Two bisimilarities in ν-calculus *

Kohei Honda

Department of Computer Science
Keio University
3-14-1 Hiyoshi, Kohoku-ku, Yokohama, 223,
Japan

Abstract

We study two bisimilarities for ν-calculus, a small but expressive formalism with asynchronous communication primitive. Weak bisimilarities based on two different notions of observables are formulated and studied. Both bisimilarities are congruence relations, while asynchronous bisimilarity is proved to be more inclusive than the synchronous counterpart. Equivalence of two semantic theories in expressive power is studied via mutual embeddings of modal logics characterizing their respective bisimilarities. Apart from the interest in their own right, our results strongly relativize the notion of observables as a semantic foundation of concurrency. They also lay basis for operational study of ν-calculus.

1 Introduction

The aim of the present paper is two-fold. First we study basic properties of two bisimilarities coming from quite different notions of observables, for a single process calculus called ν-calculus [9, 10][1]. The calculus has asynchronous communication primitive and inherits capability to generate and pass names from its ancestor, π-calculus [13, 16, 17]. Two bisimilarities for the calculus, one based on classical synchronous observation and the other based on asynchronous observation, are shown to be equally consistent and expressive, while inducing significant difference in equating terms. With its simple construction, it is hoped that the notion of asynchronous observation and related construction would enrich the semantic theory of concurrency. It should also be noticed that our study in the present exposition strongly *relativises* the notion of observables. While this has already been pointed out by [1] in terms of the varied notions of testing, here the situation seems a bit more serious, for two different notions of observables lead us to two completely different transition relations. Thus our work also indicates the notion of observables may not be as absolute as might have been regarded so far, though we believe labelled transition

*October 9, 1992. Revised on March 31, 1993.

[1]The system is independently studied by Boudol [5].

systems and bisimilarities (and for that matter other equivalence theories) continue to provide important syntactic tools to study concurrency.

Another interest is to provide basic tools of investigation of ν-calculus, semantic and otherwise. If we regard asynchronous communication [6, 2] as the simplest possible interaction primitive, where synchronization machinery is reduced to its bare minimum, the calculus with its striking expressive power [9, 10], shows a promising sign as a basic formalism of concurrency. We intend that bisimilarities and corresponding logics studied in the present paper may be used as important tools for understanding various aspects of the calculus.

We organize this short paper as follows. Section 2 defines terms and reduction. Section 3 formulates a labeled transition system based on the classical (synchronous) notion of observation. Section 4 then gives another transition system whose underlying idea is "observation without synchronization". The basic properties of induced bisimilarity are studied, including the fact that it strictly contains the synchronous counterpart. Section 5 studies relationship between two semantic theories using modal logics corresponding to two bisimilarities. After the characterisation theorems, it is shown that each logic has a sound and complete embedding in another. This tells us that two theories are equally expressive at least in their representation in logics. Finally Section 6 concludes with some further issues.

All the proofs are omitted in the main sections but significant ones are found in the Appendix.

Note. Throughout the exposition, $\stackrel{def}{=}$ denotes definitional equality, $\langle x,\ y\rangle$ denotes a pair of x and y, and $\Rightarrow$ and $\Leftrightarrow$ denote logical implication and equivalence, respectively.

2 The Calculus

The following is a brief introduction to ν-calculus as far as needed in the subsequent discussions.

Syntax

ν-calculus, born from π-calculus [18, 13, 16], is a small formalism of concurrency with the notion of asynchronous communication as the interaction primitive [9, 10, 5]. The simple primitive, coupled with capability to generate and pass port names, gives ν-calculus enough power to construct any imaginable structures of interaction, just as the simple operation called *application* in λ-calculus gives it enough power to construct any imaginable applicative behaviour. In the present paper we use a simplified form of recursion (cf. [9]), which is suitable for our subsequent development.

Definitions 2.1 *Syntax. Let* $\boldsymbol{N}$ *be the set of infinite (port) names, ranged over by* $a, b, c, ..$ *or* $x, y, z, ...$ *The set of terms* $\boldsymbol{C}$*, ranged over by* $P, Q, R, ..,$ *is given by the fol-*

lowing grammar.

$$\begin{array}{rcll} P & ::= & \leftarrow av & \textit{(a message)} \\ & | & ax.P & \textit{(a receptor)} \\ & | & !\, ax.P & \textit{(a lazy replicator)} \\ & | & |x|P & \textit{(scope restriction)} \\ & | & P, Q & \textit{(concurrent composition)} \\ & | & \Lambda & \textit{(the null term)} \end{array}$$

where in $!\, ax.P$ *we let* $a \neq x$. ∎

Conventions 2.2

- *(i)* *The constructor "," is the weakest in association. "P, Q, R" denotes "$(P, Q), R$". We will often write* $|xyz|P$ *to mean* $|x|(|y|(|z|P))$*, and, specifically,* $|\varepsilon|P \stackrel{def}{=} P$*. In spite of these conventions, we will freely use parenthesis to be explicit about the syntactic structure,*
- *(ii)* $\tilde{a}, \tilde{b}, \tilde{c}, ..$ *range over* $\boldsymbol{N}^*$. ε *is the empty string.* $\tilde{w}\backslash v$ *denote a string of names where the occurrences of* v *are taken away from* $\tilde{w}$. $\{\tilde{w}\}$ *is a set of names from* $\tilde{w}$. $len(\tilde{w})$ *denotes the length of* $\tilde{w}$. *If* $\mathcal{F}$ *is the set of names,* $\overline{\mathcal{F}} \stackrel{def}{=} \boldsymbol{N}\backslash\mathcal{F}$.
- *(iii)* *Unless otherwise stated, different meta-variables for names denote different names, i.e. in* $\leftarrow av$ *we usually have* $a \neq v$. ∎

Examples 2.3

- *(i)* $\leftarrow av$. *A message which carries a name* v *to a receptor whose name is* a.
- *(ii)* $ax. \leftarrow ax$. *A receptor who, when receiving a message, creates the same message, and disappears.*
- *(iii)* Λ. *This is a convention to denote nothing.*
- *(iv)* $|x|(\leftarrow ax),\ \leftarrow ax$. *Here "x" in the right message is local to its scope, hence differs from "x" in the message on the left. Indeed the whole term is* α*-convertible to* $|y|(\leftarrow ay),\ \leftarrow ax$.
- *(v)* $!\, ax.P$. *This is a lazy replicator who will copy its body,* $ax.P$*, as required by a message.*[2] ∎

There are two kinds of bindings in the calculus.

Definitions 2.4 *Bindings.*

- *(i)* *In* $ax.P$*, the formal parameter* x*, occurring bound itself, binds free occurrences of the same name in its scope,* P.

[2] The lazy replicator simplifies recursive receptor in [9], but already appeared in [13]. These and Milner's eager replicator [16], or other versions, which would be studied in the subsequent paper, have essentially the same expressive power, and technical results presented herein hold using any one of them.

(ii) In $|v|P$, v of $|v|$, occurring bound, binds its free occurrences in its scope, P. ∎

Then we define various notions for classifying names which occur in a term.

Definitions 2.5

(i) The set of free names in P, *denoted by $\mathcal{FN}(P)$, is inductively given by:*
$\mathcal{FN}(ax.P) \stackrel{def}{=} \{a\} \cup \mathcal{FN}(P) \backslash \{x\}$, $\mathcal{FN}(!\,ax.P) \stackrel{def}{=} \mathcal{FN}(ax.P)$, $\mathcal{FN}(\leftarrow av) \stackrel{def}{=} \{a, v\}$,
$\mathcal{FN}(|x|P) \stackrel{def}{=} \mathcal{FN}(P) \backslash \{x\}$, $\mathcal{FN}(P, Q) \stackrel{def}{=} \mathcal{FN}(P) \cup \mathcal{FN}(Q)$, $\mathcal{FN}(\Lambda) \stackrel{def}{=} \emptyset$.

(ii) The set of bound names in P, *denoted by $\mathcal{BN}(P)$, is inductively given by:*
$\mathcal{BN}(ax.P) \stackrel{def}{=} \{x\} \cup \mathcal{BN}(P)$, $\mathcal{BN}(!\,ax.P) \stackrel{def}{=} \mathcal{BN}(ax.P)$, $\mathcal{BN}(\leftarrow av) \stackrel{def}{=} \emptyset$,
$\mathcal{BN}(|x|P) \stackrel{def}{=} \{x\} \cup \mathcal{BN}(P)$, $\mathcal{BN}(P, Q) \stackrel{def}{=} \mathcal{BN}(P) \cup \mathcal{BN}(Q)$, $\mathcal{BN}(\Lambda) \stackrel{def}{=} \emptyset$.

(iii) The set of names in P, *denoted by $\mathcal{N}(P)$, is given by $\mathcal{N}(P) \stackrel{def}{=} \mathcal{FN}(P) \cup \mathcal{BN}(P)$.*

(iv) The set of active input names in P, *denoted by $\mathcal{AN}_i(P)$, is inductively given by:*
$\mathcal{AN}_i(ax.P) \stackrel{def}{=} \{a\}$, $\mathcal{AN}_i(!\,ax.P) \stackrel{def}{=} \mathcal{AN}_i(ax.P)$, $\mathcal{AN}_i(\leftarrow av) \stackrel{def}{=} \emptyset$,
$\mathcal{AN}_i(|x|P) \stackrel{def}{=} \mathcal{AN}_i(P) \backslash \{x\}$, $\mathcal{AN}_i(P, Q) \stackrel{def}{=} \mathcal{AN}_i(P) \cup \mathcal{AN}_i(Q)$, $\mathcal{AN}_i(\Lambda) \stackrel{def}{=} \emptyset$.

