and for all i we have $C_i[\widetilde{V}] \widehat{\mathcal{E}^\star} C_i[\widetilde{W}]$ and we are done.

—Constants. If $V = c$ then also $W = c$.

—If $V = l$ then $W = l'$, for some l' with $(l, l') \in \mathcal{E}$. We have to check 2 properties:
  —$(s(l), t(l')) \in \mathcal{E}'^{\widehat{\star}}$ for some extension $\mathcal{E}'$ of $\mathcal{E}$ with $(\mathcal{E}', s, t) \in \mathcal{Y}$. This holds because $\mathcal{Y}$ is a bisimulation and $(\mathcal{E}, s, t) \in \mathcal{Y}$ (once more, we make use of the up to environment).
  —for all $(V_1, W_1) \in \mathcal{E}^\star$, it holds that $(\widehat{\mathcal{E}^\star}, s[l = V_1], t[l' = W_1]) \in \mathcal{X}$. Again, this holds because $\mathcal{Y}$ is a bisimulation.

—Store extension. It can be dealt with in a way similar to the other cases, exploiting the fact that $\mathcal{Y}$ is a bisimulation and $(\mathcal{E}, s, t) \in \mathcal{Y}$. □

C. PROOF OF LEMMA 5.13

In this section, we prove the soundness of the "up to environment, reduction, and context" technique, in the imperative λ-calculus, stated in Lemma 5.13. In the proof, we make use of a (very) simple instance of it, namely the "up to environment" technique of Definition B.1. The structure of the proof is similar to that of Lemma B.4; some important details are however different; for this reason, and because the proof is fairly delicate, we report all details.

We recall the statement of the lemma: If $\mathcal{Y}$ is a bisimulation up to environment, reduction and context, then $\mathcal{Y} \subseteq \approx$.

PROOF. Suppose $\mathcal{Y}$ is a bisimulation up to environment, reduction, and context, and take

$$\begin{aligned}\mathcal{X} \stackrel{\text{def}}{=} \; & \{(\widehat{\mathcal{E}^\star}, s, t, M, N) \text{ s.t. } \langle s\,;\, M\rangle \; \mathcal{Y}^{\rightarrow\star}_{\mathcal{E}} \; \langle t\,;\, N\rangle \\ & \cup \{(\widehat{\mathcal{E}^\star}, s, t) \text{ s.t. } (\mathcal{E}, s, t) \in \mathcal{Y}\}\end{aligned}$$

We show that this is a bisimulation up to environment. First, some facts that we shall need in this proof.

FACT C.1. *Suppose $\langle s\,;\, P\rangle \; \mathcal{X}^{\rightarrow\star}_{\mathcal{E}'} \; \langle t\,;\, Q\rangle$, and $\mathcal{E} \subseteq \mathcal{E}'$, $[\cdot]_1$ is in redex position in C, and $\widetilde{V} \; \widehat{\mathcal{E}^\star} \; \widetilde{W}$. Then also $\langle s\,;\, C[P, \widetilde{V}]\rangle \; \mathcal{X}^{\rightarrow\star}_{\mathcal{E}'} \; \langle t\,;\, C[Q, \widetilde{W}]\rangle$.*
Fact C.1 implies, for instance, that if $\langle s\,;\, P\rangle \; \mathcal{X}^{\rightarrow\star}_{\mathcal{E}'} \; \langle t\,;\, Q\rangle$, $\mathcal{E} \subseteq \mathcal{E}'$, and $P' \; \mathcal{E}^\star \; Q'$, then $\langle s\,;\, PP'\rangle \; \mathcal{X}^{\rightarrow\star}_{\mathcal{E}'} \; \langle t\,;\, QQ'\rangle$ and $\langle s\,;\, P := P'\rangle \; \mathcal{X}^{\rightarrow\star}_{\mathcal{E}'} \; \langle t\,;\, Q := Q'\rangle$.

FACT C.2. *Suppose $\mathcal{X}$ and $\mathcal{Y}$ are defined as in the proof of the lemma, $\lambda x.P\ \widehat{\mathcal{E}^\star}\ \lambda x.Q$, and $V\ \widehat{\mathcal{E}^\star}\ W$, and $(\mathcal{E},s,t) \in \mathcal{Y}$. Then $\langle s\,;\,P\{^V/_x\}\rangle\ \mathcal{X}^{\rightarrow\star}_{\mathcal{E}'}\ \langle t\,;\,Q\{^W/_x\}\rangle$ for some $\mathcal{E}'$ with $\mathcal{E} \subseteq \mathcal{E}'$.*

PROOF. We distinguish two cases: $\lambda x.P\ \mathcal{E}\ \lambda x.Q$, and the case $P = D[\widetilde{V'}]$ and $Q = D[\widetilde{W'}]$ for $\widetilde{V'}\ \mathcal{E}\ \widetilde{W'}$. In the first case, since $\mathcal{Y}$ is a bisimulation up to reduction and context, we get $\langle s\,;\,P\{^V/_x\}\rangle\ \mathcal{Y}^{\rightarrow\star}_{\mathcal{E}'}\ \langle t\,;\,Q\{^W/_x\}\rangle$ for some $\mathcal{E}'$ with $\mathcal{E} \subseteq \mathcal{E}'$, which shows that $\langle s\,;\,P\{^V/_x\}\rangle\ \mathcal{X}_{\mathcal{E}'}\ \langle t\,;\,Q\{^W/_x\}\rangle$.
In the second case, we have $\langle s\,;\,D[\widetilde{V'}]\{^V/_x\}\rangle\ \mathcal{Y}^\star_{\mathcal{E}}\ \langle t\,;\,D[\widetilde{W'}]\{^W/_x\}\rangle$, hence also $\langle s\,;\,D[\widetilde{V'}]\{^V/_x\}\rangle\ \mathcal{Y}^{\rightarrow\star}_{\mathcal{E}}\ \langle t\,;\,D[\widetilde{W'}]\{^W/_x\}\rangle$, and the conclusion is as before. □

Now, the main proof that $\mathcal{X}$ is a bisimulation up to environment.

Consider an element $(\mathcal{E},s,t,M,N)$ such that there are s',M',t',N' with $\langle s\,;\,M\rangle \Longrightarrow \langle s'\,;\,M'\rangle$, $\langle t\,;\,N\rangle \Longrightarrow \langle t'\,;\,N'\rangle$, and with either

$$M' = C[M''\widetilde{V}], N' = C[N''\widetilde{W}], \tag{11}$$

and $[\cdot]_1$ is in redex position in C, and $V_i\ \mathcal{E}\ W_i$, and $\langle s'\,;\,M''\rangle\ \mathcal{Y}_{\mathcal{E}}\ \langle t'\,;\,N''\rangle$; or

$$M' = C[\widetilde{V}], N' = C[\widetilde{W}], \tag{12}$$

and $V_i\ \mathcal{E}\ W_i$ and $(\mathcal{E},s',t') \in \mathcal{Y}$.

