Bisimulations in SSCC

Luís Cruz-Filipe
Ivan Lanese
Francisco Martins
António Ravara
Vasco Vasconcelos

DI–FCUL TR–2007–37

December 2007

Departamento de Informática
Faculdade de Ciências da Universidade de Lisboa
Campo Grande, 1749–016 Lisboa
Portugal

Bisimulations in SSCC

Luís Cruz-Filipe
Dept. Informatics
Faculty of Sciences
University of Lisbon

Ivan Lanese
Computer Science Department,
University of Bologna

Francisco Martins
Dept. Informatics
Faculty of Sciences
University of Lisbon

António Ravara
SQIG, Instituto de Telecomunicações
and Dept. Mathematics, IST,
Technical University of Lisbon

Vasco Vasconcelos
Dept. Informatics
Faculty of Sciences
University of Lisbon

December 2007

Abstract

This report studies different definitions of bisimulation within the Stream-Based Service-Centered Calculus (SSCC) and shows that both strong and weak ground bisimulation are non-input congruences.

Keywords: Process calculus, bisimulation, services.

1 Introduction

The Stream-based Service-Centered Calculus (SSCC) [3] looks for solutions to expressiveness problems posed by the original Service-Centered Calculus [2]. It differs from the latter in the existence of streams, which are the privileged means of communication between services and the processes invoking them.

This note studies two notions of bisimilarity in SSCC, based on what was already proposed in the original paper. We show that both strong and weak bisimilarity are non-input congruences in the class of SSCC processes. Although the general strategy is the same as for π-calculus, the proof techniques themselves differ significantly.

We define a labeled transition system (LTS) in the early style, and study notions of bisimilarity known in the literature as *strong ground bisimilarity* and *weak ground bisimilarity.* The reason for choosing these is simple: we are interested in capturing contextual equivalence, so bisimilarity should be a congruence. Therefore, we choose the simplest possible setting where this may happen. It is well-known already from the π-calculus that ground bisimilarity over a late LTS is not preserved by parallel composition, requiring the more demanding notions of late and early bisimilarity (which in turn are not preserved by input prefix, since they are not closed under general substitutions). Not surprisingly, this fact also occurs in SSCC: ground bisimilarity is a non-input congruence. Sangiorgi and Walker present counter-examples for the preservation of both strong and weak bisimulation in the synchronous π-calculus without sum and match (see pages 224 and 225 of [4]). Of these, the first can be easily translated to our language, using sessions to mimic π's input and output constructors; for the weak case, the counter-example is not so directly transposable, but it can still be adapted.

The LTS we define herein adapts the original one (albeit in an insignificant way in terms of expressive power, as discussed below) to allow for the results we are seeking.

2 Language

Recall the definition of SSCC processes.

Definition 1. *The* (run-time) processes *of SSCC are generated by the following grammar.*

$$\begin{array}{rcl} P, Q & := & \mathsf{0} \;[\!]\; v.P \;[\!]\; (x).P \;[\!]\; P \mid Q \;[\!]\; (\nu a)P \;[\!]\; X \;[\!]\; \mathsf{rec}\; X.P \\ & & [\!]\; a \Rightarrow P \;[\!]\; a \Leftarrow P \;[\!]\; r \rhd P \;[\!]\; r \lhd P \;[\!]\; (\nu r)P \\ & & [\!]\; \mathsf{feed}\; v.P \;[\!]\; f(x).P \;[\!]\; \mathsf{stream}\; P \;\mathsf{as}\; f = \vec{v} \;\mathsf{in}\; Q \end{array}$$

To simplify notation, when stream f is empty, we write $\mathsf{stream}\; P \;\mathsf{as}\; f \;\mathsf{in}\; Q$ for $\mathsf{stream}\; P \;\mathsf{as}\; f = \langle\rangle \;\mathsf{in}\; Q$.

In processes $(x).P$, $f(x).P$, $(\nu a)P$, and $(\nu r)P$, occurrences of names x, a, r are bound in P. We assume the Barendregt convention on variables (i.e., bound occurrences are all distinct, and differ from all the free variables [1]); furthermore, a process is well-formed only if it contains no free process variables. Abbreviations like $r \bowtie P$ and $a \Leftrightarrow (r)$ are defined in [3].

Definition 2. *The set of* actions*, ranged over by μ, is defined by the following grammar.*

$$\begin{array}{rcl} \mu & ::= & \tau \;[\!]\! \uparrow v \;[\!]\! \downarrow v \;[\!]\; (a) \downarrow v \;[\!]\! \Uparrow v \;[\!]\; f \Downarrow v \;[\!]\; (a) f \Downarrow v \\ & & a \Leftarrow (r) \;[\!]\; a \Rightarrow (r) \;[\!]\; r \bowtie \uparrow v \;[\!]\; r \bowtie (a) \uparrow v \;[\!]\; r \bowtie \downarrow v \;[\!]\; r \bowtie (a) \downarrow v \end{array}$$

The intended meaning of these actions is as follows:

- τ is the internal action;
- for every value v, $\uparrow v$ and $\downarrow v$ are the output and input actions, respectively;
- for every value v and every stream f, $\Uparrow v$ and $f \Downarrow v$ are the feed and read actions, respectively;
- for every service name a and session name r, $a \Leftarrow (r)$ and $a \Rightarrow (r)$ are the invocation and acknowledge actions, respectively;
- for every output or feed action μ, the name a is restricted in $(a)\mu$;

- for every session name r and (restricted) output or input action μ, $r \bowtie \mu$ is the corresponding action within session r.

The labeled transition system for SSCC is slightly different from that of [3]. The reason of this is that the presence of the congruence rule L-STRUCT poses unique problems when doing proofs by induction; see the discussion at the end of Section 4.

Definition 3. *The* labeled transition system *for SSCC contains all the rules in [3] except for* L-STRUCT*, together with the following new rules:*

$$\frac{Q \xrightarrow{\mu} Q' \qquad \mathsf{bn}(\mu \cap \mathsf{fn}(P)) = \emptyset}{P \mid Q \xrightarrow{\mu} P \mid Q'}\ \textsc{L-par}' \qquad \frac{P\left[{}^{\mathsf{rec}\ X.P}/_{X}\right] \xrightarrow{\mu} P'}{\mathsf{rec}\ X.P \xrightarrow{\mu} P'}\ \textsc{L-rec}$$

$$\frac{P \xrightarrow{r\bowtie(a)\uparrow a} P' \qquad Q \xrightarrow{r\overline{\bowtie}\downarrow a} Q'}{P \mid Q \xrightarrow{r\tau} (\nu a)(P' \mid Q')}\ \textsc{L-par-close}$$

$$\frac{P \xrightarrow{r\bowtie\downarrow a} P' \qquad Q \xrightarrow{r\overline{\bowtie}(a)\uparrow a} Q'}{P \mid Q \xrightarrow{r\tau} (\nu a)(P' \mid Q')}\ \textsc{L-par-close}'$$

$$\frac{P \xrightarrow{(a)\Uparrow a} P'}{\mathsf{stream}\ P\ \mathsf{as}\ f = \vec{w}\ \mathsf{in}\ Q \xrightarrow{\tau} (\nu a)(\mathsf{stream}\ P'\ \mathsf{as}\ f = a :: \vec{w}\ \mathsf{in}\ Q)}\ \textsc{L-feed-close}$$

$$\frac{P \xrightarrow{r\bowtie(a)\uparrow a} P' \qquad Q \xrightarrow{r\overline{\bowtie}\downarrow a} Q'}{\mathsf{stream}\ P\ \mathsf{as}\ f\ \mathsf{in}\ Q \xrightarrow{r\tau} (\nu a)(\mathsf{stream}\ P'\ \mathsf{as}\ f\ \mathsf{in}\ Q')}\ \textsc{L-sess-com-close}$$

$$\frac{P \xrightarrow{r\bowtie\downarrow a} P' \qquad Q \xrightarrow{r\overline{\bowtie}(a)\uparrow a} Q'}{\mathsf{stream}\ P\ \mathsf{as}\ f\ \mathsf{in}\ Q \xrightarrow{r\tau} (\nu a)(\mathsf{stream}\ P'\ \mathsf{as}\ f\ \mathsf{in}\ Q')}\ \textsc{L-sess-com-close}'$$

The following lemmas show the derivability of the new transition rules in the original system.