(v) The set of active output names in P, *denoted by $\mathcal{AN}_o(P)$, is inductively given by:* $\mathcal{AN}_o(ax.P) \stackrel{def}{=} \emptyset$, $\mathcal{AN}_o(!\,ax.P) \stackrel{def}{=} \emptyset$, $\mathcal{AN}_o(\leftarrow av) \stackrel{def}{=} \{a\}$,
$\mathcal{AN}_o(|x|P) \stackrel{def}{=} \mathcal{AN}_o(P) \backslash \{x\}$, $\mathcal{AN}_o(P, Q) \stackrel{def}{=} \mathcal{AN}_o(P) \cup \mathcal{AN}_o(Q)$, $\mathcal{AN}_i(\Lambda) \stackrel{def}{=} \emptyset$.

(vi) $\mathcal{AN}(P) \stackrel{def}{=} \mathcal{AN}_i(P) \cup \mathcal{AN}_o(P)$. ∎

Then we have the notion of *substitution*. We assume a fresh name is always found uniquely (assume e.g. $\mathbf{N}$ is linearly ordered and take the least unused element).

Definitions 2.6 *Substitution and simultaneous substitution.*

(i) The substitution[3], *written $P[y/x]$ or $a[y/x]$, is defined by:*

$$
\begin{array}{rcll}
a[y/x] & \stackrel{def}{=} & y & (x = a) \\
a[y/x] & \stackrel{def}{=} & a & (x \neq a) \\
(\leftarrow av)[y/x] & \stackrel{def}{=} & \leftarrow(a[y/x])(v[y/x]) & \\
(az.P)[y/x] & \stackrel{def}{=} & (a[y/x])z.(P[y/x]) & (x \neq z \wedge x \neq y) \\
(ax.P)[y/x] & \stackrel{def}{=} & (a[y/x])x.P & \\
(ay.P)[y/x] & \stackrel{def}{=} & (a[y/x])z.(P[z/y][y/x]) & (z \textit{ fresh}) \\
(!\,az.P)[y/x] & \stackrel{def}{=} & !\,((az.P)[y/x]) & \\
(P, Q)[y/x] & \stackrel{def}{=} & (P[y/x]),\ (Q[y/x]) & \\
(|z|P)[y/x] & \stackrel{def}{=} & |z|(P[y/x]) & (x \neq z \wedge x \neq y) \\
(|x|P)[y/x] & \stackrel{def}{=} & |x|P & \\
(|y|P)[y/x] & \stackrel{def}{=} & |z|(P[z/y][y/x]) & (z \textit{ fresh})
\end{array}
$$

[3]The operation is of type $(\mathbf{C} \cup \mathbf{N}) \times \mathbf{N} \times \mathbf{N} \to (\mathbf{C} \cup \mathbf{N})$.

(ii) The simultaneous substitution[4], *written* $P\{\tilde{y}/\tilde{x}\}$, *which is defined only when names in* $\tilde{x}$ *are all distinct and* $len(\tilde{x}) = len(\tilde{y})$, *is given by:*

$$\begin{aligned} P\{\varepsilon/\varepsilon\} &\stackrel{def}{=} P \\ P\{y\tilde{v}/x\tilde{w}\} &\stackrel{def}{=} P[z/x]\{\tilde{v}/\tilde{w}\}[y/z] \quad (z \text{ fresh}) \quad \blacksquare \end{aligned}$$

Conventions 2.7 *We assume the constructs* $[\cdot/\cdot]$ *and* $\{\cdot/\cdot\}$ *associate relatively stronger than other syntactic constructors, e.g.* $ax.P[v/w] \stackrel{def}{=} ax.(P[v/w])$ *etc. Also we assume, whenever we write* $P\{\tilde{v}/\tilde{x}\}$, *it is defined, i.e. names in* $\tilde{x}$ *are all distinct and* $len(\tilde{x}) = len(\tilde{v})$. ∎

Now we define α-convertibility among terms.

Definitions 2.8 *We define* $\equiv_\alpha$ *as the smallest relation generated by:*

(i) $|x|P \equiv_\alpha |z|(P[z/x])$ *if* $z \notin \mathcal{FN}(P)$.

(ii) $ax.P \equiv_\alpha az.P[z/x]$ *if* $z \notin \mathcal{FN}(P)$.

(iii) $\equiv_\alpha$ *is a congruence relation.* ∎

To avoid technical nuisances the following convention is useful (cf. footnote 5).

Conventions 2.9 We regard two α-convertible terms as identical, incorporating the α-convertibility into the definitional equality. ∎

Let us list some basic facts about substitution (cf. Chapter 1 of [8]). To show (vi), which may be new, one needs (ii), (iii) and (i) (in this order).

Lemma 2.10 In (iv) and (v) let x and v be distinct.

(i) $P[x/x] \stackrel{def}{=} P$

(ii) $z \notin \mathcal{FN}(P) \quad \Rightarrow \quad P[w/z] \stackrel{def}{=} P$

(iii) $z \notin \mathcal{FN}(P) \quad \Rightarrow \quad P[z/x][y/z] \stackrel{def}{=} P[y/x]$

(iv) $P[w/x][v/y] \stackrel{def}{=} P[v/y][(w[v/y])/x]$

(v) $P[v/x][w/x] \stackrel{def}{=} P[v/x]$

(vi) $z \notin \mathcal{FN}(P) \cup \mathcal{FN}(Q) \quad \Rightarrow \quad (P[v/x],\ Q) \stackrel{def}{=} (P,\ Q[z/x])[v/x][x/z]$.

(vii) $P\{v/x\} \stackrel{def}{=} P[v/x]$.

(viii) With $len(\tilde{w}_1) = len(\tilde{y}_1)$, $P\{\tilde{w}_1 v_1 v_2 \tilde{w}_2/\tilde{y}_1 x_1 x_2 \tilde{y}_2\} \stackrel{def}{=} P\{\tilde{w}_1 v_2 v_1 \tilde{w}_2/\tilde{y}_1 x_2 x_1 \tilde{y}_2\}$. ∎

Also important regarding the substitution is the notion of a *name variant*. If a term can be transformed to another term by bijective simultaneous substitution, we say the latter is a name variant of the former. Better:

[4] Of type $\mathbf{C} \times \mathbf{N}^* \times \mathbf{N}^* \to \mathbf{C}$.

Definitions 2.11 *Name variant.* $P \doteq Q$, *read* P *is a name variant of* Q, *is defined by:*

(i) $P \doteq Q$ *if* $Q \stackrel{def}{=} P[v/x]$ *with* $v \notin \mathcal{FN}(P)$.
(ii) $\doteq$ *is transitive.* ∎

Note that $\doteq$ is an equivalence relation (for reflexivity use (ii) of Lemma 2.10, while for symmetry one can use (iii) of the same lemma by letting $y = x$, together with transitivity of $\doteq$). When discussing multi-step transition relation, the notion becomes essential.

Structural Equivalence

The following structural rules, following Milner [16], define equivalence over syntactic representation. For port-passing calculi, structural rules play an essential role for tractable formulation of transition rules. The set of rules below is weaker than Milner's construction, but enough to induce required reduction and other relation. Note we already equate α-convertible terms by Convention 2.9[5].

Definitions 2.12 Structural equivalence. $\equiv$ *is the smallest relation over terms defined by:*

(i) $(P, Q), R \equiv P, (Q, R)$
(ii) $|x|P, Q \equiv |x|(P, Q) \qquad (x \notin \mathcal{FN}(Q))$
(iii) $P, Q \equiv Q, P$
(iv) $!\,ax.P \equiv ax.(P, !\,ax.P)$
(v) $P \equiv Q$ *then* $P, R \equiv Q, R$ *and* $|x|P \equiv |x|Q$.
(vi) $\equiv$ *is an equivalence relation.* ∎

Fact 2.13 $\{P \mid P \equiv P'\}$ *is finite for any* P' *(modulo* α*-convertibility).* ∎

Note that $\equiv$ preserves basic properties of terms, i.e.

Lemma 2.14 For any P, Q, x, v,

(i) $P \equiv Q \;\Rightarrow\; \mathcal{FN}(P) = \mathcal{FN}(Q) \wedge \mathcal{AN}_i(P) = \mathcal{AN}_i(Q) \wedge \mathcal{AN}_o(P) = \mathcal{AN}_o(Q)$.
(ii) $P \equiv Q \;\Rightarrow\; P[v/x] \equiv Q[v/x]$. ∎

Another fact is that we can always transform a term into a certain normal form.

[5] Without the convention we could have incorporated $\equiv_\alpha$ explicitly in $\equiv$: since reduction rules and various transition rules later are defined incorporating $\equiv$, it is easy to verify these rules with and without Convention 2.9, are essentially equivalent.

Proposition 2.15 Let us call terms of the forms $\leftarrow av$, $ax.P$, and Λ, *base terms*. Then, for any $P \in \mathbf{C}$, we have $P' \equiv P$ such that, for some $m \geq 0$,

$$P' \stackrel{def}{=} |\tilde{w}|(P_1, P_2, ..., P_m)$$

where $P_1, ..P_m$ denote base terms. We call such P' (usually not unique) a *normal form* of P. Above we say $P_1, ..P_m$ are *active* in P'. ∎

In the proposition above, we can write $P_1, P_2, ...$ etc. without any ambiguity (up to $\equiv$), due to the rule (i) of Definition 2.12. Also note that, with P in a normal form, $\mathcal{AN}_i(P)$ is by definition the set of free input port names of active receptors in P.

