Appendices to "Environmental Bisimulations for Higher-Order Languages"

Davide Sangiorgi
University of Bologna
Davide.Sangiorgi@cs.unibo.it

Naoki Kobayashi Eijiro Sumii
Tohoku University
{koba,sumii}@ecei.tohoku.ac.jp

A Additional material on Section 3

Congruence proofs Lemmas 3.7 and 3.8 are proved simultaneously, thus:

Lemma A.1 *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

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

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

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

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 $\mathcal{E}^\star$. This is sufficient, because $\mathcal{E}^\star \in \mathcal{X}$. 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

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

and $\mathcal{E}^\star$ 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. □

The next lemma shows that $\simeq$ is preserved by all contexts (this is Lemma 3.10 from the main text).

Lemma A.2 *$M \simeq N$ implies, for all context C, $C[M] \simeq C[N]$.*

Proof The following relation is a bisimulation up-to bisimilarity:

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

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

The case of elements of $\simeq^\star$, 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 $P\{^V/x\} \ \mathcal{X}_{\simeq^\star} \ Q\{^W/x\}$. □

Up-to techniques

Definition A.3 *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* $\mathcal{E}' \supseteq \mathcal{E}$.

Definition A.4 *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* $P\{^{M_1}/x\} \simeq \mathcal{X}_{\mathcal{E}} \simeq Q\{^{N_1}/x\}$.

Definition A.5 *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* $M'\ \mathcal{X}_{\mathcal{E}}\ N'$.

We write $M\ \mathcal{X}_{\mathcal{E}}^{(\star)}\ N$ if 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 $M'\ \mathcal{X}_{\mathcal{E}}\ N'$; or else $M = C[\widetilde{V}], N = C[\widetilde{W}]$, and $V_i\ \mathcal{E}\ W_i$ and $\mathcal{E} \in \mathcal{X}$.

Definition A.6 *An environmental relation* $\mathcal{X}$ *is an* environmental bisimulation up to contexts *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* $\mathcal{E} \cup \{(V, W)\} \subseteq \mathcal{E}'^\star$, *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* $P\{^{M_1}/x\}\ \mathcal{X}_{\mathcal{E}}^{(\star)}\ Q\{^{N_1}/x\}$.

B Additional material for logical bisimulations

This section has been superceeded by "Logical Bisimulations and Functional Languages" to appear in Post-Proceedings of IPM International Symposium on Fundamentals of Software Engineering (FSEN 07), Lecture Notes in Computer Science, Springer-Verlag. The former Example B.7 now corresponds to Section 3.4, Example 1.

C 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:

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

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

We highlight what changes in the theory for call-by-name of the previous sections. For a relation $\mathcal{R}$ we write $\mathcal{R}^{\widehat{\star}}$ 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 $\mathcal{E}^{\widehat{\star}}$ (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.

All up-to techniques described for call-by-name are valid also for call-by-value. In addition, we can also derive the soundness (and completeness) of a form of logical bisimulation with big-step restricted to values (in call-by-value, applicative bisimulation is normally defined this way) that we call *value big-step logical bisimulation*.

Definition C.1 *A relation* $\mathcal{E}$ *on closed values is a* value big-step logical bisimulation *if for all* $V\ \mathcal{E}\ W$ *and* $V_1\ \mathcal{E}^{\widehat{\star}}\ W_1$, *if* $VV_1 \Longrightarrow V'$ *then there is* W' *such that* $WW_1 \Longrightarrow W'$ *and* $V'\mathcal{E}W'$; *and the converse, on the reductions from* W.

Example C.2 *This example uses a simply-typed call-by-value 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

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

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

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 $g^n(m)$ if $n \geq 0$, diverge otherwise. In both cases, however, the computations made are different. We show $F_1\, g\, n\, m \cong F_2\, g\, n\, m$ using an up-to technique for logical bisimulations. For this, we use the following relation $\mathcal{R}$:

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

We show that $\mathcal{R}$ is a logical 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:

$$\begin{array}{rcl} g^r(F_1\, g\, 0\, m) & \longrightarrow\!\!\!\!\Longrightarrow & g^r(m) \\ & & \mathcal{R}^\star \\ F_2\, g\, n\, (g^r(m)) & \longrightarrow\!\!\!\!\succeq & g^r(m) \end{array}$$

So, the required condition holds. 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

$$\begin{array}{rcl} F_2\, g\, n\, (g^r(m)) & \longrightarrow\!\!\!\!\succeq & F_2\, g\, (n \hat{-} 1)\, (g(g^r(m))) \\ & \succeq & F_2\, g\, (n-1)\, (g^{r+1}(m)). \end{array}$$

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

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

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.

D Additional material for the imperative call-by-value λ-calculus

Syntax and reduction rules

We use s, t to range over stores, i.e., finite mappings from locations to closed values and l, k over locations. Then $s[l = V]$ 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. $\star$ is the unit value (i.e., nullary tuple). 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:

$$\begin{array}{rcll} M & ::= & x & \text{variables} \\ & \mid & c & \text{constants} \\ & \mid & \lambda x.\, M & \text{functions} \\ & \mid & M_1 M_2 & \text{applications} \\ & \mid & \texttt{op}(M_1, \ldots, M_n) & \text{primitive operations} \\ & \mid & \texttt{if}\ M_1\ \texttt{then}\ M_2\ \texttt{else}\ M_3 & \text{if-then-else} \\ & \mid & \#_i(M) & \text{projection} \\ & \mid & (M_1, \ldots, M_n) & \text{tuples} \\ & \mid & l & \text{locations} \\ & \mid & \boldsymbol{\nu} x\, M & \text{new locations} \\ & \mid & !M & \text{deferencing} \\ & \mid & M_1 := M_2 & \text{assignments} \end{array}$$

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.