Lemma 1. *Rule* L-PAR' *is admissible in the original LTS for SSCC.*

Proof. The following derivation shows that any instance of L-PAR' can be derived in the old LTS.

$$\frac{\dfrac{Q \xrightarrow{\mu} Q' \quad \mathsf{bn}(\mu \cap \mathsf{fn}(P)) = \emptyset}{Q \mid P \xrightarrow{\mu} Q' \mid P}\ \textsc{L-par} \qquad Q \mid P \equiv P \mid Q \quad Q' \mid P \equiv P \mid Q'}{P \mid Q \xrightarrow{\mu} P \mid Q'}\ \textsc{L-str}$$

□

Lemma 2. *Rule* L-REC *is admissible in the original LTS for SSCC.*

Proof. The following derivation shows that any instance of L-REC can be derived in the old LTS.

$$\frac{P\left[{}^{\mathsf{rec}\ X.P}/_{X}\right] \xrightarrow{\mu} P' \qquad P\left[{}^{\mathsf{rec}\ X.P}/_{X}\right] \equiv \mathsf{rec}\ X.P \qquad P' \equiv P'}{\mathsf{rec}\ X.P \xrightarrow{\mu} P'}\ \textsc{L-struct}$$

□

Lemma 3. *Let P and P' be processes and a be a name.*

(i) If $P \xrightarrow{(a)\uparrow a} P'$ then $P \equiv (\nu a)R$ for some R such that $R \xrightarrow{\uparrow a} P'$.

(ii) If $P \xrightarrow{r\bowtie(a)\uparrow a} P'$ then $P \equiv (\nu a)R$ for some R such that $R \xrightarrow{r\bowtie\uparrow a} P'$.

(iii) If $P \xrightarrow{(a)\Uparrow a} P'$ then $P \equiv (\nu a)R$ for some R such that $R \xrightarrow{\Uparrow a} P'$.

Proof. All three parts of the lemma are proved by induction on the proof of the transition.

For (i), the base case is when rule L-EXTR is applied. Then the thesis follows immediately from the premise of the rule and reflexivity of $\equiv$. The induction cases are when one out of L-PAR, L-PAR', L-STREAM-PASS-P, L-STREAM-PASS-Q, L-RES or L-REC is applied.

The first four cases are analogous. Suppose rule L-PAR was applied; then P is $P_1 \mid P_2$, P' is $P_1' \mid P_2$, $P_1 \xrightarrow{(a)\uparrow a} P_1'$ and a is not a free name of P_2. By induction hypothesis, $P_1 \equiv (\nu a)R$ with $R \xrightarrow{\uparrow a} P_1'$; also $P_1 \mid P_2 \equiv (\nu a)R \mid P_2$. By rule L-PAR, $R \mid P_2 \xrightarrow{\uparrow a} P'$. The case of L-REC is also straightforward: since $\mathsf{rec}\ X.P \equiv P\left[{}^{\mathsf{rec}\ X.P}/_{X}\right]$, the induction hypothesis immediately establishes the result. Finally, for L-RES, simply apply the induction hypothesis and use S-SWAP to conclude the thesis.

The proof of (ii) is completely similar except for the base case. Here, the rule being applied may also be L-SESS-VAL, in which case P is $r \bowtie Q$ and P' is $r \bowtie Q'$. By (i), also $Q \equiv (\nu a)R$ with $R \xrightarrow{\uparrow a} Q'$, whence $r \bowtie R \xrightarrow{r\bowtie\uparrow a} P'$. Since $r \bowtie (\nu a)R \equiv (\nu a)r \bowtie R$, the thesis follows.

The last case is analogous to the first. □

Lemma 4. *Rules* L-PAR-CLOSE *and* L-PAR-CLOSE' *are admissible in the original LTS for SSCC.*

Proof. Suppose $P \mid Q \xrightarrow{r\tau} (\nu a)P' \mid Q'$ by L-PAR-CLOSE. By (ii) of Lemma 3, there exists R such that $P \equiv (\nu a)R$ and $R \xrightarrow{r\bowtie\uparrow a} P'$. The following derivation shows that this instance of L-PAR-CLOSE can be derived in the old LTS.

$$\dfrac{\dfrac{\dfrac{R \xrightarrow{r\bowtie\uparrow a} P' \quad Q \xrightarrow{r\bowtie\downarrow a} Q'}{R \mid Q \xrightarrow{r\tau} P' \mid Q'}\ \text{L-PAR}}{(\nu a)R \mid Q \xrightarrow{r\tau} (\nu a)P' \mid Q'}\ \text{L-RES} \qquad (\nu a)R \equiv P \quad Q \equiv Q}{P \mid Q \xrightarrow{r\tau} (\nu a)P' \mid Q'}\ \text{L-STRUCT}$$

For rule L-PAR-CLOSE', apply the previous construction with L-PAR' instead of L-PAR and invoke Lemma 1. □

Lemma 5. *Rules* L-SESS-CLOSE *and* L-SESS-CLOSE' *are admissible in the original LTS for SSCC.*

Proof. Analogous to the previous one. □

Lemma 6. *Rule* L-FEED-CLOSE *is admissible in the original LTS for SSCC.*

Proof. Suppose $\mathsf{stream}\ P\ \mathsf{as}\ f = \vec{w}\ \mathsf{in}\ Q \xrightarrow{\tau} (\nu a)\mathsf{stream}\ P'\ \mathsf{as}\ f = a :: \vec{w}\ \mathsf{in}\ Q$ by L-FEED-CLOSE. By part (iii) of Lemma 3, $P \equiv (\nu a)R$ for some R such that $R \xrightarrow{\Uparrow a} P'$. The following derivation shows that this instance of L-PAR-CLOSE can be derived in the old LTS.

$$\dfrac{\dfrac{\dfrac{R \xrightarrow{\Uparrow a} P'}{\mathsf{stream}\ R\ \mathsf{as}\ f = \vec{w}\ \mathsf{in}\ Q \xrightarrow{\tau} \mathsf{stream}\ P'\ \mathsf{as}\ f = a :: \vec{w}\ \mathsf{in}\ Q}\ \text{L-STREAM-FEED}}{(\nu a)\mathsf{stream}\ R\ \mathsf{as}\ f = \vec{w}\ \mathsf{in}\ Q \xrightarrow{\tau} (\nu a)\mathsf{stream}\ P'\ \mathsf{as}\ f = a :: \vec{w}\ \mathsf{in}\ Q}\ \text{L-RES}}{\mathsf{stream}\ P\ \mathsf{as}\ f = \vec{w}\ \mathsf{in}\ Q \xrightarrow{\tau} (\nu a)\mathsf{stream}\ P'\ \mathsf{as}\ f = a :: \vec{w}\ \mathsf{in}\ Q}\ \text{L-STRUCT}$$

Observe that the application of L-STRUCT is sound: from $(\nu a)R \equiv P$, it follows that $(\nu a)\mathsf{stream}\ R\ \mathsf{as}\ f = \vec{w}\ \mathsf{in}\ Q \equiv \mathsf{stream}\ P\ \mathsf{as}\ f = \vec{w}\ \mathsf{in}\ Q$ (for the lefthand-side); also $\equiv$ is reflexive (for the righthand-side). □

This LTS is strictly weaker than the original one, since it is not always the case that two structurally congruent processes can always evolve via the same action to the same process, as the following example shows.

Example 1. *In the original LTS, $a \mid b \xrightarrow{\uparrow a} b \mid \mathsf{0}$, whereas this does not hold with the new LTS: the proof uses* L-SEND*,* L-PAR *and* L-STRUCT*, and this last step cannot be captured in the modified LTS.*

However, it is always the case that structurally congruent processes can evolve via the same action to structurally congruent processes; this is precisely the statement of the Harmony Lemma, proved in the next section. Observe that this is sufficient for both LTSs to yield the same notion of bisimilarity: the Harmony Lemma implies that structural equivalence is a bisimulation, while all other transition rules are preserved.

Example 1 (contd)**.** *In the modified LTS,* L-SEND *and* L-PAR *can be used to infer that $a \mid b \xrightarrow{\uparrow a} \mathsf{0} \mid b$. The latter process is structurally congruent to $b \mid \mathsf{0}$.*

3 The Harmony Lemma

Theorem 1 (Harmony Lemma)**.** *Let P and Q be processes with $P \equiv Q$. If $P \xrightarrow{\alpha} P'$, then $Q \xrightarrow{\alpha} Q'$ with $P' \equiv Q'$, and vice-versa.*

Proof. By induction on the proof that $P \equiv Q$.