In reasoning about terms in normal form, the following notation is convenient.

Conventions 2.16 *∂, ∂', etc. range over a sequence of concurrent composition of base terms.* ∎

Reduction

Reduction relation defines the fundamental mechanism of computing in the calculus. The underlying idea is a quite primitive notion of interaction where a piece of information is simply consumed by a computing entity, which in turn generates a set of new terms after substitution.

Definitions 2.17 *Reduction.*

(i) One-step reduction, or simply reduction, denoted $\longrightarrow$, is the smallest relation over terms generated by the following rules.

$$COM: \qquad |\tilde{w}|(\partial, \leftarrow av, ax.P, \partial') \longrightarrow |\tilde{w}|(\partial, P[v/x], \partial')$$

$$STRUCT: \qquad \frac{P_1' \equiv P_1, \ P_1 \longrightarrow P_2, \ P_2 \equiv P_2'}{P_1' \longrightarrow P_2'}$$

where in COM we call $\leftarrow av, ax.P$ a *redex pair* or simply a redex.

(ii) Multi step reduction $\longrightarrow\!\!\!\!\rightarrow$ is defined: $\longrightarrow\!\!\!\!\rightarrow \stackrel{def}{=} (\equiv \cup \longrightarrow^*)$. ∎

The reduction relation itself is not the focus of our study in the present exposition, but it is worth emphasizing that the core of the calculus consists of its syntax and related basic notions and reduction relation, coupled with structural rules, and labelled transition relation later defined should be regarded useful but auxiliary extension of the calculus. Since $\equiv$ includes $\equiv_\alpha$, our identification of α-convertible terms is sound with respect to the definition of $\longrightarrow$. Some examples of reduction are listed below.

Examples 2.18 *Reduction.*

(i) A simple reduction.

$$\leftarrow av, \ ax. \leftarrow ax \ \longrightarrow \leftarrow av$$

(ii) Replication.

$$\leftarrow av,\ \leftarrow aw,\ !\,ax.P \ \longrightarrow\ P[v/x],\ \leftarrow aw,\ !\,ax.P \ \longrightarrow\ P[v/x],\ P[w/x],\ !\,ax.P$$

(iii) Scope restriction and reduction.

$$|c|(\leftarrow ac,\ Q),\ ax.P \longrightarrow |c|(Q, P[c/x]) \quad (c \notin \mathcal{FN}(ax.P)) \qquad \blacksquare$$

Some important expressions to be used later are defined below.

Definitions 2.19 *Agents and expressions.*

(i) $\lambda x.P \odot v \stackrel{def}{=} |c|(cx.P,\ \leftarrow cv)$ *(c fresh).*

(ii) $P \oplus Q \stackrel{def}{=} |c|(\leftarrow c,\ c.P,\ c.Q)$. *(c fresh).*

(iii) $\mathcal{I}_0(a) \stackrel{def}{=} ay. \leftarrow ay.$ $(a \neq y)$.

(iv) $\mathcal{I}_1(a) \stackrel{def}{=} !\,ay. \leftarrow ay.$ *Sometimes we write* $\mathcal{I}(a)$ *for* $\mathcal{I}_1(a)$.

(v) $\mathcal{I}_2(a_1..a_n) \stackrel{def}{=} |c|(!cx.xy.(\leftarrow xy, \leftarrow cx \leftarrow cy),\ \leftarrow ca_1, .., \leftarrow ca_n)$ $(c, x, y$ *all fresh).*

(vi) $\Omega \stackrel{def}{=} |o|(\leftarrow oo,\ \mathcal{I}(o))$.

(vii) $\Omega_2 \stackrel{def}{=} |o|(\leftarrow oo,\ \mathcal{I}_2(oo..o))$. $(\Omega_2$ *is any term in this form).* $\blacksquare$

Note (i) is almost like $P[v/x]$ while (ii) behaves *either* as P *or* Q. As to (iii) to (vi), they all represent terms which does not give effect to outside (except useless reduction). Indeed we have:

Proposition 2.20 Let $\mathbf{O} \stackrel{def}{=} \{P \mid P_0 \longrightarrow\!\!\!\!\!\rightarrow P,\ P_0$ is either $\Lambda, \Omega, \Omega_2, \mathcal{I}_0(a), \mathcal{I}(a),$ or $\mathcal{I}_2(\tilde{a}).\ \}$. Then:

$$(P,\ Q) \ \longrightarrow\!\!\!\!\!\rightarrow\ R \ \wedge\ Q \in \mathbf{O} \quad \Rightarrow \quad R \equiv (P', Q') \ \wedge\ P \longrightarrow\!\!\!\!\!\rightarrow P' \ \wedge\ Q' \in \mathbf{O}. \qquad \blacksquare$$

Specifically we call $\mathcal{I}_0(a)$, $\mathcal{I}(a)$, and $\mathcal{I}_2(\tilde{a})$, *identity receptors.* Agents in Proposition 2.20 which are not identity receptors, e.g. Ω_2, have the following property, which makes satisfaction of the proposition immediate:

$$\Omega_2 \longrightarrow\!\!\!\!\!\rightarrow P \quad \Rightarrow \quad \mathcal{AN}(P) = \emptyset\,.$$

In contrast identity receptors *do interact with terms in the outside*: however in effect they do not give any influence. This property of identity receptors will turn out to be essential in measuring difference between two bisimilarities we will study hereafter.

3 Synchronous Transition and Bisimilarity

This section formulates a classical version of labelled transition system and weak bisimilarity based on the transition relation.

Synchronous Transition

The notion of observation in the foregoing process calculi is based on synchronous communication: one observes the system's behaviour by synchronously interacting with it. We carry out similar construction for ν-calculus. The labelled transition relation then becomes the classical "synchronous" transition system naturally restricted to our syntactic construction [6]. The underlying concept is if and only if the outside observer can have interaction with the term, the interaction is counted as *observable actions*.

We first introduce *labels* for the transition.

Definitions 3.1 *Labels. The set of labels* $\boldsymbol{L}$*, ranged over by* $l, l', ..,$ *is given by the following grammar.*

$$l \;=\; \tau \mid \downarrow ax \mid \downarrow a(x) \mid \uparrow ax \mid \uparrow a(x)\ . \qquad \blacksquare$$

Some related definitions follow.

Definitions 3.2

(i) *We define the sets of free names, bound names, and names of a label as follows.* $\mathcal{FN}(\downarrow av) = \mathcal{FN}(\uparrow av) = \{a, v\}$, $\mathcal{FN}(\downarrow a(v)) = \mathcal{FN}(\uparrow a(v)) = \{a\}$, $\mathcal{FN}(\tau) = \tau$, $\mathcal{BN}(\downarrow av) = \mathcal{BN}(\uparrow av) = \mathcal{BN}(\tau) = \emptyset$, $\mathcal{BN}(\downarrow a(v)) = \mathcal{BN}(\uparrow a(v)) = \{v\}$, *and* $\mathcal{N}(l) = \mathcal{FN}(l) \cup \mathcal{BN}(l)$.

(ii) $l[v/x]$ *and* $l\{\tilde{v}/\tilde{x}\}$*, are defined as in the case of terms, without affecting the bound occurrence of names (if any), e.g.* $\downarrow a(w)[v/x] \stackrel{def}{=} \downarrow a[v/x](w)$*. Then we define* α*-convertibility among labels which is again an equivalence relation, and denote the relation by* $\equiv_\alpha$*, e.g.* $\downarrow a(x) \equiv_\alpha \downarrow a(y)$*. But* we may not identify α-convertible labels: *see Lemma 3.8.*

(iii) *In* $\downarrow av, \downarrow a(v), \uparrow av,$ *and* $\uparrow a(v)$*,* a *is called its subject and* v *is called its object[13]. Functions are defined to extract them from a label:* $\pi_1(\uparrow av) = \pi_1(\uparrow a(v)) = \pi_1(\downarrow av) = \pi_1(\downarrow a(v)) = a$*,* $\pi_2(\uparrow av) = \pi_2(\uparrow a(v)) = \pi_2(\downarrow av) = \pi_2(\downarrow a(v)) = v$*, while* $\pi_1(\tau)$ *and* $\pi_2(\tau)$ *is undefined.* $\blacksquare$

The following comments on the nature of labels, being not formal, would still be useful to have an intuitive grasp of the underlying notions. Our interpretation of $\downarrow a(v)$ may be new (cf. Remark 5.21).

(1) τ denotes internal computation, i.e. communication occurring inside the configuration, thus unseen from the outside.

(2) $\downarrow av$ (free input) and $\uparrow av$ (free output) denote incoming and outgoing messages which carry free names, respectively. Here "freeness" means carried names *might* be known to the target of the message.

[6] The issue raised by [13, 14] concerning *late* and *early* instantiation schemes will not be treated here. Our subsequent work will explicate the issue, but we have a strong reason to select the present version of "early" scheme.

(3) $\downarrow a(v)$ (bound input) and $\uparrow a(v)$ (bound output) are then messages carrying bound names, i.e. names which *cannot* be known to the target of the message.