First, in case the reduction $\langle s\,;\,M\rangle \Longrightarrow \langle s'\,;\,M'\rangle$ has length greater than 0, we are done, for if $\langle s\,;\,M\rangle \longrightarrow \langle s_1\,;\,M_1\rangle$ then also $\langle s_1\,;\,M_1\rangle \Longrightarrow \langle s'\,;\,M'\rangle$ and we have $(\mathcal{E},s_1,t,M_1,N) \in \mathcal{X}$.

We suppose therefore below that the reduction has length 0 and thus $\langle s\,;\,M\rangle = \langle s'\,;\,M'\rangle$. We consider first the situation in (12). In this situation, clause (1.b) of bisimulation is immediate: By Fact B.2 if $C[\widetilde{V}]$ is a value, then also $C[\widetilde{W}]$ is a value and they are in $\widehat{\mathcal{E}^\star}$. This is sufficient, because $(\widehat{\mathcal{E}^\star},s,t) \in \mathcal{X}$.

Consider now clause (1.a). Recall that $\langle t\,;\,N\rangle \Longrightarrow \langle t'\,;\,C[\widetilde{W}]\rangle$. We proceed by induction on C. First, the possible transitions for when the context C is of the form C_1C_2.

—Suppose $\langle s\,;\,C_1[\widetilde{V}]\rangle \longrightarrow \langle s_1\,;\,P\rangle$. Then by induction $\langle t'\,;\,C_1[\widetilde{W}]\rangle \Longrightarrow \langle t''\,;\,Q\rangle$ with $\langle s_1\,;\,P\rangle\ \mathcal{X}^{\rightarrow\star}_{\mathcal{E}'}\ \langle t''\,;\,Q\rangle$ for some $\mathcal{E}'$ with $\mathcal{E} \subseteq \mathcal{E}'$. Applying Fact C.1, also

$$\langle s_1\,;\,PC_2[\widetilde{V}]\rangle\ \mathcal{X}^{\rightarrow\star}_{\mathcal{E}'}\ \langle t''\,;\,QC_2[\widetilde{W}]\rangle.$$

—Suppose $\langle s\,;\,C_2[\widetilde{V}]\rangle \longrightarrow \langle s_1\,;\,P\rangle$. This means that $C_1[\widetilde{V}]$ is a value; hence also $C_1[\widetilde{W}]$ is a value (Fact B.2). By induction $\langle t'\,;\,C_2[\widetilde{W}]\rangle \Longrightarrow \langle t''\,;\,Q\rangle$ with $\langle s_1\,;\,P\rangle\ \mathcal{X}^{\rightarrow\star}_{\mathcal{E}'}\ \langle t''\,;\,Q\rangle$ for some $\mathcal{E}'$ with $\mathcal{E} \subseteq \mathcal{E}'$. Applying Fact C.1, also $\langle s_1\,;\,C_1[\widetilde{V}]P\rangle\ \mathcal{X}^{\rightarrow\star}_{\mathcal{E}'}\ \langle t''\,;\,C_1[\widetilde{W}]Q\rangle$.

—Suppose $C_1[\widetilde{V}]$ and $C_2[\widetilde{V}]$ are values, say $\lambda x.P$ and V, and $\langle s\,;\,C_1[\widetilde{V}]C_2[\widetilde{V}]\rangle \longrightarrow \langle s\,;\,P\{^V/_x\}\rangle$. Then also $C_1[\widetilde{W}]$ and $C_2[\widetilde{W}]$ are values, say $\lambda x.Q$ and W, and

$$\langle t'\,;\,C_1[\widetilde{W}]C_2[\widetilde{W}]\rangle \longrightarrow \langle t'\,;\,Q\{^W/_x\}\rangle.$$

We can now conclude using Fact C.2.

Now the remaining possibilities for the context.

—$C = (C_1, \ldots, C_n)$. Suppose $\langle s\,;\, C[\widetilde{V}]\rangle \longrightarrow \langle s_1\,;\, (C_1[\widetilde{V}], \ldots, P, \ldots, C_n[\widetilde{V}])\rangle$ because $\langle s\,;\, C_i[\widetilde{V}]\rangle \longrightarrow \langle s_1\,;\, P\rangle$. This means that $C_1[\widetilde{V}], \ldots, C_{i-1}[\widetilde{V}]$ are values. Hence also $C_1[\widetilde{W}], \ldots, C_{i-1}[\widetilde{W}]$ are values (Fact B.2). Moreover, by induction, $\langle t'\,;\, C_i[\widetilde{W}]\rangle \Longrightarrow \langle t''\,;\, Q\rangle$ with $\langle s_1\,;\, P\rangle\ \mathcal{X}^{\rightarrow\star}_{\mathcal{E}'}\ \langle t''\,;\, Q\rangle)$ for some $\mathcal{E}'$ with $\mathcal{E} \subseteq \mathcal{E}'$. We also have

$$\langle t'\,;\, C[W]\rangle \Longrightarrow \langle t''\,;\, (C_1[\widetilde{W}], \ldots, Q, \ldots, C_n[\widetilde{W}])\rangle.$$

Applying Fact C.1, also

$$\langle s_1\,;\, (C_1[\widetilde{V}], \ldots, P, \ldots, C_n[\widetilde{V}])\rangle\ \mathcal{X}^{\rightarrow\star}_{\mathcal{E}'}\ \langle t''\,;\, (C_1[\widetilde{W}], \ldots, Q, \ldots, C_n[\widetilde{W}])\rangle.$$

—$C = \mathtt{op}(\widetilde{C})$. Similar to the previous one.

—$C = \mathtt{if}\, C_1\, \mathtt{then}\, C_2\, \mathtt{else}\, C_3$. There are 3 possibilities of reduction, corresponding to the 3 rules in the operational semantics of the `if-then-else` construct. The case when $\langle s\,;\, C_1[\widetilde{V}]\rangle \longrightarrow \langle s_1\,;\, P\rangle$ can be dealt with as previous cases. In the case when $C_1[\widetilde{V}] = c$, for $c \in \{\mathrm{true}, \mathrm{false}\}$ it must also be $C_1[\widetilde{W}] = c$, and then the rest is easy.

—$C = \#_i C_1$. The case when $C_1 = [\cdot]_j$, for some j, is easy (using the fact that $V_j \mathcal{E} W_j$ and $(\mathcal{E}, s, t') \in \mathcal{Y}$). Otherwise, there are 2 possibilities of reduction. The case when $\langle s\,;\, C_1[\widetilde{V}]\rangle \longrightarrow \langle s_1\,;\, P\rangle$ can be dealt with as previous cases. Now the case when $C_1 = (C'_1, \ldots, C'_n)$, and $\langle s\,;\, \#_i(C'_1[\widetilde{V}], \ldots, C'_n[\widetilde{V}])\rangle \longrightarrow \langle s\,;\, C'_i[\widetilde{V}]\rangle$. Term $C[\widetilde{W}]$ has the same shape (Fact B.2), and we have

$$\langle t'\,;\, \#_i(C'_1[\widetilde{W}], \ldots, C'_n[\widetilde{W}])\rangle \longrightarrow \langle t'\,;\, C'_i[\widetilde{W}]\rangle,$$

and we are done.