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.

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

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

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

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

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

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

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

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

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

where evaluation contexts are:

$$\begin{array}{rcl} C & := & [\cdot] \mid CM \mid VC \mid \texttt{op}(\widetilde{c}, C, \widetilde{M}) \mid (\widetilde{V}, C, \widetilde{M}) \mid \#_i C \mid \\ & & \texttt{if}\ C\ \texttt{then}\ M_1\ \texttt{else}\ M_2 \mid !C \mid C := M \mid l := C \end{array}$$

Contextual equivalence

We recall the standard definition of contextual equivalence.

Definition D.1 (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 $\langle s\,;\,C[N]\rangle \Downarrow$.*

Remark D.2 [on Lemma 3.10] The correspondent of Lemma 3.10 does not hold in the imperative λ-calculus, i.e., we cannot replace evaluation contexts with arbitrary contexts in Lemma 4.4. For example, let M and N be $\texttt{if}\ !l = 0\ \texttt{then}\ l := 1\ \texttt{else}\ \Omega$ 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 $C = [\cdot]_1; [\cdot]_1$. To obtain the congruence theorem we have to refine the lemma, using the same bisimulation requirement as in Theorem 4.5 (the free locations in the tested terms should be made available, as values, to the observer).

Up to environment, reduction and contexts

We write $\langle s\,;\,M\rangle\ \mathcal{X}_{\mathcal{E}}^{\rightarrow\star}\ \langle t\,;\,N\rangle$ 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 D.3 (up to environment, reduction, and contexts) *An environmental relation $\mathcal{X}$ is a* bisimulation up to environment, reduction, and contexts *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 $\langle s'\,;\,M'\rangle\ \mathcal{X}_{\mathcal{E}'}^{\rightarrow\star}\ \langle t\,;\,N\rangle$ for some $\mathcal{E}'$ with $\mathcal{E} \subseteq \mathcal{E}'$;*

(b) *if $M = V$ then $\langle t\,;\,N\rangle \Longrightarrow \langle t'\,;\,W\rangle$ (where W is a value) 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 above 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 $\langle s\,;\,P\{^{V_1}/x\}\rangle\ \mathcal{X}_{\mathcal{E}'}^{\rightarrow\star}\ \langle t\,;\,Q\{^{W_1}/x\}\rangle$, for some $\mathcal{E}'$ with $\mathcal{E} \subseteq \mathcal{E}'$;*

(f) *the converse of the above five conditions, on W.*

E Additional definitions for HOπ

Syntax Below is the syntax for HOπ.

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

The replication operator ($!P$) is derivable (see [37, 43]).

Transitions Now the LTS. There are three forms of transitions: τ transitions $M \xrightarrow{\tau} M'$; input transitions $M \xrightarrow{aM'} M''$, meaning that M receives at name a the term M' and then evolves to M''; and output transitions $M \xrightarrow{(\boldsymbol{\nu}\widetilde{b})\overline{a}M'} M''$ meaning that M emits M' at a, and in doing so it extrudes the names $\widetilde{b}$ and evolves to M'' (names $\widetilde{b}$ are private between M' and M''). We use α to indicate a generic label of a transition.

$$a(x).\,P \xrightarrow{aQ} P\{^{Q}/x\}$$

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

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

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

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

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

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.

An up-to technique We report the full definition of the up-to technique mentioned in Section 5.

A context C is an *evaluation context* if the hole $[\cdot]_1$ occurs in C exactly once, and not underneath a prefix. We write $P \ \mathcal{X}^{\mathrm{c}}_{\mathcal{E}\,;\,r} \ Q$ if there is an evaluation context C and processes $\widetilde{M}, \widetilde{N}$ such that:

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

We write $P \ \mathcal{X}^{\mathrm{c1}}_{\mathcal{E}\,;\,r} \ Q$ if there is an evaluation context C such that:

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

In *strong environmental bisimilarity* (whose basic properties, including congruence, are established in the same manner as for weak bisimilarity) only strong transitions are used in the bisimulation game. We write $\sim$ for the strong version of $\simeq$.

Definition E.1 *An environmental relation $\mathcal{X}$ is an* environmental bisimulation up-to contexts, bisimilarity, environment *if $M \ \mathcal{X}_{\mathcal{E}\,;\,r} \ N$ implies:*

1. *if $M \xrightarrow{\tau} M'$ then $N \stackrel{\tau}{\Longrightarrow} N'$ and $M' \sim \mathcal{X}^{\mathrm{c}}_{\mathcal{E}\,;\,r} \simeq N'$;*
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 $M' \sim \mathcal{X}^{\mathrm{c}}_{\mathcal{E}\,;\,r} \simeq N'$;*
3. *if $M \xrightarrow{(\boldsymbol{\nu}\widetilde{b})\overline{a}P} M'$ with $a \in r$, then there are $Q, N', \widetilde{c}$ such that $N \stackrel{(\boldsymbol{\nu}\widetilde{c})\overline{a}Q}{\Longrightarrow} N'$ and $M' \sim \mathcal{X}^{\mathrm{c}}_{\mathcal{E},(P,Q)\,;\,r} \simeq N'$;*
4. *if $(P, Q) \in \mathcal{E}$ then $P \mid M \sim \mathcal{X}^{\mathrm{c1}}_{\mathcal{E}\,;\,r} \sim Q \mid N$*
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 the actions from N.*

Expansion could also be used, in place of $\sim$.