Now we formulate the labelled transition system which defines transition relation called *synchronous transition.*

Definitions 3.3 *Synchronous transition.* Synchronous (one-step) transition, denoted $\xrightarrow{l}_s$, is the smallest relation generated by:

$$IN(f)_s : \quad |\tilde{w}|(\partial, ax.P, \partial') \xrightarrow{\downarrow av}_s |\tilde{w}|(\partial, P[v/x], \partial') \qquad {\scriptstyle (a,v \notin \{\tilde{w}\})}$$

$$IN(b)_s : \quad |\tilde{w}|(\partial, ax.P, \partial') \xrightarrow{\downarrow a(v)}_s |\tilde{w}|(\partial, P[v/x], \partial') \qquad {\scriptstyle (a \notin \{\tilde{w}\} \ \wedge \ v \notin \mathcal{N}(|\tilde{w}|(\partial, ax.P, \partial')))}$$

$$OUT(f)_s : \quad |\tilde{w}|(\partial, \leftarrow av, \partial') \xrightarrow{\uparrow av}_s |\tilde{w}|(\partial, \partial') \qquad {\scriptstyle (a,v \notin \{\tilde{w}\})}$$

$$OUT(b)_s : \quad |\tilde{w}|(\partial, \leftarrow av, \partial') \xrightarrow{\uparrow a(v)}_s |\tilde{w} \backslash v|(\partial, \partial') \qquad {\scriptstyle (a \notin \{\tilde{w}\} \ \wedge \ v \in \{\tilde{w}\})}$$

$$COM_s : \quad |\tilde{w}|(\partial, \leftarrow av, ax.P, \partial') \xrightarrow{\tau}_s |\tilde{w}|(\partial, P[v/x], \partial')$$

$$STRUCT_s : \quad \frac{P_1' \equiv P_1, \ P_1 \xrightarrow{l}_s P_2, \ P_2 \equiv P_2'}{P_1' \xrightarrow{l}_s P_2'}$$

where in COM we call $\leftarrow av, ax.P$ a *redex pair* or simply a redex. ∎

Note $\xrightarrow{\tau}_s = \longrightarrow$. Also note that the synchronous "input" observation can only proceed when some receptors are ready to receive them. Including this, we have the following.

Proposition 3.4 Let us write $P \xrightarrow{l}_s$ when $P \xrightarrow{l}_s P'$ for some P'. Then

(i) $P \xrightarrow{\downarrow av}_s \quad \Leftrightarrow \quad a \in \mathcal{AN}_i(P)$.

(ii) $P \xrightarrow{l}_s \wedge\, l \neq \tau \quad \Leftrightarrow \quad \pi_1(l) \in \mathcal{AN}(P)$. ∎

Remarks 3.5 *Concerning transitions with bound labels, note:*

(i) $P \xrightarrow{\downarrow a(v)}_s P' \quad \Leftrightarrow \quad P \xrightarrow{\downarrow av}_s P'$ *with* $v \notin \mathcal{FN}(P)$.

(ii) $P \xrightarrow{\uparrow a(v)}_s P' \quad \Rightarrow \quad$ *not* $P \xrightarrow{\uparrow av}_s$

where (i) tells us that bound input can be induced from the free input by inspecting name occurrences, while the second says that we can first merge free output and bound output then induce the latter from the merged transition. Thus transition with bound labels are redundant as far as one-step transition is concerned, though their technical convenience is notable, especially when we deal with multi-step transitions. ∎

Several facts on $\xrightarrow{l}_s$ (cf. Lemmas 1-3 in [13]) will be useful for our later discussions. The first lemma is about relationship between transition and free names in a term.

Lemma 3.6 *Free name lemma for* $\xrightarrow{l}_s$.

$$\begin{array}{ll}
(1) & P \xrightarrow{\downarrow av}_s P' \vee P \xrightarrow{\downarrow a(v)}_s P' \quad \Rightarrow \quad \mathcal{FN}(P) \cup \{v\} = \mathcal{FN}(P') \cup \{a\} \\
(2) & P \xrightarrow{\uparrow av}_s P' \quad \Rightarrow \quad \mathcal{FN}(P) = \mathcal{FN}(P') \cup \{a, v\} \\
(3) & P \xrightarrow{\uparrow a(v)}_s P' \quad \Rightarrow \quad \mathcal{FN}(P) \cup \{v\} = \mathcal{FN}(P') \cup \{a, v\} \\
(4) & P \xrightarrow{\tau}_s P' \quad \Rightarrow \quad \mathcal{FN}(P) \supset \mathcal{FN}(P') \qquad \blacksquare
\end{array}$$

Before the second lemma we need a small definition.

Definitions 3.7 *We define the* substitution induced by α-conversion, $\{l \backslash\backslash l'\}$ *for* $l' \equiv_\alpha l$, *as follows.*

$$\begin{array}{ll}
(i) & l \equiv_\alpha l' \wedge \mathcal{BN}(l) = \emptyset \quad \Rightarrow \quad P\{l \backslash\backslash l'\} \stackrel{def}{=} P \;. \\
(ii) & l \equiv_\alpha l' \wedge \mathcal{BN}(l) \neq \emptyset \quad \Rightarrow \quad P\{l \backslash\backslash l'\} \stackrel{def}{=} P[\pi_2(l')/\pi_2(l)] \;. \\
(iii) & l \not\equiv_\alpha l' \quad \Rightarrow \quad P\{l \backslash\backslash l'\} \textit{ is undefined.} \qquad \blacksquare
\end{array}$$

$P\{l \backslash\backslash l'\}$ reads "P after the substitution induced by converting l to l'". Using the operation we state the following lemma. It tells us that, once we have a bound input (resp. a bound output) for *some* bound name, *any* name can occur bound to induce essentially the same transition, as far as it does not collide with free names in the initial term.

Lemma 3.8 α-conversion in labels. *Suppose* $\mathcal{BN}(l) \neq \emptyset$ *below.*

$$P \xrightarrow{l}_s Q \;\wedge\; l \equiv_\alpha l' \;\wedge\; \mathcal{BN}(l') \not\subset \mathcal{FN}(P) \quad \Rightarrow \quad P \xrightarrow{l'}_s Q\{l \backslash\backslash l'\} \doteq Q \;. \qquad \blacksquare$$

The third lemma is about relationship between substitution and transition.

Lemma 3.9 *Substitution lemma for* $\xrightarrow{l}_s$.

(i) $P \xrightarrow{l}_s P' \;\wedge\; \mathcal{BN}(l) = \emptyset \quad \Rightarrow \quad P\{\tilde{v}/\tilde{x}\} \xrightarrow{l\{\tilde{v}/\tilde{x}\}}_s P'\{\tilde{v}/\tilde{x}\}$.

(ii) $P \xrightarrow{l}_s P' \wedge \mathcal{BN}(l) \neq \emptyset \wedge \mathcal{BN}(l) \cap \{\tilde{v}\} = \emptyset \quad \Rightarrow \quad P\{\tilde{v}/\tilde{x}\} \xrightarrow{l\{\tilde{v}/\tilde{x}\}}_s P'[z/\pi_2(l)]\{\tilde{v}/\tilde{x}\}[\pi_2(l)/z]$
with z fresh. $\blacksquare$

Note in (ii) if $\mathcal{BN}(l) \cap \{\tilde{x}\} = \emptyset$ in addition, the right hand-side becomes

$$P\{\tilde{v}/\tilde{x}\} \xrightarrow{l\{\tilde{v}/\tilde{x}\}}_s P'\{\tilde{v}/\tilde{x}\}$$

which is much simpler.

Weak Synchronous Transition

Weak transitions are those which ignore τ actions. Semantically the relation may be regarded as more important than one-step transition. We first define some syntactic notions. The third notion, *regularity*, is important for our subsequent development. If t is regular a fresh name is always used for its binding occurrences.

Definitions 3.10

(i) $t, t', ..$ *range over* $(\boldsymbol{L}\backslash\{\tau\})^*$. ε *denotes the empty string. Concatenation is denoted by* $\cdot$, *e.g.* $t \cdot t'$, *or when there is no ambiguity, just by juxtaposition, e.g.* tt'.

(ii) The binding is defined as: e.g. in $t\cdot \downarrow a(x) \cdot t'$ *the binding occurrence "x" in the middle binds free occurrences of x in t'. Then* $\mathcal{FN}(t), \mathcal{BN}(t)$, *and* $\mathcal{N}(t)$ *are the sets of free names, bound names, and their union occurring in t, respectively.* $t[v/x]$ *are defined as in terms, avoiding the capture of names. Then we write* $t \equiv_\alpha t'$ *to denote two sequences are α-convertible. For example,*
$\downarrow a(x)\cdot \uparrow xv\cdot \downarrow bv\cdot \uparrow x(y)\cdot \uparrow yx \ \equiv_\alpha \downarrow a(z)\cdot \uparrow zv\cdot \downarrow bv\cdot \uparrow z(w)\cdot \uparrow wz$ *etc.*

(iii) We say t is regular *if the following holds:* $t \stackrel{def}{=} t_1\cdot t_2 \quad\Rightarrow\quad \mathcal{N}(t_1)\cap\mathcal{BN}(t_2) = \emptyset$
∎

We also have an operation analogous to that defined in Definition 3.7.

Definitions 3.11 Substitution induced by the α-conversion on label sequences, $\{t\backslash\backslash t'\}$ *for* $t \equiv_\alpha t'$, *is defined by:*

(i) $P\{\varepsilon\backslash\backslash\varepsilon\} \stackrel{def}{=} P$

(ii) $t \cdot l \ \equiv_\alpha \ t' \cdot l' \wedge \mathcal{BN}(l) = \emptyset \quad\Rightarrow\quad P\{(t\cdot l)\backslash\backslash(t'\cdot l')\} \stackrel{def}{=} P\{t\backslash\backslash t'\}$.