—$C = !C_1$. Here: either $C_1 = [\cdot]_i$, for some i; or $C_1[\widetilde{V}]$ is not a value. In the first case, the interesting case (the only that does not give deadlock) is when $V_i = l$, for some l. It must be $W_i = l'$, for some l' with $(s(l), t'(l') \in \mathcal{E}'^{\widehat{\star}}$ for an extension $\mathcal{E}'$ with $(\mathcal{E}', s, t') \in \mathcal{Y}$. We have $\langle s\,;\, !l\rangle \longrightarrow \langle s\,;\, s(l)\rangle$ and $\langle t'\,;\, !l'\rangle \longrightarrow \langle t'\,;\, t'(l')\rangle$, and we are done, since $(s(l), t'(l') \in \mathcal{E}'^{\widehat{\star}}$.
The second case can be dealt with as previous cases.

—C is $C_1 := C_2$.
We have 3 possibilities of reduction for $C[\widetilde{V}]$, corresponding to the cases when $C_1[\widetilde{V}]$ reduces, $C_2[\widetilde{V}]$ reduces, and when the assignment is made. We consider the third one only. In this case, $C_1[\widetilde{V}] = l$, $C_2[\widetilde{V}]$ is a value, and we have

$$\langle s\,;\, l := C_2[\widetilde{V}]\rangle \longrightarrow \langle s[l = C_2[\widetilde{V}]]\,;\, \star\rangle.$$

Since contexts cannot contain locations, we have $C_1[\widetilde{V}] = V_i = l$, for some i. By definition of bisimulation, it must be $C_1[\widetilde{W}] = W_i = l'$ for some l'. Further, we have $(C_2[\widetilde{V}], C_2[\widetilde{W}]) \in \mathcal{E}^{\widehat{\star}}$ and

$$\langle t'\,;\, l' := C_2[\widetilde{W}]\rangle \longrightarrow \langle t'[l' = C_2[\widetilde{W}]]\,;\, \star\rangle.$$

Since $(l, l') \in \mathcal{E}$ and $(\mathcal{E}, s, t') \in \mathcal{Y}$, by definition of $\mathcal{Y}$ we get $(\mathcal{E}', s[l = C_2[\widetilde{V}]], t'[l' = C_2[\widetilde{W}]]) \in \mathcal{Y}$, for $\mathcal{E} \subseteq \mathcal{E}'$. Hence also $(\mathcal{E}'^{\widehat{\star}}, s[l = C_2[\widetilde{V}]], t'[l' = C_2[\widetilde{W}]]) \in \mathcal{X}$, and we are done.

—$C = \nu x\, C_1$. We can pick l fresh. Let $|\ \widetilde{V}\ | = n$ (that is, C has at most holes $[\cdot]_1, \ldots, [\cdot]_n$). Define C_2 as the context obtained from C_1 by replacing all

occurrences of x with $[\cdot]_{n+1}$. We have

$$\langle s\,;\, \nu x\, C_1[\widetilde{V}]\rangle \longrightarrow \langle s[l=0]\,;\, C_2[\widetilde{V}][l]_{n+1}\rangle$$

(since $C_2[\widetilde{V}][l]_{n+1} = C_1[\widetilde{V}]\{^l/_x\}$) and, similarly,

$$\langle t'\,;\, \nu x\, C_1[\widetilde{W}]\rangle \longrightarrow \langle t'[l=0]\,;\, C_2[\widetilde{W}][l]_{n+1}\rangle.$$

By clause (2.c) of the definition of bisimulation up to environment, reduction, and context, from $(\mathcal{E}, s, t') \in \mathcal{Y}$ we get

$$(\mathcal{E}', s[l \leftarrow 0], t'[l \leftarrow 0]) \in \mathcal{Y}$$

for some extension $\mathcal{E}'$ of $\mathcal{E}$ with $(l, l) \in \mathcal{E}'$. We can therefore conclude that

$$\langle s[l=0]\,;\, C_2[\widetilde{V}][l]_{n+1}\rangle \;\mathcal{X}_{\mathcal{E}'^{\star}}\; \langle t'[l=0]\,;\, C_2[\widetilde{W}][l]_{n+1}\rangle.$$

Now elements of the form (11). Suppose M'' is a value. In this case, $\langle t'\,;\, N''\rangle \Longrightarrow \langle t''\,;\, W\rangle$ and $(\mathcal{E}', s_1, t'') \in \mathcal{Y}$ for some $\mathcal{E}'$ with $\mathcal{E} \subseteq \mathcal{E}'$ and $M''\ \mathcal{E}'^{\star}\ W$. By Lemma B.3(1), also $\langle t'\,;\, C[N''\widetilde{W}]\rangle \Longrightarrow \langle t''\,;\, C[W\widetilde{W}]\rangle$. Now we have two terms $C[M''\widetilde{V}]$ and $C[W\widetilde{W}]$ and stores s, t'' of the form (12). Using this property, and the fact that the bisimulation clauses on elements of the form (12) hold (as we just proved) it is immediate to get that the bisimulation clauses also hold on elements of the form (11).

On the other hand, if M'' is not a value, then $C[M''\widetilde{V}]$ is not a value either. Therefore $\langle s'\,;\, C[M''\widetilde{V}]\rangle \longrightarrow \langle s''\,;\, P\rangle$, for some P, and only clause (1.b) of the definition of bisimulation applies. By Lemma B.3(2), $P = C[M'''\widetilde{V}]$ with $\langle s\,;\, M''\rangle \longrightarrow \langle s''\,;\, M'''\rangle$. By definition of bisimulation up to environment, reduction, and context, $\langle t'\,;\, N''\rangle \Longrightarrow \langle t''\,;\, N'''\rangle$ and $\langle s''\,;\, M'''\rangle\ \mathcal{Y}^{\rightarrow}_{\mathcal{E}'^{\star}}\ \langle t''\,;\, N'''\rangle$ for some $\mathcal{E}'$ with $\mathcal{E} \subseteq \mathcal{E}'$, and by Lemma B.3(1), $\langle t'\,;\, C[N''\widetilde{W}]\rangle \Longrightarrow \langle t''\,;\, C[N'''\widetilde{W}]\rangle$. We are done, since we have (by Fact C.1)

$$(\mathcal{E}'^{\star}, s'', t'', C[M'''\widetilde{V}], C[N'''\widetilde{W}]) \in \mathcal{X}.$$

Finally, we have to check elements of the form

$$(\mathcal{E}^{\star}, s, t)$$

(this final part of the proof is essentially the same as the corresponding part in the proof for Lemma B.4, though it does not appear to be possible to merge them). We know $(\mathcal{E}, s, t) \in \mathcal{Y}$. First, we note that if $V\,\mathcal{E}^{\star}\,W$ then their outermost construct is the same. This holds because either V, W have a common context, or because they are in $\mathcal{E}$ and then it follows from the definition of bisimulation. Therefore there are the following 5 cases to consider, one for each case in clause (2) of the definition of bisimulation.

—$V = \lambda x.\, P$ and $W = \lambda x.\, Q$, and take $V'\ \mathcal{E}^{\star}\ W'$. We conclude for $P\{^{V'}/_x\}$ and $Q\{^{W'}/_x\}$ using Fact C.2.