- *Equivalence relation*
  - Reflexivity. Immediate, taking Q' to be P'.
  - Symmetry. Immediate consequence of the induction hypothesis, since the thesis of the theorem is symmetric.
  - Transitivity. Assume $P \equiv Q$ because $P \equiv R$ and $R \equiv Q$, and suppose that $P \xrightarrow{\alpha} P'$. By induction hypothesis, $R \xrightarrow{\alpha} R'$ with $P' \equiv R'$; hence, again by induction hypothesis, $Q \xrightarrow{\alpha} Q'$ with $R' \equiv Q'$. By transitivity of $\equiv$, it follows that $P' \equiv Q'$.
- *Congruence properties*

- Parallel composition. Suppose $P \equiv Q$. For each of the possible transitions of $P \mid R$, it is straightforward to verify that $Q \mid R$ can simulate them, eventually using the induction hypothesis; similarly, $R \mid Q$ can simulate $R \mid P$. Notice that the side conditions in the transition rules always hold since structurally congruent processes have the same free names.
- Composition with stream. Analogous.
- Name restriction. Suppose $P \equiv Q$ and let a be a name. For each of the possible transitions of $(\nu a)P$, it is again easy to check that $(\nu a)Q$ can simulate them, eventually using the induction hypothesis.
- Session input/output. Straightforward, observing (for input) that structural congruence is preserved under substitution.
- Stream input/output. Analogous.
- Service definition/invocation. Straightforward.

• *Monoid structure*

- Unit. Let Q be $P \mid \mathsf{0}$. If $P \xrightarrow{\alpha} P'$, then by rule L-PAR also $P \mid \mathsf{0} \xrightarrow{\alpha} P' \mid \mathsf{0}$, since $\mathsf{0}$ has no free names, and the latter process is congruent to P'. Reciprocally, if $P \mid \mathsf{0} \xrightarrow{\alpha} P$', then the only rule that can have been applied is L-PAR (since $\mathsf{0} \not\rightarrow$), whence P' is $P'' \mid \mathsf{0}$ with $P \xrightarrow{\alpha} P''$.
- Commutativity. Assume P is $R \mid S$ and Q is $S \mid R$. Take any proof of $R \mid S \xrightarrow{\alpha} T$ and replace occurrences of L-PAR by L-PAR', of L-SESS-COM-PAR by L-SESS-COM-PAR', of L-SERV-COM-PAR by L-SERV-COM-PAR' and vice-versa; it is straightforward to verify that this yields a proof that $S \mid R \xrightarrow{\alpha} T'$ with $T \equiv T'$. The converse is analogous.
- Associativity. Let P be $R \mid (S \mid T)$ and Q be $(R \mid S) \mid T$. Suppose that $P \xrightarrow{\alpha} P'$; there are six rules that can be used in the last step of the proof of this transition. For simplicity, in the proofs below we omit side conditions related to bound names, since it is simple to verify that they always follow from the assumptions.
  - ∗ L-PAR: then $R \xrightarrow{\alpha} R'$ and P' is $R' \mid (S \mid T)$. The following proof shows that $(R \mid S) \mid T \xrightarrow{\alpha} (R' \mid S) \mid T$, which establishes the thesis.

$$\cfrac{\cfrac{R \xrightarrow{\alpha} R'}{R \mid S \xrightarrow{\alpha} R' \mid S}\ \text{L-PAR}}{(R \mid S) \mid T \xrightarrow{\alpha} (R' \mid S) \mid T}\ \text{L-PAR}$$

  - ∗ L-PAR': then $S \mid T \xrightarrow{\alpha} U$; there are six sub-cases, according to the rule used to derive this transition.
    - · The rule applied is L-PAR, so $S \xrightarrow{\alpha} S'$ and U is $S' \mid T$; then the following proof establishes the thesis.

$$\cfrac{\cfrac{S \xrightarrow{\alpha} S'}{R \mid S \xrightarrow{\alpha} R \mid S'}\ \text{L-PAR'}}{(R \mid S) \mid T \xrightarrow{\alpha} (R \mid S') \mid T}\ \text{L-PAR}$$

· The rule applied is L-PAR’, so $T \xrightarrow{\alpha} T'$ and U is $S \mid T'$; then the following proof establishes the thesis.

$$\frac{T \xrightarrow{\alpha} T'}{(R \mid S) \mid T \xrightarrow{\alpha} (R \mid S) \mid T'}\ \text{L-PAR'}$$

· The rule applied is L-SESS-COM-PAR, so $S \xrightarrow{r \bowtie \updownarrow v} S'$, $T \xrightarrow{r \overline{\bowtie} \overline{\updownarrow} v} T'$, U is $S' \mid T'$ and α is $r\tau$ for some fresh r; then the following proof establishes the thesis.

$$\frac{\dfrac{S \xrightarrow{r \bowtie \updownarrow v} S'}{R \mid S \xrightarrow{r \bowtie \updownarrow v} R \mid S'}\ \text{L-PAR'} \qquad T \xrightarrow{r \overline{\bowtie} \overline{\updownarrow} v} T'}{(R \mid S) \mid T \xrightarrow{r\tau} (R \mid S') \mid T'}\ \text{L-SESS-COM-PAR}$$

· The rule applied is L-SERV-COM-PAR, so α is τ, $S \xrightarrow{a \Leftrightarrow (r)} S'$, $T \xrightarrow{a \overline{\Leftrightarrow} (r)} T'$ and U is $(\nu r)(S' \mid T')$; then the following proof establishes the thesis, since $(\nu r)((R \mid S') \mid T') \equiv R \mid ((\nu r)(S' \mid T'))$ as r is not a free name of R.

$$\frac{\dfrac{S \xrightarrow{a \Leftrightarrow (r)} S'}{R \mid S \xrightarrow{a \Leftrightarrow (r)} R \mid S'}\ \text{L-PAR'} \qquad T \xrightarrow{a \overline{\Leftrightarrow} (r)} T'}{(R \mid S) \mid T \xrightarrow{\tau} (\nu r)((R \mid S') \mid T')}\ \text{L-SERV-COM-PAR}$$

· The rule applied is L-PAR-CLOSE, so $S \xrightarrow{r \bowtie (a) \uparrow a} S'$, $T \xrightarrow{r \overline{\bowtie} \downarrow a} T'$, U is $(\nu a)(S' \mid T')$ and α is $r\tau$. This case is analogous to that of L-SESS-COM-PAR, the extra name restriction in the resulting processes posing no additional problem.

· The rule applied is L-PAR-CLOSE’, so $S \xrightarrow{r \bowtie \downarrow a} S'$, $T \xrightarrow{r \overline{\bowtie} (a) \uparrow a} T'$, U is $(\nu a)(S' \mid T')$ and α is $r\tau$. This case is analogous to the previous one.

∗ L-SESS-COM-PAR: then $R \xrightarrow{r \bowtie \updownarrow v} R'$ and $S \mid T \xrightarrow{r \overline{\bowtie} \overline{\updownarrow} v} U$; there are two similar sub-cases, according to whether the last transition is proved via L-PAR or via L-PAR’. Without loss of generality, assume that the former is the case; then the following proof establishes the thesis.

$$\frac{\dfrac{R \xrightarrow{r \bowtie \updownarrow v} R' \quad S \xrightarrow{r \overline{\bowtie} \overline{\updownarrow} v} S'}{R \mid S \xrightarrow{r\tau} R' \mid S'}\ \text{L-SESS-COM-PAR}}{(R \mid S) \mid T \xrightarrow{r\tau} (R' \mid S') \mid T}\ \text{L-PAR'}$$

∗ L-SERV-COM-PAR: then $R \xrightarrow{a \Leftrightarrow (r)} R'$ and $S \mid T \xrightarrow{a \overline{\Leftrightarrow} (r)} U$; again there are two similar sub-cases, according to whether the last transition is proved via L-PAR or via L-PAR’. Without loss of generality, assume that the former is the case; then the following proof establishes the thesis.

$$\dfrac{\dfrac{R \xrightarrow{a\Leftrightarrow(r)} R' \quad S \xrightarrow{a\overline{\Leftrightarrow}(r)} S'}{R \mid S \xrightarrow{\tau} (\nu r)(R' \mid S')}\ \text{L-SERV-COM-PAR}}{(R \mid S) \mid T \xrightarrow{\tau} (\nu r)(R' \mid S') \mid T}\ \text{L-PAR'}$$

Since r is not a free name of T, the latter process is structurally congruent to $(\nu r)(R' \mid (S' \mid T))$.