(iii) $t\cdot l \ \equiv_\alpha \ t'\cdot l' \wedge \mathcal{BN}(l) \neq \emptyset \quad\Rightarrow\quad P\{(t\cdot l)\backslash\backslash(t'\cdot l')\} \stackrel{def}{=} P[z/\pi_2(l)]\{t\backslash\backslash t'\}[\pi_2(l')/z]$ *(z fresh).*

(iv) $t \not\equiv_\alpha t' \quad\Rightarrow\quad P\{t\backslash\backslash t'\}$ *is undefined.* ∎

Now we are ready to define the weak transition relation. Its element is in the form $(P,\ t,\ Q)$ written as $P \stackrel{t}{\Longrightarrow}_s Q$.

Definitions 3.12 Weak synchronous transitions. *We inductively define the weak synchronous transition, denoted by* $\stackrel{t}{\Longrightarrow}_s$, *as the smallest relation generated by the following rules.*

$$\frac{P \stackrel{l}{\longrightarrow}_s P'}{P \stackrel{l}{\Longrightarrow}_s P'}\ (l \neq \tau) \qquad\qquad \frac{P \stackrel{\tau}{\longrightarrow}_s P'}{P \stackrel{\varepsilon}{\Longrightarrow}_s P'}$$

$$\frac{P \stackrel{t}{\Longrightarrow}_s P' \qquad P' \stackrel{t'}{\Longrightarrow}_s P''}{P \stackrel{t\cdot t'}{\Longrightarrow}_s P''}. \qquad ∎$$

Basic lemmas on transition relation follow.

Lemma 3.13 *α-conversion on weak transition. Suppose t' is regular with $t \equiv_\alpha t'$, and also $\mathcal{BN}(t') \cap (\mathcal{FN}(P) \cup \mathcal{N}(t)) = \emptyset$. Then we have*

$$P \overset{t}{\Longrightarrow}_s Q \quad \Rightarrow \quad P \overset{t'}{\Longrightarrow}_s Q\{t\backslash\backslash t'\} \doteq Q \qquad \blacksquare$$

Lemma 3.14 *Substitution lemma for $\overset{t}{\Longrightarrow}_s$.*

(i) $P \overset{\varepsilon}{\Longrightarrow}_s P' \quad \Rightarrow \quad P\{\tilde{v}/\tilde{x}\} \overset{\varepsilon}{\Longrightarrow}_s P'\{\tilde{v}/\tilde{x}\}$

(ii) $P \overset{l}{\Longrightarrow}_s P' \wedge \mathcal{BN}(l) \neq \emptyset \wedge \mathcal{BN}(l) \cap \{\tilde{v}\} = \emptyset \quad \Rightarrow \quad P\{\tilde{v}/\tilde{x}\} \overset{l\{\tilde{v}/\tilde{x}\}}{\Longrightarrow}_s P'[z/\pi_2(l)]\{\tilde{v}/\tilde{x}\}[\pi_2(l)/z]$

(iii) In general, $P \overset{t}{\Longrightarrow}_s P' \wedge \{\tilde{x}\tilde{v}\} \cap \mathcal{BN}(t) = \emptyset \quad \Rightarrow \quad P\{\tilde{v}/\tilde{x}\} \overset{t\{\tilde{v}/\tilde{x}\}}{\Longrightarrow}_s P'\{\tilde{v}/\tilde{x}\} \quad \blacksquare$

Before defining bisimilarity, we motivate its introduction a little (cf. [3]): the following shows that mere traces are not adequate even in the asynchronous setting.

Definitions 3.15 Traces. *The set of (weak)* synchronous traces *of a term P, denoted by $tr_s\langle P\rangle$, is given as follows.*

$$tr_s\langle P\rangle \overset{def}{=} \{t \mid \exists P' \ P \overset{t}{\Longrightarrow}_s P'\}. \qquad \blacksquare$$

Easily we have:

Proposition 3.16 $tr_s\langle P \oplus Q\rangle \overset{def}{=} tr_s\langle P\rangle \cup tr_s\langle Q\rangle. \quad \blacksquare$

Therefore:

Fact 3.17 $tr_s\langle \leftarrow av \oplus (\leftarrow av, \leftarrow bw)\rangle = tr_s\langle \leftarrow av, \leftarrow bw\rangle. \quad \blacksquare$

But they should be differentiated since, put in a context

$$C[\cdot] \overset{def}{=} |ab|(ax.(by. \leftarrow xy), \ [\cdot]),$$

one may or may not emit $\leftarrow vw$ while the other surely does so. Thus the information about *branching* [12] is essential here.

Weak Synchronous Bisimilarity

Now we formulate weak synchronous bisimilarity after a small definition. We first need the notion of soundness.

Definitions 3.18 t is sound with respect to P and Q if and only if $\mathcal{BN}(t) \cap (\mathcal{FN}(P) \cup \mathcal{FN}(Q)) = \emptyset. \quad \blacksquare$

The main definition follows.

Definitions 3.19 *Synchronous bisimulation.* P_1 and Q_1 are synchronously bisimilar, denoted by $P_1 \approx_s Q_1$, if and only if $\langle P_1, \ Q_1\rangle \in \mathcal{R}$ with

(i) For any $\langle P,\ Q\rangle \in \mathcal{R}$, whenever $P \overset{t}{\Longrightarrow}_s P'$ where t is regular and also is sound with respect to P and Q, there exists Q' such that $Q \overset{t}{\Longrightarrow}_s Q'$ and $\langle P',\ Q'\rangle \in \mathcal{R}$.

(ii) $\mathcal{R}$ is symmetric[7]. ∎

We call $\mathcal{R}$ as above a *synchronous bisimulation.* Then $\approx_s$ (the *synchronous bisimilarity*) is the union of all the bisimulations. First we recover a usual definition of bisimulation (cf. [12]) which is often convenient for reasoning about the bisimilarity. The proof needs some care about soundness condition (see Appendix).

Definitions 3.20 *We define $\hat{l}$ of type $\boldsymbol{L} \to (\boldsymbol{L}\backslash\{\tau\})^*$ as follows.*

$$\begin{array}{lcll} \hat{l} & \stackrel{def}{=} & l & (l \neq \tau) \\ \hat{\tau} & \stackrel{def}{=} & \varepsilon & \end{array}$$

Note, in the first clause, l on the right hand side is l as a string. ∎

Proposition 3.21 Define $P_1 \approx_s' Q_1$ when $\langle P_1,\ Q_1\rangle \in \mathcal{R}$ with

(i) For any $\langle P,\ Q\rangle \in \mathcal{R}$, whenever $P \overset{l}{\longrightarrow}_s P'$ where l is sound with respect to P and Q, there exists Q' such that $Q \overset{\hat{l}}{\Longrightarrow}_s Q'$ and $\langle P',\ Q'\rangle \in \mathcal{R}$.

(ii) $\mathcal{R}$ is symmetric.

Then $P \approx_s' Q$ if and only if $P \approx_s Q$. ∎

We will often be using this characterisation instead of the "genuine" definition of the synchronous bisimilarity.

Now as is easily seen, $\approx_s$ strictly extends $\equiv$.

Proposition 3.22

(i) $P \equiv Q \quad \Rightarrow \quad P \approx_s Q$.

(ii) $P \approx_s Q$ *does not imply* $P \equiv Q$. ∎

In addition there are several interesting (in)equations given by $\approx_s$.

Examples 3.23 *Some (in)equations by $\approx_s$.*

(i) $P, \Lambda \ \approx_s \ P$.

(ii) $|c|(cx.P) \ \approx_s \ |c|(\leftarrow cx) \approx_s \Lambda$.

(iii) $\lambda x.P \odot v \approx_s P[v/x]$.

[7]The requirement for symmetry (cf. [19]) can be weakened as shown in [16], but no difference comes about in the bisimilarity itself.

(iv) $P \oplus P \;\approx_s\; P, \quad P \oplus Q \;\approx_s\; Q \oplus P, \quad$ *but* $\quad P \oplus (Q \oplus R) \;\not\approx_s\; (P \oplus Q) \oplus R.$

(v) *Not necessarily* $P \oplus Q \;\approx_s\; P$, *e.g.* $\leftarrow av \oplus \leftarrow bw \not\approx_s \leftarrow av.$

(vi) $\Omega \;\approx_s\; \Lambda.$

(vii) $\Omega_2 \;\approx_s\; \Lambda.$ ∎

In terms of the last two equations, we note:

Proposition 3.24 $P \;\approx_s\; \Lambda \quad \Leftrightarrow \quad \forall P'.(P \longrightarrow\!\!\!\!\rightarrow P' \quad \Rightarrow \quad \mathcal{AN}(P') = \emptyset).$ ∎

The proposition has an important corollary.

Corollary 3.25 *None of* $\mathcal{I}_0(a)$, $\mathcal{I}_1(a)$, *and* $\mathcal{I}_2(a)$ *is synchronously bisimilar with* Λ. ∎

which shows some discrepancy between $\approx_s$ *and the behaviour of identity receptors as exemplified by Proposition 2.20.*

Remarks 3.26 *We note that we can have a better characterisation result than Proposition 3.21. Define* $\approx''_s$ *as* $P_1 \approx''_s Q_1$ *if and only if* $\langle P_1,\ Q_1\rangle \in \mathcal{R}$ *with*

(i) *For any* $\langle P,\ Q\rangle \in \mathcal{R}$, *whenever* $P \;\xrightarrow{l}_s\; P'$ *where* l *is sound with respect to* P *and* Q, *and also* $\mathcal{FN}(l) \subset \mathcal{FN}(P, Q)$, *there exists* Q' *such that* $Q \;\stackrel{\hat{l}}{\Longrightarrow}_s\; Q'$ *and* $\langle P',\ Q'\rangle \in \mathcal{R}$.