—$V = (\widetilde{V})$ and $W = (\widetilde{W})$. If $V\,\mathcal{E}\,W$ then for all i, we have $V_i\,\mathcal{E}'\,W_i$ for an extension $\mathcal{E}'$ of $\mathcal{E}$, and we are done.
Otherwise, $V = (C_1[\widetilde{V}], \ldots, C_n[\widetilde{V}])$ and $W = (C_1[\widetilde{W}], \ldots, C_n[\widetilde{W}])$ and for all i we have $C_i[\widetilde{V}]\,\mathcal{E}^{\star}\,C_i[\widetilde{W}]$ and we are done.

—Constants. If $V = c$ then also $W = c$.

—If $V = l$ then $W = l'$, for some l' with $(l, l') \in \mathcal{E}$. We have to check 2 properties:

—$(s(l), t(l')) \in \mathcal{E}'^{\widehat{\star}}$ for some extension $\mathcal{E}'$ of $\mathcal{E}$ with $(\mathcal{E}', s, t) \in \mathcal{Y}$. This holds because $(\mathcal{E}, s, t) \in \mathcal{Y}$ and $\mathcal{Y}$ is a bisimulation up to environment, reduction and context.
—for all $(V_1, W_1) \in \mathcal{E}^{\widehat{\star}}$, it holds that $(\mathcal{E}'^{\widehat{\star}}, s[l = V_1], t[l' = W_1]) \in \mathcal{X}$, for some extension $\mathcal{E}'$ of $\mathcal{E}$. Again, this holds because $\mathcal{Y}$ is a bisimulation up to environment, reduction and context.

—Store extension. It can be dealt with in a way similar to the other cases, exploiting the fact that $\mathcal{Y}$ is a bisimulation up to environment, reduction, and context, and $(\mathcal{E}, s, t) \in \mathcal{Y}$.

□

D. PROOF OF THEOREM 6.18

We show here the details of Theorem 6.18: the congruence proof of bisimulation for parallel composition.

PROOF OF THEOREM 6.18. Suppose $\mathcal{Y}$ is a bisimulation. Define $\mathcal{X}$ as the set of all tuples

$$(r\,;\,(r\,;\,\mathcal{E})^\star\,;\,P \mid G\{\widetilde{M}/\widetilde{x}\}\,;\,Q \mid G\{\widetilde{N}/\widetilde{x}\})$$

such that

—$(r\,;\,\mathcal{E}\,;\,P\,;\,Q) \in \mathcal{Y}$ and $(G\{\widetilde{M}/\widetilde{x}\}, G\{\widetilde{N}/\widetilde{x}\}) \in (r\,;\,\mathcal{E})^\star$ with $(\widetilde{M}, \widetilde{N}) \in \mathcal{E}$ (and all related side conditions on the names of the common term G, that is, $\mathsf{fn}(G) \subseteq r$, $\mathsf{fv}(G) \subseteq X$ and the $\mathsf{bn}(G)$ are fresh);

—G is guarded.

We prove that this is a bisimulation up-to $=_{\mathrm{str}}$, restriction, and environment. We use a case analysis on the possible forms of action. We sometimes omit the free name component (i.e., r) when clear (because it stays unchanged).

First we consider the case of an interaction between P and $G\{\widetilde{M}/\widetilde{x}\}$, and write F for the derivative.

—$P \xrightarrow{(\nu\widetilde{b})\overline{a}M_0} P'$ and $G\{\widetilde{M}/\widetilde{x}\} \xrightarrow{a(x_0)} R$ and F is $\nu\widetilde{b}\,(P' \mid R\{M_0/x_0\})$.

Since $\mathcal{Y}$ is a bisimulation, $Q \overset{(\nu\widetilde{c})\overline{a}N_0}{\Longrightarrow} Q'$ and

$$P'\;\mathcal{Y}_{\mathcal{E},(M_0,N_0);r}\;Q'\,. \tag{13}$$

Also, by Lemma 6.5, there is G' such that $R = G'\{\widetilde{M}/\widetilde{x}\}$, and $G\{\widetilde{N}/\widetilde{x}\} \xrightarrow{a(x_0)} G'\{\widetilde{N}/\widetilde{x}\}$. Thus we derive:

$$Q \mid G\{\widetilde{N}/\widetilde{x}\} \xrightarrow{\tau} \nu\widetilde{c}\,(Q' \mid G'\{\widetilde{N}/\widetilde{x}\}\{N_0/x_0\})\,.$$

We have to prove that

$$\nu\widetilde{b}\,(P' \mid G'\{\widetilde{M}/\widetilde{x}\}\{M_0/x_0\})\;\mathcal{X}_{\mathcal{E};r}\;\nu\widetilde{c}\,(Q' \mid G'\{\widetilde{N}/\widetilde{x}\}\{N_0/x_0\}) \tag{14}$$

up-to restriction, environment and $=_{\mathrm{str}}$. We can assume $\widetilde{b}$ and $\widetilde{c}$ fresh.

Now, applying Lemma 6.6 to G', there is G'' guarded, a multiset J over the indices of the variables $x_0, \widetilde{x}$ of G', such that:

$$G' =_{\mathrm{str}} \prod_{j\in J} x_j \mid G''\,.$$

Furthermore, for

$$P_2 \stackrel{\text{def}}{=} \prod_{j\in J} M_j$$

$$Q_2 \stackrel{\text{def}}{=} \prod_{j\in J} N_j$$

it also holds that

$$G'\{\widetilde{M}/\widetilde{x}\}\{M_0/x_0\} =_{\text{str}} P_2 \mid G''\{\widetilde{M}/\widetilde{x}\}\{M_0/x_0\}$$
$$G'\{\widetilde{N}/\widetilde{x}\}\{N_0/x_0\} =_{\text{str}} Q_2 \mid G''\{\widetilde{N}/\widetilde{x}\}\{N_0/x_0\}.$$

From (13), using clause (4) of bisimulation, we derive

$$P' \mid P_2 \; \mathcal{Y}_{\mathcal{E},(M_0,N_0);r} \; Q' \mid Q_2.$$

We can therefore conclude that

$$P' \mid P_2 \mid G''\{\widetilde{M}/\widetilde{x}\}\{M_0/x_0\} \; \mathcal{X}_{\mathcal{E}';r} \; Q' \mid Q_2 \mid G''\{\widetilde{N}/\widetilde{x}\}\{N_0/x_0\}$$

holds, for $\mathcal{E}' \stackrel{\text{def}}{=} (r\,;\,\mathcal{E},(M_0,N_0))^\star$. This is sufficient for (14) (up-to environment, restriction, and $=_{\text{str}}$).

— $P \xrightarrow{a(x)} P'$ and $G\{\widetilde{M}/\widetilde{x}\} \xrightarrow{(\nu\widetilde{b})\overline{a}P_1} R$ and F is $\nu\widetilde{b}\,(P'\{P_1/x\} \mid R)$.