  * L-PAR-CLOSE: then $R \xrightarrow{r\bowtie(a)\uparrow a} R'$ and $S \mid T \xrightarrow{r\overline{\bowtie}\downarrow a} U$. This case is analogous to that of L-SESS-COM-PAR, the extra name restrictions in the resulting processes posing no additional problem.
  * L-PAR-CLOSE': then $R \xrightarrow{r\bowtie\downarrow a} R'$ and $S \mid T \xrightarrow{r\overline{\bowtie}(a)\uparrow a} U$. This case is again analogous to the previous one.

The case when $Q \xrightarrow{\alpha} Q'$ is dealt with by a similar case analysis.

- *Name restriction*
  - Parallel composition. Suppose P is $((\nu n)R) \mid S$ and Q is $(\nu n)(R \mid S)$. Assume first that $P \xrightarrow{\alpha} P'$; there are three different cases, according to which transition rule was used.
    * L-PAR: then $(\nu n)R \xrightarrow{\alpha} R'$. There are three possible sub-cases.
      · Suppose $(\nu n)R \xrightarrow{\alpha} R'$ follows by L-RES. Then n is not a name in α, R' is $(\nu n)R''$ and $R \xrightarrow{\alpha} R''$. Since n is also not a name in S, the following derivation establishes the thesis.

$$\dfrac{\dfrac{R \xrightarrow{\alpha} R''}{R \mid S \xrightarrow{\alpha} R'' \mid S}\ \text{L-PAR}}{(\nu n)(R \mid S) \xrightarrow{\alpha} (\nu n)(R'' \mid S)}\ \text{L-RES}$$

      · Suppose $(\nu n)R \xrightarrow{\alpha} R'$ follows by L-SESS-RES. Then α is τ, R' is $(\nu n)R''$ and $R \xrightarrow{n\tau} R''$. Again, since n is also not a name in S, the following derivation establishes the thesis.

$$\dfrac{\dfrac{R \xrightarrow{n\tau} R''}{R \mid S \xrightarrow{n\tau} R'' \mid S}\ \text{L-PAR}}{(\nu n)(R \mid S) \xrightarrow{\tau} (\nu n)(R'' \mid S)}\ \text{L-SESS-RES}$$

      · Suppose $(\nu n)R \xrightarrow{\alpha} R'$ follows by L-EXTR. Then α is $(n)\uparrow n$ and $R \xrightarrow{\uparrow n} R'$. Again, since n is also not a name in S, the following derivation establishes the thesis.

$$\dfrac{\dfrac{R \xrightarrow{\uparrow n} R'}{R \mid S \xrightarrow{\uparrow n} R' \mid S}\ \text{L-PAR}}{(\nu n)(R \mid S) \xrightarrow{(n)\uparrow n} (R' \mid S)}\ \text{L-EXTR}$$

In either case, it is easy to verify that $(\nu n)(R \mid S)$ evolves to a process structurally congruent to the evolution of $((\nu n)R) \mid S$.

* L-PAR': then $S \xrightarrow{\alpha} S'$. Since $(\nu n)R \mid S$ is well-formed, n does not occur in S; therefore n cannot occur in α. Then the following derivation establishes the thesis.

$$\dfrac{\dfrac{S \xrightarrow{\alpha} S'}{R \mid S \xrightarrow{\alpha} R \mid S'}\ \text{L-PAR'}}{(\nu n)(R \mid S) \xrightarrow{\alpha} (\nu n)(R \mid S')}\ \text{L-RES}$$

* L-SESS-COM-PAR: then α is $r\tau$, $(\nu n)R \xrightarrow{r\bowtie\updownarrow v} R'$ and $S \xrightarrow{r\overline{\bowtie}\overline{\updownarrow}v} S'$. Then necessarily R' is $(\nu n)R''$ and the former transition is inferred via L-RES. The following derivation establishes the thesis.

$$\dfrac{\dfrac{R \xrightarrow{r\bowtie\updownarrow v} R'' \qquad S \xrightarrow{r\overline{\bowtie}\overline{\updownarrow}v} S'}{R \mid S \xrightarrow{r\tau} R'' \mid S'}\ \text{L-SESS-COM-PAR}}{(\nu n)(R \mid S) \xrightarrow{r\tau} (\nu n)(R'' \mid S')}\ \text{L-RES}$$

  The cases when the rule applied is L-SERV-COM-PAR, L-PAR-CLOSE or L-PAR-CLOSE' are similar, except that further applications of S-SWAP may be necessary to verify that both processes evolve to structurally congruent processes.

Assume now that $Q \xrightarrow{\alpha} Q'$. Since the top-level constructor in Q is name restriction, there are three possible cases.

* Assume the last rule applied is L-RES. Then $R \mid S \xrightarrow{\alpha} T$, with Q' being $(\nu n)T$ and n a name not occurring in α. There are six sub-cases, corresponding to the six different rules that may be used to infer the transition of $R \mid S$.
  · L-PAR: then T is $R' \mid S$ with $R \xrightarrow{\alpha} R'$; then the following proof establishes the thesis.

$$\dfrac{\dfrac{R \xrightarrow{\alpha} R'}{(\nu n)R \xrightarrow{\alpha} (\nu n)R'}\ \text{L-RES}}{((\nu n)R) \mid S \xrightarrow{\alpha} ((\nu n)R') \mid S}\ \text{L-PAR}$$

  · L-PAR': then T is $R \mid S'$ with $S \xrightarrow{\alpha} S'$; the following proof establishes the thesis.

$$\dfrac{S \xrightarrow{\alpha} S'}{((\nu n)R) \mid S \xrightarrow{\alpha} ((\nu n)R) \mid S'}\ \text{L-PAR'}$$

  · L-SESS-COM-PAR: then α is $r\tau$, $R \xrightarrow{r\bowtie\updownarrow v} R'$, $S \xrightarrow{r\overline{\bowtie}\overline{\updownarrow}v} S'$ and T is $R' \mid S'$. Care must be taken to distinguish whether n is v.
  If n is not v, then the following derivation establishes the thesis.

$$\dfrac{\dfrac{R \xrightarrow{r\bowtie\updownarrow v} R'}{(\nu n)R \xrightarrow{r\bowtie\updownarrow v} (\nu n)R'}\ \text{L-RES} \qquad S \xrightarrow{r\overline{\bowtie}\overline{\updownarrow}v} S'}{((\nu n)R) \mid S \xrightarrow{r\tau} ((\nu n)R') \mid S'}\ \text{L-SESS-COM-PAR}$$

If n is v, then by well-formedness the process performing the output must be R (otherwise S would contain a binder n, which violates the assumption that all bound names in $R \mid S$ are distinct); the following proof establishes the thesis.

$$\dfrac{\dfrac{R \xrightarrow{r\bowtie\updownarrow v} R'}{(\nu n)R \xrightarrow{r\bowtie(n)\updownarrow v} R'}\ \textsc{L-extr} \qquad S \xrightarrow{r\overline{\bowtie}\overline{\updownarrow} v} S'}{((\nu n)R) \mid S \xrightarrow{r\tau} (\nu n)(R' \mid S')}\ \textsc{L-sess-close}$$

· L-SERV-COM-PAR: then α is τ, $R \xrightarrow{a\Leftrightarrow(r)} R'$, $S \xrightarrow{a\overline{\Leftrightarrow}(r)} S'$ and T is $(\nu r)(R' \mid S')$. Notice that from the hypothesis it follows that n is distinct from r. Consider the following derivation.

$$\dfrac{\dfrac{R \xrightarrow{a\Leftrightarrow(r)} R'}{(\nu n)R \xrightarrow{a\Leftrightarrow(r)} (\nu n)R'}\ \textsc{L-res} \qquad S \xrightarrow{a\overline{\Leftrightarrow}(r)} S'}{((\nu n)R) \mid S \xrightarrow{\tau} (\nu r)(((\nu n)R') \mid S')}\ \textsc{L-serv-com-par}$$

Finally, $(\nu r)(((\nu n)R') \mid S') \equiv (\nu r)(\nu n)(R' \mid S') \equiv Q'$ follows by S-EXTR-PAR and S-SWAP.