(ii) $\mathcal{R}$ *is symmetric.*

Then it is easy, using Proposition 3.21, that $P \approx''_s Q$ *if and only if* $P \approx_s Q$ *(see Appendix). Using this it is immediate that we can restrict* $\mathcal{FN}(t)$ *in Definition 3.19 itself in the same way. We do not chose to incorporate this in Definition, since the restriction works only when we have bound labels both for input and output, which may not be absolute as discussed in Remark 3.5.* ∎

Congruence of $\approx_s$

The lack of summation and matching allows us to establish congruence of $\approx_s$ (in [13] it was noted that, in the face of summation and matching the substitutivity cannot hold even for strong bisimilarity, leading them to the notion of ground and non-ground bisimilarities). Two technicalities in the following development are notable. One is the presence of name substitution, while the other is the lazy replicator. For proofs consult Appendix.

First, we need the following lemma, which is essential for later development. The proof uses Lemmas 3.8 and 3.9.

Lemma 3.27 *Variant Property.* Suppose $v \notin \mathcal{FN}(P, Q)$ below.

$$P \approx_s Q \quad \Rightarrow \quad P[v/x] \approx_s Q[v/x]\ .$$ ∎

Next, using again Lemmas 3.8 and 3.9 for transitivity,

Proposition 3.28 $\approx_s$ is an equivalence relation. That is, it is symmetric, reflexive and transitive. ∎

Now the first compatibility results for $\approx_s$ follow.

Proposition 3.29 For any P, Q, R and x,

(i) $P \approx_s Q \quad \Rightarrow \quad |x|P \approx_s |x|Q$.
(ii) $P \approx_s Q \quad \Rightarrow \quad (P, R) \approx_s (Q, R)$ ∎

For (i) one needs Lemma 3.27. The proof of (ii) uses (i) along the way.

With this we prove the general form of Lemma 3.27. We do not know whether there is a way to prove this without the previous development. The present proof uses, as well as Lemmas 3.6 and 3.9, Proposition 3.29 in an essential fashion: see Appendix.

Lemma 3.30 *Substitution lemma for $\approx_s$. For any P, Q, v and x,*

$$P \approx_s Q \quad \Rightarrow \quad P[v/x] \approx_s Q[v/x]. \qquad \blacksquare$$

This leads us easily to:

Proposition 3.31 For any P, Q and a, x,

$$P \ \approx_s \ Q \quad \Rightarrow \quad ax.P \ \approx_s \ ax.Q \ . \qquad \blacksquare$$

Finally we should show $\approx_s$ is closed under the only source of infinity in the calculus, *replication*. The following lemma, which shows closeness of our construction with Milner's "eager" replicator in [16], is essential for that.

Lemma 3.32 $!\,ax.P \ \approx_s \ ax.P, !\,ax.P.$ ∎

Using the lemma we can prove the following.

Proposition 3.33 $ax.P \approx_s by.Q \quad \Rightarrow \quad !\,ax.P \approx_s !\,by.Q.$

which, together with Propositions 3.29 and 3.31, establishes:

Proposition 3.34 *$\approx_s$ is a congruence relation.* ∎

This finishes our elementary investigation of $\approx_s$. In the next section we will see another weak bisimilarity with significant difference from $\approx_s$. Section 5 proves characterisation of $\approx_s$ by the corresponding Hennssy-Milner logic, and shows two bisimilarities have basically the same expressive power as far as their logical representations go.

4 Asynchronous Transition and Bisimulation

This section formulates another bisimilarity for ν-calculus, which is based on the notion of asynchronous observables. Essential difference with $\approx_s$ is noted and its congruence is proved.

Asynchronous Transition

The communication primitive of ν-calculus is asynchronous. This means, among others, that the sender does not participate in the event of message reception: it does not care how the messages it sends will be treated by the other party (syntactically this is represented by the lack of next action in output). To faithfully capture this notion in the framework of observables, we introduce *asynchronous transition relation* [9] in the following, where the basic intuition is that we regard synchronization is *not* essential for observing systems' behaviour

For labels and related syntactic notion, we refer to Definitions 3.1 and 3.2. The element of the transition relation is of the form (P, l, P'), written as $P \xrightarrow{l}_a P'$.

Definitions 4.1 *Asynchronous transition.* Asynchronous (one-step) transition, denoted $\xrightarrow{l}_a$, is the smallest relation generated by:

$$IN(f)_a: \quad |\tilde{w}|(\partial, \partial') \xrightarrow{\downarrow av}_a |\tilde{w}|(\partial, \leftarrow av, \partial') \qquad (a, v \notin \{\tilde{w}\})$$

$$IN(b)_a: \quad |\tilde{w}|(\partial, \partial') \xrightarrow{\downarrow a(v)}_a |\tilde{w}|(\partial, \leftarrow av, \partial') \qquad (a \notin \{\tilde{w}\} \wedge v \notin \mathcal{N}(\partial, \partial') \cup \{a\tilde{w}\})$$

$$OUT(f)_a: \quad |\tilde{w}|(\partial, \leftarrow av, \partial') \xrightarrow{\uparrow av}_a |\tilde{w}|(\partial, \partial') \qquad (a, v \notin \{\tilde{w}\})$$

$$OUT(b)_a: \quad |\tilde{w}|(\partial, \leftarrow av, \partial') \xrightarrow{\uparrow a(v)}_a |\tilde{w} \backslash v|(\partial, \partial') \qquad (a \notin \{\tilde{w}\} \wedge v \in \{\tilde{w}\})$$

$$COM_a: \quad |\tilde{w}|(\partial, \leftarrow av, ax.P, \partial') \xrightarrow{\tau}_a |\tilde{w}|(\partial, P[v/x], \partial')$$

$$STRUCT_a: \quad \frac{P_1' \equiv P_1, \; P_1 \xrightarrow{l}_a P_2, \; P_2 \equiv P_2'}{P_1' \xrightarrow{l}_a P_2'}$$

where in COM we call $\leftarrow av, ax.P$ a *redex pair* or simply a redex. ∎

Note the only differences from synchronous transition system lies in $IN(f)_a$ and $IN(b)_a$ rules. Regarding $IN(f)$ we have:

Proposition 4.2 For any $P \in \mathbf{C}$ and a, v, $\quad P \xrightarrow{\downarrow av}_a (P, \leftarrow av)$. ∎

which is in sharp contrast with Proposition 3.4 in the case of $\xrightarrow{l}_s$. We also note bound labels are not essential as far as one step transition goes in $\xrightarrow{l}_a$ just as in $\xrightarrow{l}_s$.

The following fact shows an elementary relationship between $\xrightarrow{l}_s$ and $\xrightarrow{l}_a$.

Proposition 4.3 For a specific l, we let $\xrightarrow{l}_a$ represent all $\langle P,\ Q\rangle$ such that (P, l, Q) is in the transition relation. With $\downarrow\alpha$ (resp. $\uparrow\alpha$) standing for both free and bound input (resp. output) labels, we have:

$$\begin{array}{rcl} \xrightarrow{\uparrow\alpha}_a & = & \xrightarrow{\uparrow\alpha}_s \\ \xrightarrow{\tau}_a & = & \xrightarrow{\tau}_s \\ \xrightarrow{\downarrow\alpha}_a\xrightarrow{\tau}_a & \supset & \xrightarrow{\downarrow\alpha}_s\ . \end{array} \qquad \blacksquare$$

Other facts concerning $\xrightarrow{l}_a$ are listed below. They respectively correspond to Lemmas 3.6, 3.8, and 3.9 in Section 3. The first one has a minor difference from the corresponding one.

Lemma 4.4 *Free name lemma for* $\xrightarrow{l}_a$.

(1) $P \xrightarrow{\downarrow av}_a P' \vee P \xrightarrow{\downarrow a(v)}_a P' \quad \Rightarrow \quad \mathcal{FN}(P) \cup \{a, v\} = \mathcal{FN}(P')$

(2) $P \xrightarrow{\uparrow av}_a P' \quad \Rightarrow \quad \mathcal{FN}(P) = \mathcal{FN}(P') \cup \{a, v\}$

(3) $P \xrightarrow{\uparrow a(v)}_a P' \quad \Rightarrow \quad \mathcal{FN}(P) \cup \{v\} = \mathcal{FN}(P') \cup \{a, v\}$

(4) $P \xrightarrow{\tau}_a P' \quad \Rightarrow \quad \mathcal{FN}(P) \supset \mathcal{FN}(P')$ $\blacksquare$

Lemma 4.5 α-conversion in labels. *Suppose* $\mathcal{BN}(l) \neq \emptyset$ *below.*

$$P \xrightarrow{l}_a Q \ \wedge \ l \equiv_\alpha l' \ \wedge \ \mathcal{BN}(l') \not\subset \mathcal{FN}(P) \quad \Rightarrow \quad P \xrightarrow{l'}_a Q\{l\backslash\backslash l'\} \doteq Q\ . \qquad \blacksquare$$

Lemma 4.6 *Substitution lemma for* $\xrightarrow{l}_a$.

(i) $P \xrightarrow{l}_a P' \ \wedge \ \mathcal{BN}(l) = \emptyset \quad \Rightarrow \quad P\{\tilde{v}/\tilde{x}\} \xrightarrow{l\{\tilde{v}/\tilde{x}\}}_a P'\{\tilde{v}/\tilde{x}\}$.