By Lemma 6.5, there are G', G'' s.t. $R = G'\{\widetilde{M}/\widetilde{x}\}$, $P_1 = G''\{\widetilde{M}/\widetilde{x}\}$, and we also have $G\{\widetilde{N}/\widetilde{x}\} \xrightarrow{(\nu\widetilde{b})\overline{a}Q_1} G'\{\widetilde{N}/\widetilde{x}\}$, for $Q_1 \stackrel{\text{def}}{=} G''\{\widetilde{N}/\widetilde{x}\}$. Since $\mathcal{Y}$ is a bisimulation, we can extend the set of free names to include $\widetilde{b}$:

$$P \; \mathcal{Y}_{\mathcal{E};r,\widetilde{b}} \; Q. \tag{15}$$

Now we have

$$(G''\{\widetilde{M}/\widetilde{x}\}, G''\{\widetilde{N}/\widetilde{x}\}) \in (r, \widetilde{b}\,;\,\mathcal{E})^\star \tag{16}$$

Therefore, since $\mathcal{Y}$ is a bisimulation, from $P \xrightarrow{aP_1} P'\{P_1/x\} \stackrel{\text{def}}{=} P''$, and (15) and (16), we deduce there is Q' s.t. $Q \stackrel{aQ_1}{\Longrightarrow} Q''$ and we have

$$P'' \; \mathcal{Y}_{\mathcal{E};r,\widetilde{b}} \; Q''.$$

We can also infer that

$$Q \mid G\{\widetilde{N}/\widetilde{x}\} \xrightarrow{\tau} \nu\widetilde{b}\,(Q'' \mid G'\{\widetilde{N}/\widetilde{x}\}).$$

We have to prove that

$$\nu\widetilde{b}\,(P'' \mid G'\{\widetilde{M}/\widetilde{x}\}) \; \mathcal{X}_{\mathcal{E};r} \; \nu\widetilde{b}\,(Q'' \mid G'\{\widetilde{N}/\widetilde{x}\}) \tag{17}$$

up-to restriction, $=_{\text{str}}$ and environment. Using Lemma 6.6, we have: $G' =_{\text{str}} \prod_{j\in J} x_j \mid H$ where H is guarded and J is a multiset. Hence

$$\nu\widetilde{b}\,(P'' \mid G'\{\widetilde{M}/\widetilde{x}\}) =_{\text{str}} \nu\widetilde{b}\left(\left(P'' \middle| \prod_{j\in J} M_j\right) \middle| H\{\widetilde{M}/\widetilde{x}\}\right)$$

and similarly

$$\nu\widetilde{b}\,(Q'' \mid G'\{\widetilde{N}/\widetilde{x}\}) =_{\text{str}} \nu\widetilde{b}\left(\left(Q'' \middle| \prod_{j\in J} N_j\right) \middle| H\{\widetilde{N}/\widetilde{x}\}\right)$$

with

$$P''' \stackrel{\text{def}}{=} P'' \Big| \prod_{j\in J} M_j \;\; \mathcal{Y}_{\mathcal{E};r,\tilde{b}} \;\; Q'' \Big| \prod_{j\in J} N_j \stackrel{\text{def}}{=} Q'''.$$

We therefore conclude that

$$P''' \mid H\{\widetilde{M}/\tilde{x}\} \; \mathcal{X}_{\mathcal{E}';r,\tilde{b}} \; Q''' \mid H\{\widetilde{N}/\tilde{x}\}$$

and $\mathcal{E}' \stackrel{\text{def}}{=} (r, \tilde{b}\,;\, \mathcal{E})^\star$. This is sufficient for (17), up-to $=_{\text{str}}$ and environment.

We consider now the actions from $G\{\widetilde{M}/\tilde{x}\}$. By Lemma 6.5, an action from $G\{\widetilde{M}/\tilde{x}\}$ stems from one originating from G.

—τ action: immediate, using Lemma 6.5.

—Input action $G \stackrel{a(x)}{\longrightarrow} G'$. We have thus actions $G\{\widetilde{M}/\tilde{x}\} \stackrel{a(x)}{\longrightarrow} G'\{\widetilde{M}/\tilde{x}\}$ and $G\{\widetilde{N}/\tilde{x}\} \stackrel{a(x)}{\longrightarrow} G'\{\widetilde{M}/\tilde{x}\}$.
Take $(A, B) \in (r\,;\, \mathcal{E})^\star$. We can show that

$$P \mid G'\{\widetilde{M}/\tilde{x}\}\{A/x\} \; \mathcal{X}_{\mathcal{E};r} \; Q \mid G'\{\widetilde{N}/\tilde{x}\}\{B/x\}$$

up to $=_{\text{str}}$. If $(A, B) \in (r\,;\, \mathcal{E})^\star$ then there is a process H and processes $\widetilde{A}, \widetilde{B}$ in $\mathcal{E}$ with $A = H\{\widetilde{A}/\tilde{y}\}$ and $B = H\{\widetilde{B}/\tilde{y}\}$. Further, composing substitutions, we find a process G'' s.t.

$$G'\{\widetilde{M}/\tilde{x}\}\{A/x\} = G''\{\widetilde{M}, \widetilde{A}/\tilde{x}, \tilde{y}\} \qquad G'\{\widetilde{N}/\tilde{x}\}\{B/x\} = G''\{\widetilde{N}, \widetilde{B}/\tilde{x}, \tilde{y}\}$$

We can finally proceed as in previous cases above so to extract from this context G'' a guarded context component and a parallel composition of holes. Then, up-to $=_{\text{str}}$, we close the bisimulation.

—Output action $G \xrightarrow{(\nu\tilde{b})\overline{a}G''} G'$. We have thus actions $G\{\widetilde{M}/\tilde{x}\} \xrightarrow{(\nu\tilde{b})\overline{a}G''\{\widetilde{M}/\tilde{x}\}} G'\{\widetilde{M}/\tilde{x}\}$ and $G\{\widetilde{N}/\tilde{x}\} \xrightarrow{(\nu\tilde{b})\overline{a}G''\{\widetilde{N}/\tilde{x}\}} G'\{\widetilde{N}/\tilde{x}\}$. Further,

$$(G''\{\widetilde{M}/\tilde{x}\}, G''\{\widetilde{N}/\tilde{x}\}) \in (r, \tilde{b}\,;\, \mathcal{E})^\star.$$

We want to show that

$$P \mid G'\{\widetilde{M}/\tilde{x}\} \; \mathcal{X}_{\mathcal{E};r} \; Q \mid G'\{\widetilde{N}/\tilde{x}\} \tag{18}$$

up-to restriction, $=_{\text{str}}$ and environment.
Using clause (5) of bisimulation, we have $P \; \mathcal{Y}_{\mathcal{E};r,\tilde{b}} \; Q$. As in previous cases, using Lemma 6.6 we can isolate a guarded component G'' in G', thus for some J:

$$P \mid G'\{\widetilde{M}/\tilde{x}\} =_{\text{str}} P \Big| \prod_{j\in J} M_j \Big| G''\{\widetilde{M}/\tilde{x}\}$$

$$Q \mid G'\{\widetilde{N}/\tilde{x}\} =_{\text{str}} Q \Big| \prod_{j\in J} N_j \Big| G''\{\widetilde{N}/\tilde{x}\}$$

and (by clause (4) of bisimulation) $P \mid \prod_{j\in J} M_j \; \mathcal{Y}_{\mathcal{E};r,\tilde{b}} \; Q \mid \prod_{j\in J} N_j$. We can thus conclude

$$P \Big| \prod_{j\in J} M_j \Big| G''\{\widetilde{M}/\tilde{x}\} \; \mathcal{X}_{\mathcal{E}';r,\tilde{b}} \; Q \Big| \prod_{j\in J} N_j \Big| G''\{\widetilde{N}/\tilde{x}\}$$

for $\mathcal{E}' \stackrel{\text{def}}{=} (r, \tilde{b}\,;\, \mathcal{E})^\star$. This proves (18), up-to $=_{\text{str}}$, restriction and environment.