· L-PAR-CLOSE: then α is $r\tau$, $R \xrightarrow{r\bowtie(a)\uparrow a} R'$, $S \xrightarrow{r\overline{\bowtie}\downarrow a} S'$ and T is $(\nu a)(R' \mid S')$. Again, by well-formedness, a is distinct from n. The following derivation establishes the thesis.

$$\dfrac{\dfrac{R \xrightarrow{r\bowtie(a)\uparrow a} R'}{(\nu n)R \xrightarrow{r\bowtie(a)\uparrow a} (\nu n)R'}\ \textsc{L-res} \qquad S \xrightarrow{r\overline{\bowtie}\downarrow a} S'}{((\nu n)R) \mid S \xrightarrow{r\tau} (\nu a)(((\nu n)R') \mid S')}\ \textsc{L-sess-close}$$

· L-PAR-CLOSE': then α is $r\tau$, $R \xrightarrow{r\bowtie\downarrow a} R'$, $S \xrightarrow{r\overline{\bowtie}(a)\uparrow a} S'$ and T is $(\nu a)(R' \mid S')$. Again, by well-formedness, a is distinct from n. The following derivation establishes the thesis.

$$\dfrac{\dfrac{R \xrightarrow{r\bowtie\downarrow a} R'}{(\nu n)R \xrightarrow{r\bowtie\downarrow a} (\nu n)R'}\ \textsc{L-res} \qquad S \xrightarrow{r\overline{\bowtie}(a)\uparrow a} S'}{((\nu n)R) \mid S \xrightarrow{r\tau} (\nu a)(((\nu n)R') \mid S')}\ \textsc{L-sess-close'}$$

∗ Assume the last rule applied is L-SESS-RES. This case is very similar to the previous one, but simpler: since session names may not be communicated, there are less possible cases and no need arises to use close rules.

∗ Assume the last rule applied is L-EXTR. Then α is $(n)\mu$, where μ is an output (session or stream). By well-formedness, n does not occur in S, whence necessarily $R \xrightarrow{\mu} R'$ and Q' is $R' \mid S$. Then the following proof shows that $P \xrightarrow{\alpha} Q'$.

$$\dfrac{\dfrac{R \xrightarrow{\mu} R'}{(\nu n)R \xrightarrow{(n)\mu} R'}\ \textsc{L-extr}}{((\nu n)R) \mid S \xrightarrow{(n)\mu} R' \mid S}\ \textsc{L-par}$$

– Composition with stream. There are two congruence rules for this case; both of them require a case analysis that is completely similar to that in the previous case (since composition with a stream is very similar to parallel composition). The extra case arising from L-FEED-CLOSE is similar to the other close rules.

– Session. Assume P is $r \bowtie ((\nu a)R)$ and Q is $(\nu a)(r \bowtie R)$. Suppose first that $P \xrightarrow{\alpha} P'$; there are two different cases.

  ∗ L-SESS-VAL: then α is $r \bowtie \mu$, where μ is an input/output action. There are two possible sub-cases, according to how the transition of $(\nu a)R$ is inferred (since L-SESS-RES does not apply).

    · L-RES: then P' is $r \bowtie ((\nu a)R')$ with $R \xrightarrow{\mu} R'$, and the following derivation establishes the thesis.

$$\dfrac{\dfrac{R \xrightarrow{\mu} R'}{r \bowtie R \xrightarrow{r \bowtie \mu} r \bowtie R'}\ \text{L-SESS-VAL}}{(\nu a)(r \bowtie R) \xrightarrow{r \bowtie \mu} (\nu a)(r \bowtie R')}\ \text{L-RES}$$

    · L-EXTR: then P' is $r \bowtie R'$, μ is $\uparrow a$, and the following derivation establishes the thesis.

$$\dfrac{\dfrac{R \xrightarrow{\uparrow a} R'}{r \bowtie R \xrightarrow{r \bowtie \uparrow a} r \bowtie R'}\ \text{L-SESS-VAL}}{(\nu a)(r \bowtie R) \xrightarrow{r \bowtie (a) \uparrow a} r \bowtie R'}\ \text{L-EXTR}$$

  ∗ L-SESS-PASS: this case is very similar with only two differences. In the case of L-EXTR, μ is now $\Uparrow a$, and the rest follows as before. There is also the extra case of L-SESS-RES, which is straightforward.

  Assume next that $Q \xrightarrow{\alpha} Q'$. The proof is very similar, so we will only sketch it; there are three cases.

  ∗ L-RES: then Q' is $(\nu a)S$ with $r \bowtie R \xrightarrow{\alpha} S$. There are two cases for the latter transition; in either of them, S must be of the form $r \bowtie R'$ and the thesis follows by swapping the application of the two rules.

  ∗ L-SESS-RES: similar, but now there is only one sub-case, corresponding to L-SESS-PASS.

  ∗ L-EXTR: then $r \bowtie R \xrightarrow{\mu} Q'$ and either α is $(a)\mu$ or α is $s \bowtie (a) \uparrow a$ and μ is $s \bowtie a$ for some session name s. Again there are two cases for the latter transition, and a straightforward swapping of the two rules yields the proof that $P \xrightarrow{\alpha} Q'$.

– Commutativity. Straightforward, since two different names are involved and well-formedness of the processes guarantees that all side conditions in the relevant rules will hold.

– Zero. Straightforward, since $(\nu a)\mathsf{0} \not\rightarrow$ and $\mathsf{0} \not\rightarrow$.

- *Recursion* This case is completely straightforward: if $\mathsf{rec}\ X.R \xrightarrow{\alpha} P'$, then the only rule that can have been used to infer that transition is L-REC, whence it immediately follows that $R\left[{}^{\mathsf{rec}\ X.R}/_{X}\right] \xrightarrow{\alpha} P'$. Reciprocally, if the latter condition holds, then by L-REC also $\mathsf{rec}\ X.R \xrightarrow{\alpha} P'$. □

4 Strong bisimilarity

Strong bisimilarity, hereafter referred to simply as "bisimilarity", is defined as usual over the class of all processes.

Definition 4.

- *A symmetric binary relation $\mathcal{R}$ on processes is a* (strong) bisimulation *if, for any processes P, Q such that $P\mathcal{R}Q$, if $P \xrightarrow{\alpha} P'$ for some process P' and action α such that no bound name in α is free in P or Q, there exists a process Q' such that $Q \xrightarrow{\alpha} Q'$ for some Q' with $P' \mathcal{R} Q'$.*
- *(Strong) bisimilarity $\sim$ is the largest bisimulation.*
- *Two processes P and Q are said to be* (strongly) bisimilar *if $P \sim Q$.*

Notice that bisimilarity can be obtained as the union of all bisimulations or as a fixed-point of a suitable monotonic operator; also it is well defined, as the next result shows.

Theorem 2. *Structurally congruent processes are bisimilar.*

Proof. It suffices to show that $\equiv$ is a bisimulation, which is an immediate consequence of the Harmony Lemma. □

We now show that bisimilarity is a non-input congruence, just as in π-calculus. The strategy of the proof is the same as in [4], based on the notion and properties of a relation *progressing* to another relation.

Definition 5. *A relation $\mathcal{R}$ on processes* strongly progresses *to another relation $\mathcal{S}$, denoted $\mathcal{R} \rightsquigarrow \mathcal{S}$, if, whenever $P\mathcal{R}Q$, $P \xrightarrow{\alpha} P'$ implies $Q \xrightarrow{\alpha} Q'$ for some Q' with $P'\mathcal{S}Q'$, and vice-versa.*

Definition 6. *A function $\mathcal{F}$ on processes is* strongly safe *if $\mathcal{R} \subseteq S$ and $\mathcal{R} \rightsquigarrow \mathcal{S}$ imply $\mathcal{F}(\mathcal{R}) \subseteq \mathcal{F}(\mathcal{S})$ and $\mathcal{F}(\mathcal{R}) \rightsquigarrow \mathcal{F}(\mathcal{S})$.*

Lemma 7. *If $\mathcal{F}$ is strongly safe and $\sim \subseteq \mathcal{F}(\sim)$, then $F(\sim) = \sim$.*

Proof. See [4]. □

Given a function $\mathcal{F}$, define $\mathcal{F}^*$ such that $\mathcal{F}^*(\mathcal{R})$ is the transitive closure of $\mathcal{F}(\mathcal{R})$.

Lemma 8. *If $\mathcal{F}$ is such that $\mathcal{R} \subseteq S$ and $\mathcal{R} \rightsquigarrow \mathcal{S}$ imply that $\mathcal{F}(\mathcal{R}) \subseteq \mathcal{F}^*(\mathcal{S})$ and $\mathcal{F}(\mathcal{R}) \rightsquigarrow \mathcal{F}^*(\mathcal{S})$, then $\mathcal{F}^*$ is strongly safe.*

Proof. See [4]. □

The proof relies on defining functions $\mathcal{F}_{\text{ni1}}$ and $\mathcal{F}_{\text{ni}}$ like those for π-calculus; however, the definition of the former has to be slightly adapted.

Definition 7.