(ii) $P \xrightarrow{l}_a P' \wedge \mathcal{BN}(l) \neq \emptyset \wedge \mathcal{BN}(l) \cap \{\tilde{v}\} = \emptyset \quad \Rightarrow \quad P\{\tilde{v}/\tilde{x}\} \xrightarrow{l\{\tilde{v}/\tilde{x}\}}_a P'[z/\pi_2(l)]\{\tilde{v}/\tilde{x}\}[\pi_2(l)/z]$ with z fresh. $\blacksquare$

Weak Asynchronous Transition

A weak transition relation, derived from $\xrightarrow{l}_a$, is defined in the following.

Definitions 4.7 Weak asynchronous transitions. *We inductively define the weak asynchronous transition, denoted by* $\overset{t}{\Longrightarrow}_a$, *as the smallest relation generated by the following rules.*

$$\frac{P \xrightarrow{l}_a P'}{P \overset{l}{\Longrightarrow}_a P'}\ (l \neq \tau) \qquad\qquad \frac{P \xrightarrow{\tau}_a P'}{P \overset{\varepsilon}{\Longrightarrow}_a P'}$$

$$\frac{P \overset{t}{\Longrightarrow}_a P' \qquad P' \overset{t'}{\Longrightarrow}_a P''}{P \overset{t\cdot t'}{\Longrightarrow}_a P''} \qquad \blacksquare$$

Two essential properties of $\overset{t}{\Longrightarrow}_a$ are stated below. The second one easily follows from the first one.

Lemma 4.8 *α-conversion on weak transition. Suppose t' is regular with $t \equiv_\alpha t'$, and also $\mathcal{BN}(t') \cap (\mathcal{FN}(P) \cup \mathcal{N}(t)) = \emptyset$. Then we have*

$$P \stackrel{t}{\Longrightarrow}_a Q \quad \Rightarrow \quad P \stackrel{t'}{\Longrightarrow}_a Q\{t \backslash\backslash t'\} \doteq Q \qquad \blacksquare$$

Lemma 4.9 *Substitution lemma for $\stackrel{l}{\Longrightarrow}_a$.*

(i) $P \stackrel{\varepsilon}{\Longrightarrow}_a P' \quad \Rightarrow \quad P\{\tilde{v}/\tilde{x}\} \stackrel{\varepsilon}{\Longrightarrow}_a P'\{\tilde{v}/\tilde{x}\}$

(ii) $P \stackrel{l}{\Longrightarrow}_a P' \wedge \mathcal{BN}(l) \neq \emptyset \wedge \mathcal{BN}(l) \cap \{\tilde{v}\} = \emptyset \quad \Rightarrow \quad P\{\tilde{v}/\tilde{x}\} \stackrel{l\{\tilde{v}/\tilde{x}\}}{\Longrightarrow}_a P'[z/\pi_2(l)]\{\tilde{v}/\tilde{x}\}[\pi_2(l)/z]$

(iii) In general, $P \stackrel{t}{\Longrightarrow}_a P' \wedge \{\tilde{x}\tilde{v}\} \cap \mathcal{BN}(t) = \emptyset \quad \Rightarrow \quad P\{\tilde{v}/\tilde{x}\} \stackrel{t\{\tilde{v}/\tilde{x}\}}{\Longrightarrow}_a P'\{\tilde{v}/\tilde{x}\} \quad \blacksquare$

Asynchronous Bisimilarity

In the following we define *weak asynchronous bisimilarity.* We will later show its significant difference from the synchronous theory. The definition to follow is given based on multi-step transition (cf. [19]) though we will immediately recover the usual formulation (cf. [16]) which will be mainly used for proofs to follow.

Definitions 4.10 *Asynchronous bisimulation.* P_1 and Q_1 are asynchronously bisimilar, denoted by $P_1 \approx_a Q_1$, if and only if $(P_1, Q_1) \in \mathcal{R}$ with

(i) For any $\langle P, Q\rangle \in \mathcal{R}$, whenever $P \stackrel{t}{\Longrightarrow}_a P'$ where t is regular and is also sound with respect to P and Q, there exists Q' such that $Q \stackrel{t}{\Longrightarrow}_a Q'$ and $\langle P', Q'\rangle \in \mathcal{R}$.

(ii) $\mathcal{R}$ is symmetric. $\blacksquare$

We call $\mathcal{R}$ as above a *weak asynchronous bisimulation.* Then $\approx_a$ (the *weak asynchronous bisimilarity*) is the union of all the bisimulations. Hereafter we simply call them "asynchronous bisimulations" and "asynchronous bisimilarity", respectively. As in the synchronous case, we first recover a usual definition of bisimulation.

Proposition 4.11 Define $P_1 \approx'_a Q_1$ when $\langle P_1, Q_1\rangle \in \mathcal{R}$ with

(i) For any $\langle P, Q\rangle \in \mathcal{R}$, whenever $P \stackrel{l}{\longrightarrow}_a P'$ where l is sound with respect to P and Q, there exists Q' such that $Q \stackrel{\hat{l}}{\Longrightarrow}_a Q'$ and $\langle P, Q'\rangle \in \mathcal{R}$.

(ii) $\mathcal{R}$ is symmetric.

Then $P \approx'_a Q$ if and only if $P \approx_a Q$. $\blacksquare$

In comparison with $\approx_s$, the following is essential.

Proposition 4.12 $P \approx_s Q \quad \Rightarrow \quad P \approx_a Q.$ $\blacksquare$

The easiest proof uses the congruence of $\approx_s$ then the result is immediate via Proposition 4.3. This shows that all equations which hold in $\approx_s$ also hold in $\approx_a$.

But significant difference exists between two theories, as the following example shows.

Examples 4.13 *For any a, we have* $\mathcal{I}_0(a) \approx_a \mathcal{I}(a) \approx_a \mathcal{I}_2(a) \approx_a \Lambda$. ∎

Note how the above result is in accordance with the characterisation of these agents in Proposition 2.20, which is given purely in terms of reduction relation. Hence, as far as these equations go, $\approx_a$ (hence the notion of asynchronous observables) gives us more appropriate semantics than $\approx_s$ (hence than synchronous observables). One immediate consequence follows.

Proposition 4.14 $\approx_a$ *strictly includes* $\approx_s$. ∎

In Section 5 we will see that, in spite of the strict difference in equating agents, two bisimilarities are equally rich when represented in the form of *logics*. Also, along the way, it will be shown that the difference between two bisimilarities is closely related with the identity receptors.

Remarks 4.15 *We note that here again (cf. Remark 3.26) we can have a better characterisation result than shown before. Define* $\approx_a''$ *as* $P_1 \approx_a'' Q_1$ *if and only if* $(P_1, Q_1) \in \mathcal{R}$ *with*

- *(i) For any* $\langle P,\ Q \rangle \in \mathcal{R}$*, whenever* $P \xrightarrow{l}_a P'$ *where* l *is sound with respect to* P *and* Q*, and also* $\mathcal{FN}(l) \in \mathcal{FN}(P)$*, there exists* Q' *such that* $Q \stackrel{\hat{l}}{\Longrightarrow}_a Q'$ *and* $\langle P',\ Q' \rangle \in \mathcal{R}$ *.*
- *(ii)* $\mathcal{R}$ *is symmetric.*

And the similar restriction is possible for t *in Definition 4.10. For the same reason we noted in Remark 3.26, however, we do not incorporate it into the definition. Remark also that we should still keep the generality of* $IN(f, b)_a$ *rules in Definition 4.1, since we cannot bound possible free names occurring in* Q *(w.r.t.* P*). Any restriction in this respect would let us lose the power of equating e.g.* $\mathcal{I}(a)$ *and* Λ. ∎

Congruence of $\approx_a$

In the following we validate the congruence of $\approx_a$. We use Lemma 4.16 later instead of the variant property regarding $\approx_s$ (Lemma 3.27). Again all the proofs are listed in Appendix.

First we state the following lemma which is used in the subsequent reasoning, instead of the *variant property*, Lemma 3.27, in the case of $\approx_s$. It says that $\approx_a$ is preserved under concurrent composition with arbitrary messages.

Lemma 4.16 Let ∂ be an arbitrary sequence of concurrent composition of messages. Then

$$P \approx_a Q \quad \Rightarrow \quad (P, \partial) \approx_a (Q, \partial) . \quad ∎$$

Proposition 4.17 uses Lemmas 4.5 and 4.6. In Proposition 4.18, (i) needs the above lemma, while (ii) uses (i). Note the line of reasoning parallels $\approx_s$ (except the use of Lemma 4.16).

Proposition 4.17 $\approx_a$ is an equivalence relation. That is, it is symmetric, reflexive and transitive. ∎

Proposition 4.18 For any P, Q, R and x,

$$\text{(i)}\ P \approx_a Q \quad \Rightarrow \quad |x|P \approx_a |x|Q\ .$$
$$\text{(ii)}\ P \approx_a Q \quad \Rightarrow \quad (P, R) \approx_a (Q, R)\ . \quad \blacksquare$$

To establish the substitutivity of expressions for a body of a receptor, substitutivity under arbitrary substitution of names, is essential. The following lemma, established by extensively using Lemmas 2.10 and 4.6, shows that in our formalism the substitutivity indeed holds.