Now the actions from P.

—τ-actions and input actions: easy.

—Output actions. We have $P \xrightarrow{(\nu\widetilde{b})\overline{a}P_1} P'$ and $Q \stackrel{(\nu\widetilde{c})\overline{a}Q_1}{\Longrightarrow} Q'$ and we have

$$P' \; \mathcal{Y}_{\mathcal{E},(P_1,Q_1);r} \; Q'.$$

Therefore (up-to environment)

$$P' \mid G\{\widetilde{M}/\widetilde{x}\} \; \mathcal{X}_{\mathcal{E}';r} \; Q' \mid G\{\widetilde{N}/\widetilde{x}\}$$

for $\mathcal{E}' \stackrel{\text{def}}{=} (r\,;\,\mathcal{E},(P_1,\,Q_1))^\star$.

Finally, the bisimulation clauses (4) and (5) are easy. (For clause (4), the reasoning is similar to that of the output actions from $G\{\widetilde{M}/\widetilde{x}\}$ above.) □

E. PROOF OF THEOREM 6.35

We prove here Theorem 6.35, asserting that relations $\equiv_{\text{b}}$ and $\simeq$ coincide. As in the comparisons between bisimulation and contextual equivalence in the λ-calculi, we separate soundness and completeness.

LEMMA E.1 (SOUNDNESS OF $\simeq$). $\simeq \subseteq \equiv_{\text{b}}$.

PROOF. Follows from the fact that processes in the relation $\simeq$ have the same barbs, and $\simeq$ is preserved by all contexts. □

The hard part is completeness of $\simeq$. In the proof of completeness we use a refinement of the standard encoding of replication in HOπ (the first encoding of replication in Section 6, see also [Thomsen 1993; Sangiorgi and Walker 2001]); it is not a general encoding—we only define it for output processes—but, with respect to the standard encoding, it has the advantage that the processes resulting from the encoding are not divergent. For a process P and a channel c, we write

$$\mathtt{rep}^\bullet\, \overline{c}P \stackrel{\text{def}}{=} \nu a\,(K \mid \overline{a}K)$$

where a is fresh for P and c and

$$K \stackrel{\text{def}}{=} a(x).\,\overline{c}P.\,(x \mid \overline{a}x).$$

Finally, we set

$$\mathtt{rep}\; \overline{c}P \stackrel{\text{def}}{=} \overline{c}P.\,\mathtt{rep}^\bullet\, \overline{c}P.$$

Process $\mathtt{rep}\; \overline{c}P$ mimics the replication of $\overline{c}P$; indeed its behaviour is to repeat an output $\overline{c}P$ and a reduction, ad infinitum:

$$\mathtt{rep}\; \overline{c}P \xrightarrow{\overline{c}P} \mathtt{rep}^\bullet\, \overline{c}P \longrightarrow =_{\text{str}} \mathtt{rep}\; \overline{c}P \xrightarrow{\overline{c}P} \cdots$$

We also have:

LEMMA E.2. $\mathtt{rep}\; \overline{c}P \equiv_{\text{b}} \mathtt{rep}^\bullet\, \overline{c}P$.

PROOF. It is easy to prove that $\mathtt{rep}\; \overline{c}P \simeq \mathtt{rep}^\bullet\, \overline{c}P$, as the latter process can only reduce to the former. Then we derive the result from the inclusion $\simeq \subseteq \equiv_{\text{b}}$ (Lemma E.1). □

For the completeness proof, we also make use of internal summation between two processes P and Q, which we write $P \oplus Q$ as an abbreviation for

$$\nu a\,(a(x).\,P \mid a(x).\,Q \mid \overline{a}\mathbf{0})$$

where a and x are fresh. We have:

$$P \oplus Q \longrightarrow =_{\mathrm{str}} P \text{ and } P \oplus Q \longrightarrow =_{\mathrm{str}} Q.$$

Now the proof of completeness. Below, if s is a finite set of names s, then we write $\nu s\ P$ for the restriction of P on all names in s. We also write f, where f is a name, as an abbreviation for $f(x).\,\mathbf{0}$.

LEMMA E.3 (COMPLETENESS OF $\simeq$). $\equiv_{\mathrm{b}} \subseteq \simeq$.

PROOF. Let $\mathcal{X}$ be the relation with all elements $(r; \mathcal{E}; M; N)$ s.t. if $\mathcal{E} = \bigcup_{i\in I}\{(P_i, Q_i)\}$ then there are fresh names $\{e_i\}_{i\in I}$ and $\{f_j\}_{j\in J}$ and names s s.t. for

$$A_1 \stackrel{\mathrm{def}}{=} \Pi_{i\in I}\mathtt{rep}\,\overline{e_i}P_i \mid \Pi_{j\in J} f_j$$
$$B_1 \stackrel{\mathrm{def}}{=} \Pi_{i\in I}\mathtt{rep}\,\overline{e_i}Q_i \mid \Pi_{j\in J} f_j$$

then

$$A \stackrel{\mathrm{def}}{=} \nu s\,(A_1 \mid M) \equiv_{\mathrm{b}} \nu s\,(B_1 \mid N) \stackrel{\mathrm{def}}{=} B$$

and $\mathsf{fn}(\mathcal{E}, M, N) - s \subseteq r$.

We show that $\mathcal{X}$ is a bisimulation up to $=_{\mathrm{str}}$.

(1) Reduction.

Suppose $M \stackrel{\tau}{\to} M'$. Then $A \stackrel{\tau}{\to} \nu s\,(A_1 \mid M') \stackrel{\mathrm{def}}{=} A'$. Hence $B \Longrightarrow B' \equiv_{\mathrm{b}} A'$. The only possibility of reduction for B comes from N, therefore we have $B' = \nu s\,(B_1 \mid N')$ for some N' with $N \Longrightarrow N'$, and we are done.

—Output action, at a name $a \in r$. Suppose $M \xrightarrow{(\nu\widetilde{b})\overline{a}P_1} M'$.