- *A* context *C is a process where exactly one occurrence of $\mathsf{0}$ has been replaced by a hole $[\cdot]$. Given a process P, $C[P]$ is the process obtained by replacing the hole in C by P.*

- *An n-ary* multi-hole context *C is a process where some occurrences of* $\mathsf{0}$ *have been replaced by holes $[\cdot]_i$; each hole may occur zero or more times. Given n processes $P_1, \ldots, P_n$, $C[P_1, \ldots, P_n]$ is the process obtained by uniformly replacing all occurrences of all holes in C by the corresponding process.*

- *A (multi-hole) context is said to be* non-input *if no hole occurs under an input prefix (x) or $f(x)$.*

- *Functions $\mathcal{F}_{ni1}$ and $\mathcal{F}_{ni}$ are defined as follows.*

$$\begin{aligned}\mathcal{F}_{ni1}(\mathcal{R}) &= \{\langle C[P], C[Q]\rangle \;[]\; P\mathcal{R}Q \text{ and } C \text{ is a non-input context}\}\\ \mathcal{F}_{ni}(\mathcal{R}) &= \{\langle C[P_1,\ldots,P_n], C[Q_1,\ldots,Q_n]\rangle \;[]\; P_i\mathcal{R}Q_i\\ &\qquad\qquad \text{and } C \text{ is an } n\text{-ary non-input context}\}\end{aligned}$$

Lemma 9. $\mathcal{F}_{ni} = \mathcal{F}^*_{ni1}$.

Proof. As for π-calculus. □

Lemma 10. *Function $\mathcal{F}_{ni}$ is strongly safe.*

Proof. Applying Lemma 8, it must be shown that, whenever $\mathcal{R} \subseteq \mathcal{S}$ and $\mathcal{R} \rightsquigarrow \mathcal{S}$, both $\mathcal{F}_{\mathrm{ni1}}(\mathcal{R}) \subseteq \mathcal{F}_{\mathrm{ni1}}(\mathcal{S})$ and $\mathcal{F}_{\mathrm{ni1}}(\mathcal{R}) \rightsquigarrow \mathcal{F}_{\mathrm{ni}}(\mathcal{S})$. The first of these is trivial by definition of $\mathcal{F}_{\mathrm{ni1}}$.

Assume that $P\mathcal{R}Q$. It must be shown that, for every context C, if $C[P] \xrightarrow{\alpha} P'$, then $C[Q] \xrightarrow{\alpha} Q'$ for some P' and Q' such that there exist an n-ary context C' and processes $P_1\mathcal{S}Q_1, \ldots, P_n\mathcal{S}Q_n$ for which P' is $C'[P_1, \ldots, P_n]$ and Q' is $C'[Q_1, \ldots, Q_n]$.

The proof is by induction on the derivation tree for $C[P] \xrightarrow{\alpha} P'$. In all steps, there are two cases to consider, according to whether C is $[\cdot]$ or not; the former case is always trivial, since the hypothesis $\mathcal{R} \rightsquigarrow \mathcal{S}$ establishes the thesis. Therefore, we always assume below that C is not $[\cdot]$. The proof looks at the last rule being applied.

- L-SEND: then C is $v.C_0$ and α is $\uparrow v$ for some v. Furthermore, $v.C_0[Q] \xrightarrow{\uparrow v} C_0[Q]$; since C_0 is also a multi-hole context and $\mathcal{R} \subseteq \mathcal{S}$, it follows that $\langle C_0[P], C_0[Q]\rangle \in \mathcal{F}_{\mathrm{ni}}(\mathcal{S})$, hence the thesis holds.

- L-RECEIVE: then C is $(x).C_0$, and since C is a non-input context (by definition of $\mathcal{F}_{\mathrm{ni1}}$), it follows that C_0 does not contain holes; hence in this case $C[P]$ and $C[Q]$ coincide, and the result is trivial.

- L-FEED: then C is $\mathsf{feed}\ v.C_0$ and α is $\Uparrow v$ for some v. Furthermore, $\mathsf{feed}\ v.C_0[Q] \xrightarrow{\Uparrow v} C_0[Q]$; since C_0 is also a multi-hole context and $\mathcal{R} \subseteq \mathcal{S}$, it follows that $\langle C_0[P], C_0[Q]\rangle \in \mathcal{F}_{\mathrm{ni}}(\mathcal{S})$, hence the thesis holds.

- L-READ: then C is $f(x).C_0$, and since C is a non-input context (by definition of $\mathcal{F}_{\mathrm{ni1}}$), it follows that C_0 does not contain holes; hence in this case $C[P]$ and $C[Q]$ coincide, and the result is trivial.

- L-CALL: then C is $a \Leftarrow C_0$ and α is $a \Leftarrow (r)$ for some r not occurring free in $C_0[P]$. Furthermore, $a \Leftarrow C_0[Q] \xrightarrow{a \Leftarrow (r)} r \triangleleft C_0[Q]$, since by definition of bisimulation r does not occur free in $C_0[Q]$. Taking C' to be the context $r \triangleleft C_0$ establishes the thesis.

- L-INV: analogous.

- L-PAR: there are two cases to consider.

  - If C is $C_0 \mid R$, then $C_0[P] \xrightarrow{\alpha} P'$ and α and R share no bounded names. By induction hypothesis there exists a process Q' such that $C_0[Q] \xrightarrow{\alpha} Q'$ and P', Q' are $C_0'[P_1, \ldots, P_n]$ and $C_0'[Q_1, \ldots, Q_n]$, respectively, for some n-ary multi-hole context C_0' and processes $P_1 \mathcal{S} Q_1, \ldots, P_n \mathcal{S} Q_n$. Thus $C_0[Q] \mid R \xrightarrow{\alpha} Q' \mid R$, hence taking C' to be $C_0' \mid R$ establishes the thesis.
  - If C is $R \mid C_0$, then $R \xrightarrow{\alpha} R'$ and $C_0[P]$ and R share no bounded names. By the hypothesis of L-PAR, $C_0[Q]$ and R also share no bounded names, hence $R \mid C_0[Q] \xrightarrow{\alpha} R \mid Q'$, and taking C' to be $R \mid C_0'$ establishes the thesis.

- L-PAR': analogous (the two cases are reversed).

- L-STREAM-PASS-P and L-STREAM-PASS-Q: analogous to L-PAR and L-PAR', respectively.

- L-STREAM-FEED: there are two cases to consider.

  - If C is $\mathsf{stream}\ C_0\ \mathsf{as}\ f = \vec{w}\ \mathsf{in}\ R$, then $C_0[P] \xrightarrow{\Uparrow v} P'$; by induction hypothesis, there exists a process Q' such that $C_0[Q] \xrightarrow{\Uparrow v} Q'$, and P' and Q' are respectively $C_0'[P_1, \ldots, P_n]$ and $C_0'[Q_1, \ldots, Q_n]$ for some n-ary multi-hole context C_0' and processes $P_1 \mathcal{S} Q_1, \ldots, P_n \mathcal{S} Q_n$. Thus $\mathsf{stream}\ C_0[Q]\ \mathsf{as}\ f = \vec{w}\ \mathsf{in}\ R \xrightarrow{\tau} \mathsf{stream}\ Q'\ \mathsf{as}\ f = v :: \vec{w}\ \mathsf{in}\ R$, hence taking C' to be $\mathsf{stream}\ C_0'\ \mathsf{as}\ f = v :: \vec{w}\ \mathsf{in}\ R$ establishes the thesis.
  - If C is $\mathsf{stream}\ R\ \mathsf{as}\ f = \vec{w}\ \mathsf{in}\ C_0$, then $R \xrightarrow{\Uparrow v} R'$, hence taking C' to be $\mathsf{stream}\ R'\ \mathsf{as}\ f = v :: \vec{w}\ \mathsf{in}\ C_0$ immediately establishes the thesis.

- L-STREAM-CONS: analogous (the two cases are reversed).

- L-SESS-VAL: then C is $r \bowtie C_0$, α is $r \bowtie \mu$ for some μ, and $C_0[P] \xrightarrow{\mu} P_0$. By induction hypothesis, $C_0[Q] \xrightarrow{\mu} Q_0$ for some Q_0 such that there exist a multi-hole context C_0' and processes $P_1 \mathcal{S} Q_1, \ldots, P_n \mathcal{S} Q_n$ for which P_0 is $C_0'[P_1, \ldots, P_n]$ and Q_0 is $C_0'[Q_1, \ldots, Q_n]$. Taking C' to be $r \bowtie C_0'$ establishes the thesis, since then $r \bowtie C_0[Q] \xrightarrow{r \bowtie \mu} C'[Q_1, \ldots, Q_n]$.