Consider the context

$$C \stackrel{\mathrm{def}}{=} a(x).\,((f_1 \oplus f_2) \mid \mathtt{rep}\,\overline{e}x) \mid [\cdot]$$

where $f_1,\ f_2, e$ are fresh. We have

$$C[A] \Longrightarrow =_{\mathrm{str}} (\nu\widetilde{b}, s)\,(f_2 \mid \mathtt{rep}\,\overline{e}P_1 \mid A_1 \mid M')$$

and the barb at f_1 has been lost, whereas the barb f_2 is visible. Now, $C[B]$ can match the above move, and this can only happen if

$$N \stackrel{(\nu\widetilde{c})\overline{a}Q_1}{\Longrightarrow} N'$$

so that

$$C[B] \Longrightarrow =_{\mathrm{str}} (\nu\widetilde{c}, s)\,(f_2 \mid \mathtt{rep}\,\overline{e}Q_1 \mid B_1 \mid N').$$

Finally, from

$$(\nu\widetilde{b}, s)\,(f_2 \mid \mathtt{rep}\,\overline{e}P_1 \mid A_1 \mid M') \equiv_{\mathrm{b}} (\nu\widetilde{c}, s)\,(f_2 \mid \mathtt{rep}\,\overline{e}Q_1 \mid B_1 \mid N')$$

since $\widetilde{b}$ and $\widetilde{c}$ are fresh, we also have

$$(\nu\widetilde{b}, \widetilde{c}, s)\,(f_2 \mid \mathtt{rep}\,\overline{e}P_1 \mid A_1 \mid M') \equiv_{\mathrm{b}} (\nu\widetilde{b}, \widetilde{c}, s)\,(f_2 \mid \mathtt{rep}\,\overline{e}Q_1 \mid B_1 \mid N')$$

(the restrictions added are harmless, because the restricted names do not appear free in the body). Now we are done, for, up to $=_{\mathrm{str}}$, we have

$$M' \ \mathcal{X}_{\mathcal{E},(P_1,Q_1);r} \ N'$$

—Input action, at a name $a \in r$. Let $(P, Q) \in (r\,;\mathcal{E})^\star$. This means that there is G with free names in r and free variables in $x_1, \ldots x_n$, and pairs of processes $(P_1, Q_1), \ldots, (P_n, Q_n)$ in $\mathcal{E}$ such that $P = G\{P_1, \ldots, P_n/x_1, \ldots, n_n\}$ and also $Q = G\{Q_1, \ldots, Q_n/x_1, \ldots, n_n\}$.

Suppose now $M \stackrel{aP}{\longrightarrow} M'$. Consider the context

$$C \stackrel{\mathrm{def}}{=} e_1(x_1). \ldots e_n(x_n).\,\overline{a}G.\,(f_1 \oplus f_2) \mid [\cdot].$$

We have

$$C[A] \Longrightarrow_{=\mathrm{str}} \nu s\,(f_2 \mid A_1 \mid M') \stackrel{\mathrm{def}}{=} A'$$

and the barb at f_1 has been lost, whereas the barb f_2 is visible. Now, $C[B]$ can only match the above move if

$$C[B] \Longrightarrow_{=\mathrm{str}} \nu s\,(f_2 \mid B_1' \mid N')$$

for some N' s.t.

$$N \stackrel{aQ}{\Longrightarrow} N'$$

and B_1' s.t.

$$B_1 \stackrel{\overline{e_1}P_1}{\Longrightarrow} \ldots \stackrel{\overline{e_n}P_n}{\Longrightarrow} B_1'\,.$$

From the definition of B_1 and the processes $\mathtt{rep}\ \overline{e_i}P_i$ (Lemma E.2) we have $B_1 \equiv_{\mathrm{b}} B_1'$, so that

$$A' \equiv_{\mathrm{b}} \nu s\,(f_2 \mid B_1 \mid N').$$

Now we are done, for, up to $=_{\mathrm{str}}$, we have

$$M' \ \mathcal{X}_{\mathcal{E};r} \ N'.$$

—Now, clause (4) of bisimulation. Let $(P_i, Q_i) \in \mathcal{E}$. We have to show that also $P \mid M \ \mathcal{X}_{\mathcal{E};r} \ Q \mid N$, up to $=_{\mathrm{str}}$. Consider the context

$$C \stackrel{\mathrm{def}}{=} e_i(x).\,(x \mid (f_1 \oplus f_2)) \mid [\cdot]$$

where f_1 and f_2 are fresh. We have

$$C[A] \Longrightarrow_{=\mathrm{str}} \nu s\,(f_2 \mid A_1 \mid P_i \mid M) \stackrel{\mathrm{def}}{=} A'$$

and the barb at f_1 has been lost, whereas the barb f_2 is visible. This move can only be matched by $C[B]$ thus:

$$C[B] \Longrightarrow_{=\mathrm{str}} \nu s\,(f_2 \mid B_1' \mid Q_i \mid N)$$

for some B_1' s.t. $B_1 \stackrel{\overline{e_i}P_i}{\Longrightarrow} B_1'$. Reasoning as in the previous case, we deduce $B_1 \equiv_{\mathrm{b}} B_1'$, hence

$$A' \equiv_{\mathrm{b}} \nu s\,(f_2 \mid B_1 \mid Q_i \mid N)$$

and we are done.

—Clause (5) of the bisimulation is easy.

□

The two lemmas above conclude the proof of Theorem 6.35.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for their careful reading of an earlier draft and for many useful suggestions and remarks.

REFERENCES

ABADI, M. AND GORDON, A. D. 1998. A bisimulation method for cryptographic protocols. In *Proceedings of the European Symposium on Programming (ESOP)*. Lecture Notes in Computer Science, vol. 1381. Springer, 12–26.

ABRAMSKY, S. 1990. The lazy lambda calculus. In *Research Topics in Functional Programming*, D. A. Turner, Ed., Addison-Wesley, 65–117.

AHMED, A. 2006. Step-indexed syntactic logical relations for recursive and quantified types. In *Proceedings of the European Symposium on Programming (ESOP)*. 69–83.

AHMED, A., APPEL, A. W., AND VIRGA, R. 2003. An indexed model of impredicative polymorphism and mutable references. http://www.cs.princeton.edu/~amal/papers/impred.pdf.

AHMED, A., DREYER, D., AND ROSSBERG, A. 2009. State-dependent representation independence. In *Proceedings of the ACM Symposium on Principles of Programming Languages (POPL)*. 340–353.

APPEL, A. W. AND MCALLESTER, D. 2001. An indexed model of recursive types for foundational proof-carrying code. *ACM Trans. Program. Lang. Syst. 23,* 5, 657–683.

BALDAMUS, M. AND FRAUENSTEIN, T. 1995. Congruence proofs for weak bisimulation equivalences on higher–order process calculi. Report 95–21, Computer Science Department, Berlin University of Technology.

BIRKEDAL, L. AND HARPER, R. 1999. Relational interpretations of recursive types in an operational setting. *Inform. Comput. 155,* 1–2, 3–63.

BOREALE, M., DE NICOLA, R., AND PUGLIESE, R. 1999. Proof techniques for cryptographic processes. In *Proceedings of the Annual IEEE Symposium on Login in Computer Science (LICS)*. IEEE, Computer Society Press, 157–166.

BOREALE, M. AND SANGIORGI, D. 1998. Bisimulation in name-passing calculi without matching. In *Proceedings of the Annual IEEE Symposium on Logic in Computer Science (LICS)*. IEEE Computer Society Press.

CAO, Z. 2006. More on bisimulations for higher order *pi*-calculus. In *Proc. FoSSaCS'06*. Lecture Notes in Computer Science, vol. 3921. Springer, 63–78.

CRARY, K. AND HARPER, R. 2000. Syntactic logical relations over polymorphic and recursive types. Draft.

FERREIRA, W., HENNESSY, M., AND JEFFREY, A. 1998. A theory of weak bisimulation for core CML. *J. Func. Program. 8,* 5, 447–491.

GODSKESEN, J. AND HILDEBRANDT, T. 2005. Extending Howe's method to early bisimulations for typed mobile embedded resources with local names. In *Proceedings of the Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*. Lecture Notes in Computer Science, vol. 3821. Springer, 140–151.

GORDON, A. D. 1993. Functional programming and input/output. Ph.D. thesis, University of Cambridge.

HONDA, K. AND YOSHIDA, N. 1995. On reduction-based process semantics. *Theor. Comput. Sci. 152,* 2, 437–486.

HOWE, D. J. 1996. Proving congruence of bisimulation in functional programming languages. *Info. Comput. 124,* 2, 103–112.

JEFFREY, A. AND RATHKE, J. 1999. Towards a theory of bisimulation for local names. In *Proceedings of the Annual IEEE Symposium on Logic in Computer Science (LICS)*. 56–66.

JEFFREY, A. AND RATHKE, J. 2004. A theory of bisimulation for a fragment of concurrent ml with local names. *Theor. Comput. Sci. 323,* 1-3, 1–48.

JEFFREY, A. AND RATHKE, J. 2005. Contextual equivalence for higher-order pi-calculus revisited. *Logic. Meth. Comput. Sci. 1,* 1:4, 1–22.

KOUTAVAS, V. AND WAND, M. 2006a. Bisimulations for untyped imperative objects. In *Proceedings of the European Symposium on Programming (ESOP)*. 146–161.

KOUTAVAS, V. AND WAND, M. 2006b. Small bisimulations for reasoning about higher-order imperative programs. In *Proceedings of the ACM Symposium on Principles of Programming Languages (POPL)*. 141–152.

LASSEN, S. B. 1998. Relational reasoning about functions and nondeterminism. Ph.D. thesis, Department of Computer Science, University of Aarhus.

LENGLET, S. 2010. Bisimulations dans les calculs avec passivation. Ph.D. thesis, Université Joseph Fourier, Grenoble.

LENGLET, S., SCHMITT, A., AND STEFANI, J.-B. 2009. Howe's method for calculi with passivation. In *Proceedings of the International Conference on Concurrency Theory (CONCUR)*. Lecture Notes in Computer Science, vol. 5710. 448–462.

MERRO, M. AND HENNESSY, M. 2002. Bisimulation congruences in Safe Ambients. In *Proceedings of the ACM Symposium on Principles of Programming Languages (POPL)*. 71–80.

MERRO, M. AND NARDELLI, F. Z. 2005. Behavioral theory for mobile ambients. *J. ACM 52,* 6, 961–1023.

MILNER, R. 1989. *Communication and Concurrency*. Prentice Hall.

MILNER, R. 1999. *Communicating and Mobile Systems: the π-Calculus*. Cambridge University Press.

MILNER, R. 2006. Pure bigraphs: Structure and dynamics. *Inform. Comput. 204,* 1, 60–122.

MITCHELL, J. C. 1996. *Foundations for Programming Languages*. MIT Press.

MORRIS, J. H., JR. 1968. Lambda-calculus models of programming languages. Ph.D. thesis, Massachusetts Institute of Technology.

ONG, L. 1988. The lazy lambda calculus: an investigation into the foundations of functional programming. Ph.D. thesis, University of London.

PIERCE, B. C. AND SANGIORGI, D. 2000. Behavioral equivalence in the polymorphic pi-calculus. *J. ACM 47,* 3, 531–586.

PITTS, A. 1997. Operationally-based theories of program equivalence. In *Semantics and Logics of Computation*, A. M. Pitts and P. Dybjer Eds., Publications of the Newton Institute., Cambridge University Press, 241–298.

PITTS, A. 2005. Typed operational reasoning. In *Advanced Topics in Types and Programming Languages*, B. C. Pierce Ed., MIT Press, Chapter 7, 245–289.

POUS, D. 2008. Techniques modulo pour les bisimulations. Ph.D. thesis, ENS Lyon.

POUS, D. AND SANGIORGI, D. Enhancements of the bisimulation proof method.

REYNOLDS, J. C. 1983. Types, abstraction and parametric polymorphism. In *Inf. Proceedings of the IFIP 9th World Computer Congres*. 513–523.

SANDS, D. 1998. Improvement theory and its applications. In *Higher Order Operational Techniques in Semantics*, A. D. Gordon and A. M. Pitts Eds., Publications of the Newton Institute, Cambridge University Press, 275–306.

SANGIORGI, D. 1992. Expressing mobility in process algebras: First-order and higher-order paradigm. Ph.D. thesis, University of Edinburgh.

SANGIORGI, D. 1994. The lazy lambda calculus in a concurrency scenario. *Inform. Comput. 111,* 1, 120–153.

SANGIORGI, D. 1996. Bisimulation for Higher-Order Process Calculi. *Inform. Comput. 131,* 2, 141–178.

SANGIORGI, D. 2001. Asynchronous process calculi: the first-order and higher-order paradigms (tutorial). *Theor. Comput. Sci. 253*, 311–350.

SANGIORGI, D., KOBAYASHI, N., AND SUMII, E. 2007. Logical bisimulations and functional languages. In *Proceedings of the international Symposium on Fundamentals of Software Engineering (FSEN)*. Lecture Notes in Computer Science, vol. 4767. Springer, 364–379.

SANGIORGI, D. AND WALKER, D. 2001. *The π-calculus: a Theory of Mobile Processes*. Cambridge University Press.

SCHMITT, A. AND STEFANI, J.-B. 2004. The kell calculus: A family of higher-order distributed process calculi. In *Proceedings of the International Conference on Global Computing*. Lecture Notes in Computer Science, vol. 3267. Springer, 146–178.

STØVRING, K. AND LASSEN, S. B. 2007. A complete, co-inductive syntactic theory of sequential control and state. In *Proceedings of the ACM Symposium on Principles of Programming Languages (POPL)*. 161–172. To appear.

SUMII, E. 2009. A complete characterization of observational equivalence in polymorphic λ-calculus with general references. In *Proceedings of the International Conference on Computer Science Logic*, Lecture Notes in Computer Science, vol. 5771, Springer-Verlag, 455–469.

SUMII, E. AND PIERCE, B. C. 2003. Logical relations for encryption. *J. Comput. Secu. 11,* 4, 521–554.

SUMII, E. AND PIERCE, B. C. 2007a. A bisimulation for dynamic sealing. *Theor. Comput. Sci. 375,* 1-3, 169–192.

SUMII, E. AND PIERCE, B. C. 2007b. A bisimulation for type abstraction and recursion. *J. ACM 54,* 5.

THOMSEN, B. 1993. Plain CHOCS, a second generation calculus for higher-order processes. *Acta Informatica 30*, 1–59.

Received May 2009; revised January 2010; accepted March 2010