- L-SESS-PASS: then C is $r \bowtie C_0$, α is neither an input nor an output, and $C_0[P] \xrightarrow{\alpha} P_0$. The proof then follows as above except that the action does not change when the session is added to C_0'.

- L-SESS-COM-PAR: C is either $C_0 \mid R$ or $R \mid C_0$; the two cases are analogous, so assume the first holds. Then $C_0[P] \xrightarrow{r \bowtie \Updownarrow v} P'$, $R \xrightarrow{r \overline{\bowtie} \overline{\Updownarrow} v} R'$, α is

$r\tau$ for some r and $C_0[P] \mid R \xrightarrow{r\tau} P' \mid R'$. By induction hypothesis there exists a process Q' such that $C_0[Q] \xrightarrow{r \bowtie \updownarrow v} Q'$, and P', Q' are respectively $C'_0[P_1, \dots, P_n]$ and $C'_0[Q_1, \dots, Q_n]$ for some n-ary multi-hole context C'_0 and processes $P_1 \mathcal{S} Q_1, \dots, P_n \mathcal{S} Q_n$. Then $C_0[Q] \mid R \xrightarrow{r\tau} Q' \mid R'$, hence taking C' to be $C'_0 \mid R'$ establishes the thesis.

- L-SERV-COM-PAR,L-PAR-CLOSE, L-PAR-CLOSE', L-FEED-CLOSE, L-SESS-COM-CLOSE, L-SERV-COM-CLOSE, L-SESS-COM-STREAM and L-SERV-COM-STREAM: these cases are all very similar to the previous one.
- L-RES: then C is $(\nu a)C_0$, $(\nu a)C_0[P] \xrightarrow{\alpha} (\nu a)P'$, a is not a name in α and $C_0[P] \xrightarrow{\alpha} P'$. By induction hypothesis there exists a process Q' such that $C_0[Q] \xrightarrow{\alpha} Q'$, and P', Q' are respectively $C'_0[P_1, \dots, P_n]$ and $C'_0[Q_1, \dots, Q_n]$ for some n-ary multi-hole context C'_0 and processes $P_1 \mathcal{S} Q_1, \dots, P_n \mathcal{S} Q_n$. Thus $(\nu a)C_0[Q] \xrightarrow{\alpha} (\nu a)Q'$, hence taking C' to be $(\nu a)C'_0$ establishes the thesis.
- L-SESS-RES: analogous, only now α is τ and the induction hypothesis is applied to a transition via $a\tau$.
- L-EXTR: analogous, only now α is an action extruding a and the induction hypothesis is applied to a transition without the extrusion; furthermore, the context C' is simply C'_0.
- L-REC: the C is $\mathsf{rec}\ X.C_0$ and $C_0[P] \left[{}^{\mathsf{rec}\ X.C_0[P]}/_X\right] \xrightarrow{\alpha} P'$. Since P is a well-formed process, it contains no free occurrences of X; hence there exists a context C_1 such that $C_1[P]$ is $C_0[P] \left[{}^{\mathsf{rec}\ X.C_0[P]}/_X\right]$ and $C_1[Q]$ is $C_0[Q] \left[{}^{\mathsf{rec}\ X.C_0[Q]}/_X\right]$. Hence the induction hypothesis applies, and there exist a process Q' and a context C' such that $C_1[Q] \xrightarrow{\alpha} Q'$, P' is $C'[P]$ and Q' is $C'[Q]$. Therefore also $\mathsf{rec}\ X.C_0[Q] \xrightarrow{\alpha} Q'$, and C' is the required context. □

Theorem 3. *Bisimilarity is a non-input congruence.*

Proof. Straightforward consequence of Lemmas 7 and 10. □

At this point we can explain in more detail why the original LTS for SSCC had to be changed. Consider any derivation containing an application of L-STRUCT.

$$\frac{P \xrightarrow{\mu} P' \qquad P \equiv Q \qquad P' \equiv Q'}{Q \xrightarrow{\mu} Q'}\ \text{L-STRUCT}$$

In general, the induction hypothesis will not be applicable to the subtree that shows $P \xrightarrow{\mu} P'$, since there is no obvious relationship between P and Q; furthermore, the thesis of the induction hypothesis does not help in establishing the final result, since again there is no obvious relationship between Q' and P'.

Observe also that this is not a problem of this particular proof technique. Whether the induction were on the derivation tree (as above), on contexts (as the proof for π-calculus, see [4]) or on processes (arguably an alternative) the same problem would arise, since the issue arises from the fact that the theorem assumes hypotheses on the actual process performing the transition. This justifies the attempt to eliminate L-STRUCT from the LTS altogether.

5 Weak bisimilarity

Weak bisimilarity treats internal actions as irrelevant. As usual, we introduce some abbreviations.

$$\begin{array}{lll} P \stackrel{\tau}{\Rightarrow} Q & \text{iff} & P \stackrel{\tau}{\rightarrow} \cdots \stackrel{\tau}{\rightarrow} Q \\ P \stackrel{\alpha}{\Rightarrow} Q & \text{iff} & P \stackrel{\tau}{\Rightarrow} \stackrel{\alpha}{\rightarrow} \stackrel{\tau}{\Rightarrow} Q \quad \text{for } \alpha \neq \tau \end{array}$$

Notice that, in particular, $P \stackrel{\tau}{\Rightarrow} P$ for every process P.

Definition 8.

- *A symmetric binary relation $\mathcal{R}$ on processes is a* weak bisimulation *if, for any processes P, Q such that $P\mathcal{R}Q$, if $P \stackrel{\alpha}{\Rightarrow} P'$ for some process P' and action α such that no bound name in α is free in P or Q, there exists a process Q' such that $Q \stackrel{\alpha}{\Rightarrow} Q'$ for some Q' with $P' \mathcal{R} Q'$.*
- *Weak bisimilarity $\approx$ is the largest weak bisimulation.*
- *Two processes P and Q are said to be* weakly bisimilar *if $P \approx Q$.*

Again, weak bisimilarity can be obtained as the union of all weak bisimulations or as a fixed-point of a suitable monotonic operator.

Theorem 4. *Let $\simeq$ be the largest relation such that, whenever $P \simeq Q$, for every process P' and action α, if $P \stackrel{\alpha}{\rightarrow} P'$, then $Q \stackrel{\alpha}{\Rightarrow} Q'$ for some Q' with $P' \simeq Q'$ and vice-versa. Then $P \approx Q$ iff $P \simeq Q$.*

Proof. The direct implication is straightforward, since $P \stackrel{\alpha}{\rightarrow} P'$ implies that $P \stackrel{\alpha}{\Rightarrow} P'$. For the converse, assume that $P \stackrel{\alpha}{\Rightarrow} P'$. If P' is P and α is τ, then result is trivial; otherwise, there exist processes $P_1, \ldots, P_n$ and $P'_1, \ldots, P'_m$ such that P is P_1, $P_i \stackrel{\tau}{\rightarrow} P_{i+1}$ for $i < n$, $P_n \stackrel{\alpha}{\rightarrow} P'_1$, $P'_j \stackrel{\tau}{\rightarrow} P'_{j+1}$ for $j < m$ and P'_m is P'. By hypothesis, there exist processes $Q_1, \ldots, Q_n$ and $Q'_1, \ldots, Q'_m$ (not necessarily distinct) such that $Q_i \stackrel{\tau}{\Rightarrow} Q_{i+1}$ for $i < n$, $Q_n \stackrel{\alpha}{\Rightarrow} Q'_1$ and $Q'_j \stackrel{\tau}{\Rightarrow} Q'_{j+1}$ for $j < m$; furthermore, $P_i \simeq Q_i$ and $P'_j \simeq Q'_j$ for all $i \leq n$ and $j \leq m$. In particular, $Q \stackrel{\alpha}{\Rightarrow} Q'_m$ and $P' \simeq Q'_m$, so $\simeq$ is a weak bisimulation. □

The reason for introducing $\simeq$ is that this relation is simpler to work with when proving properties by induction. In turn, the definition of $\approx$ is more symmetric and its relationship with $\sim$ is immediate.

We now show that bisimilarity is a non-input congruence, again like in π-calculus. The strategy of the proof is once more the same as in [4].

Definition 9. *A relation $\mathcal{R}$ on processes* progresses *to another relation $\mathcal{S}$, denoted $\mathcal{R} \rightsquigarrow \mathcal{S}$, if, whenever $P\mathcal{R}Q$, $P \stackrel{\alpha}{\rightarrow} P'$ implies $Q \stackrel{\alpha}{\Rightarrow} Q'$ for some Q' with $P'\mathcal{S}Q'$, and vice-versa.*

Definition 10. *A function $\mathcal{F}$ on processes is* safe *if $\mathcal{R} \subseteq S$ and $\mathcal{R} \rightsquigarrow \mathcal{S}$ imply $\mathcal{F}(\mathcal{R}) \subseteq \mathcal{F}(\mathcal{S})$ and $\mathcal{F}(\mathcal{R}) \rightsquigarrow \mathcal{F}(\mathcal{S})$.*

Lemma 11. *If $\mathcal{F}$ is safe and $\approx \subseteq \mathcal{F}(\approx)$, then $F(\approx) = \approx$.*

Proof. See [4]. □

As is the case with π-calculus, proving that $\mathcal{F}_{\rm ni}$ is safe must be done directly, since chaining is not secure.

Lemma 12. *Function $\mathcal{F}_{ni}$ is safe.*

Proof. Let $\mathcal{R} \subseteq \mathcal{S}$ and $\mathcal{R} \rightsquigarrow \mathcal{S}$. It must be shown that $\mathcal{F}_{\rm ni1}(\mathcal{R}) \subseteq \mathcal{F}_{\rm ni1}(\mathcal{S})$ and $\mathcal{F}_{\rm ni}(\mathcal{R}) \rightsquigarrow \mathcal{F}_{\rm ni}(\mathcal{S})$. As before, the first of these is trivial by definition of $\mathcal{F}_{\rm ni1}$.

Assume that $P_i \mathcal{R} Q_i$ for $i = 1, \ldots, n$. It must be shown that, for every multi-context C, if $C[P_1, \ldots, P_n] \xrightarrow{\alpha} P'$, then $C[Q_1, \ldots, Q_n] \overset{\alpha}{\Rightarrow} Q'$ for some P' and Q' such that there exist another multi-context C' and processes $P_1 \mathcal{S} Q_1, \ldots, P_m \mathcal{S} Q_m$ for which P' is $C'[P_1, \ldots, P_m]$ and Q' is $C'[Q_1, \ldots, Q_m]$.

Once more we use induction on the derivation tree for $C[P_1, \ldots, P_n] \xrightarrow{\alpha} P'$. Most cases are very similar to the proof of Lemma 10; however, since the induction hypothesis now gives a weak transition some care must be taken.

As before, the case when C is $[\cdot]$ is straightforward; also, the cases of rules L-SEND, L-RECEIVE, L-FEED, L-READ, L-CALL, L-INV, L-SESS-VAL, L-SESS-PASS, L-SESS-COM-PAR, L-SERV-COM-PAR, L-SESS-COM-STREAM, L-SERV-COM-STREAM,L-PAR-CLOSE, L-PAR-CLOSE', L-FEED-CLOSE, L-SESS-COM-CLOSE, L-SERV-COM-CLOSE, L-RES, L-SESS-RES, L-EXTR and L-REC are dealt with as in the proof of Lemma 10, with an extra step at the end (to take care of the possible extra τ steps) similar to the cases detailed above.

The only remaining cases are those when C is either a parallel composition or a stream composition, since now both subprocesses may be contexts.

- L-PAR: then C is $C_1 \mid C_2$, $C_1[P_1, \ldots, P_n] \xrightarrow{\alpha} P'$, and α and $C_2[P_1, \ldots, P_n]$ share no bounded names. By induction hypothesis there exists a process Q' such that $C_1[Q_1, \ldots, Q_n] \overset{\alpha}{\Rightarrow} Q'$, and P', Q' are respectively $C'_1[P'_1, \ldots, P'_m]$ and $C'_1[Q'_1, \ldots, Q'_m]$ for some multi-hole context C'_1 and processes $P'_1 \mathcal{S} Q'_1, \ldots, P'_m \mathcal{S} Q'_m$. By applying L-PAR to all steps of the sequence of transitions $C_1[Q_1, \ldots, Q_n] \xrightarrow{\tau} \ldots \xrightarrow{\alpha} \ldots \xrightarrow{\tau} Q'$, we conclude that

  $$C_1[Q_1, \ldots, Q_n] \mid C_2[Q_1, \ldots, Q_n] \overset{\alpha}{\Rightarrow} C'_1[Q'_1, \ldots, Q'_m] \mid C_2[Q_1, \ldots, Q_n],$$

  hence taking C' to be $C'_1 \mid C_2$ establishes the thesis[1].

- L-PAR': analogous (the roles of C_1 and C_2 are reversed).

- L-STREAM-PASS-P and L-STREAM-PASS-Q: as before, these are analogous to L-PAR and L-PAR', respectively.

- L-STREAM-FEED: C is $\textsf{stream}\ C_1\ \textsf{as}\ f = \vec{w}\ \textsf{in}\ C_2$ and $C_1[P_1, \ldots, P_n] \xrightarrow{\Uparrow v} P'$. By induction hypothesis, there exists Q' such that $C_1[Q_1, \ldots, Q_n] \overset{\Uparrow v}{\Longrightarrow} Q'$, and P' and Q' are respectively $C'_1[P'_1, \ldots, P'_n]$ and $C'_1[Q'_1, \ldots, Q'_n]$ for some n-ary multi-hole context C'_1 and processes $P'_1 \mathcal{S} Q'_1, \ldots, P'_n \mathcal{S} Q'_n$. In other words, $C_1[Q_1, \ldots, Q_n] \overset{\tau}{\Rightarrow} Q^* \xrightarrow{\Uparrow v} Q^{**} \overset{\tau}{\Rightarrow} Q'$; using L-STREAM-PASS-

[1] We assume that a n-hole context does not have to contain occurrences of all its holes, so in particular C_1 and C_2 are n-hole contexts in which some holes may not occur.

P for the τ transitions, we conclude that

$$\begin{aligned}
&\mathsf{stream}\ C_1[Q_1,\ldots,Q_n]\ \mathsf{as}\ f=\vec{w}\ \mathsf{in}\ C_2[Q_1,\ldots,Q_n]\\
&\quad\stackrel{\tau}{\Rightarrow}\ \mathsf{stream}\ Q^*\ \mathsf{as}\ f=\vec{w}\ \mathsf{in}\ C_2[Q_1,\ldots,Q_n]\\
&\quad\stackrel{\tau}{\rightarrow}\ \mathsf{stream}\ Q^{**}\ \mathsf{as}\ f=v::\vec{w}\ \mathsf{in}\ C_2[Q_1,\ldots,Q_n]\\
&\quad\stackrel{\tau}{\Rightarrow}\ \mathsf{stream}\ C_1'[Q_1',\ldots,Q_n']\ \mathsf{as}\ f=v::\vec{w}\ \mathsf{in}\ C_2[Q_1,\ldots,Q_n],
\end{aligned}$$

hence taking C' to be $\mathsf{stream}\ C_1'\ \mathsf{as}\ f=v::\vec{w}\ \mathsf{in}\ C_2$ establishes the thesis.

- L-STREAM-CONS: analogous (the roles of C_1 and C_2 are reversed). □

Theorem 5. *Weak bisimilarity is a non-input congruence.*

Proof. Straightforward consequence of Lemmas 11 and 12. □

6 Conclusions

We introduced strong and weak ground bisimulations for SSCC and proved that they enjoy several desirable properties. In particular, the Harmony Lemma holds, and both $\sim$ and $\approx$ are non-input congruences. Furthermore, the known counter-examples for π-calculus show that it is not reasonable to expect that they be preserved under substitution.

Acknowledgments

This work was partially sponsored by project FET-GC II IST-2005-16004 SENSORIA and by FCT and FEDER via LASIGE.

References

[1] H.P. Barendregt. *The Lambda Calculus*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, revised edition, 1984. Its syntax and semantics.

[2] M. Boreale, R. Bruni, L. Caires, R. de Nicola, I. Lanese, M. Loreti, F. Martins, U. Montanari, A. Ravara, D. Sangiorgi, V. Vasconcelos, and G. Zavattaro. SCC: a Service Centered Calculus. In M. Bravetti, M. Núñez, and G. Zavattaro, editors, *Proceedings of WS-FM 2006, 3rd International Workshop on Web Services and Formal Methods*, volume 4184 of *LNCS*, pages 38–57. Springer, 2006.

[3] I. Lanese, V. Vasconcelos, F. Martins, and A. Ravara. Disciplining orchestration and conversation in service-oriented computing. In *5th IEEE International Conference on Software Engineering and Formal Methods*, pages 305–314. IEEE Computer Society Press, 2007.

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