Lemma 4.19 *Substitution lemma for $\approx_a$. For any P, Q, v and x,*

$$P \approx_a Q \;\;\Rightarrow\;\; P[v/x] \approx_a Q[v/x]. \qquad \blacksquare$$

This immediately leads us to:

Proposition 4.20 For any P, Q and a, x,

$$P \;\approx_a\; Q \quad \Rightarrow \quad ax.P \;\approx_a\; ax.P\ . \qquad \blacksquare$$

Finally we show $\approx_a$ is closed under *replication.* We first need:

Lemma 4.21 $!\,ax.P \;\approx_a\; !\,ax.P, ax.P$. ∎

which is direct from Lemma 3.32 and Proposition 4.12. This let us prove:

Proposition 4.22 $ax.P \approx_a by.Q \quad \Rightarrow \quad !\,ax.P \approx_a \,!\,by.Q.$

Summarising Propositions 4.18, 4.20 and 4.22, we have:

Proposition 4.23 $\approx_a$ *is a congruence relation.* ∎

5 Modal Logics and Their Embeddings

This section gives modal characterisation to two bisimilarities and then study relative expressive power of two semantic theories using these corresponding logics, showing each has a complete and sound embedding in another.

Modal Logics for Weak Transition

We in the following study modal logics for the calculus in the weak settings, which characterise the respective bisimilarities. It will be shown later that each logic has a sound and complete embedding in another. Thus in their counterparts in logics, which can be regarded as finer representation of respective semantic theories, two bisimilarities have essentially the same expressive power.

We formulates two logics, $\mathcal{L}_s$ and $\mathcal{L}_a$, which correspond to $\approx_s$ and $\approx_a$, respectively.

We will use the minimum set of Hennessy-Milner logic: possibility, negation, and conjunction. Necessity and disjunction can be expressed with them using the standard method. The same set of formulae are used for two logics.

Definitions 5.1 *Syntax for the logical formulae. The set of logical formulae, **F**, ranged over by A, B,.., etc. are given by the following syntax.*

$$A ::= \langle t\rangle A \mid \neg A \mid \bigwedge_{i\in I} A_i$$

where we assume I is enumerable. For a formula we define the sets of free and bound names, as follows.

(i) $\mathcal{FN}(\langle t\rangle A) \stackrel{def}{=} \mathcal{FN}(t) \cup \mathcal{FN}(A)\backslash\mathcal{BN}(t), \qquad \mathcal{BN}(\langle t\rangle A) \stackrel{def}{=} \mathcal{BN}(t) \cup \mathcal{BN}(A)$

(ii) $\mathcal{FN}(\neg A) \stackrel{def}{=} \mathcal{FN}(A), \qquad \mathcal{BN}(\neg A) \stackrel{def}{=} \mathcal{BN}(A)$

(iii) $\mathcal{FN}(\bigwedge_{i\in I} A_i) \stackrel{def}{=} \bigcup_{i\in I} \mathcal{FN}(A_i), \qquad \mathcal{BN}(\bigwedge_{i\in I} A_i) \stackrel{def}{=} \bigcup_{i\in I} \mathcal{BN}(A_i)$

(iv) $\mathcal{N}(A) \stackrel{def}{=} \mathcal{FN}(A) \cup \mathcal{BN}(A)$ ∎

For these formulae, we explicitly define the notion of *substitution*, for it involves some technicalities in the treatment of bindings.

Definitions 5.2 *Substitution on logical formulae. Let us have a* concatenation of a label to a formula *defined by:*

$$t_1 :: \langle t_2\rangle A \stackrel{def}{=} \langle t_1 t_2\rangle A$$

then substitution of a formulae is defined as follows. As usual, the substitution associates strongest, e.g. $\neg A[v/x] \stackrel{def}{=} \neg(A[v/x])$.

$$\begin{array}{lcll}
(\langle\varepsilon\rangle A)[v/x] & \stackrel{def}{=} & \langle\varepsilon\rangle(A[v/x]) & \\
(\langle lt\rangle A)[v/x] & \stackrel{def}{=} & l[v/x] :: (\langle s\rangle A)[v/x]) \ \ \textit{if}\ \mathcal{BN}(l)\cap\{x,v\}=\emptyset & \\
(\langle{\downarrow} a(x)t\rangle A)[v/x] & \stackrel{def}{=} & \langle{\downarrow} a[v/x](x)t\rangle A & \\
(\langle{\uparrow} a(x)t\rangle A)[v/x] & \stackrel{def}{=} & \langle{\uparrow} a[v/x](x)t\rangle A & \\
(\langle{\downarrow} a(v)t\rangle A)[v/x] & \stackrel{def}{=} & {\downarrow} a[v/x](z) :: (\langle t\rangle A)[z/v][v/x] & (z\ \textit{fresh}) \\
(\langle{\uparrow} a(v)t\rangle A)[v/x] & \stackrel{def}{=} & {\uparrow} a[v/x](z) :: (\langle t\rangle A)[z/v][v/x] & (z\ \textit{fresh}) \\
(\neg A)[v/x] & \stackrel{def}{=} & \neg A[v/x] & \\
(\bigwedge_{i\in I} A_i)[v/x] & \stackrel{def}{=} & \bigwedge_{i\in I} A_i[v/x] &
\end{array}$$

∎

Based on substitution, we define α-convertibility.

Definitions 5.3 *α-conversion of logical formulae. α-convertibility between formulae, $\equiv_\alpha$, is the least relation generated by the following rules.*

(i) $\langle t_1 \cdot \downarrow a(x) \cdot t_2 \rangle A \equiv_\alpha t_1 \cdot \downarrow a(z) :: (\langle t_2 \rangle A)[z/x]$ *where* $z \notin \mathcal{FN}(\langle t_2 \rangle A)$.

(ii) $\langle t_1 \cdot \uparrow a(x) \cdot t_2 \rangle A \equiv_\alpha t_1 \cdot \uparrow a(z) :: (\langle t_2 \rangle A)[z/x]$ *where* $z \notin \mathcal{FN}(\langle t_2 \rangle A)$

(iii) $\equiv_\alpha$ *is a congruence relation.* ∎

Immediately

Proposition 5.4 *α convertibility in formulae and transition.*

$$\langle t \rangle A \equiv_\alpha \langle t' \rangle A' \quad \Rightarrow \quad t \equiv_\alpha t' \ \wedge \ A' \equiv_\alpha A\{t \backslash\backslash t'\}. \qquad \blacksquare$$

Now we formulate two satisfaction relations. There are two such relations, one for asynchronous weak transition and the other for synchronous weak transition. The statement has the following form:

$$P \models_{\mathcal{L}} A$$

which reads P satisfies the formula A (in a logic $\mathcal{L}$). First we define the satisfaction relation for the synchronous logic. [8]

Definitions 5.5 *Satisfaction relation (1). We define $\models_{\mathcal{L}_s} \subset \boldsymbol{C} \times \boldsymbol{F}$ as follows.*

$$\begin{array}{rcl}
P \stackrel{t}{\Longrightarrow}_s P' \ \wedge \ P' \models_{\mathcal{L}_s} A & \Rightarrow & P \models_{\mathcal{L}_s} \langle t \rangle A \\
\textit{not} \ P \models_{\mathcal{L}_s} A & \Rightarrow & P \models_{\mathcal{L}_s} \neg A \\
\textit{for all } i \in I, \ \ P \models_{\mathcal{L}_s} A_i & \Rightarrow & P \models_{\mathcal{L}_s} \bigwedge_{i \in I} A_i \\
P \models_{\mathcal{L}_s} A \ \wedge \ A \equiv_\alpha A' & \Rightarrow & P \models_{\mathcal{L}_s} A' \qquad \blacksquare
\end{array}$$

The similar definition for the asynchronous bisimilarity is given below.

Definitions 5.6 *Satisfaction relation (2). We define $\models_{\mathcal{L}_a} \subset \boldsymbol{C} \times \boldsymbol{F}$ as follows.*

$$\begin{array}{rcl}
P \stackrel{t}{\Longrightarrow}_a P' \ \wedge \ P' \models_{\mathcal{L}_a} A & \Rightarrow & P \models_{\mathcal{L}_a} \langle t \rangle A \\
\textit{not} \ P \models_{\mathcal{L}_a} A & \Rightarrow & P \models_{\mathcal{L}_a} \neg A \\
\textit{for all } i \in I, \ \ P \models_{\mathcal{L}_a} A_i & \Rightarrow & P \models_{\mathcal{L}_a} \bigwedge_{i \in I} A_i \\
P \models_{\mathcal{L}_a} A \ \wedge \ A \equiv_\alpha A' & \Rightarrow & P \models_{\mathcal{L}_a} A' \qquad \blacksquare
\end{array}$$

Then the triple of $(\mathbf{C}, \mathbf{F}, \models_{\mathcal{L}_s})$ defines a logic and is denoted by $\mathcal{L}_s$. Similarly we have a logic $(\mathbf{C}, \mathbf{F}, \models_{\mathcal{L}_a})$, denoted by $\mathcal{L}_a$. We will often state common properties of two logics. In those cases we will often write e.g. $P \models_{\mathcal{L}_{a,s}} A$ etc. The following notations are often convenient later.

[8] Our formulation incorporate $\equiv_\alpha$ in defining relation. It is possible, as in [14], to incorporate specific conditions for the following rules and later prove the relation identifies α convertible formulae. We chose the present formulation in the same spirit as in Convention 2.9.

Definitions 5.7 *We define:*

$$\begin{aligned}
[P]_{\mathcal{L}_s} &\stackrel{def}{=} \{A \mid P \models_{\mathcal{L}_s} A\} \\
[P]_{\mathcal{L}_a} &\stackrel{def}{=} \{A \mid P \models_{\mathcal{L}_a} A\} \\
[P]^{\mathcal{F}}_{\mathcal{L}_s} &\stackrel{def}{=} \{A \mid P \models_{\mathcal{L}_s} A \ \wedge\ \mathcal{FN}(A) \subset \mathcal{F}\} \\
[P]^{\mathcal{F}}_{\mathcal{L}_a} &\stackrel{def}{=} \{A \mid P \models_{\mathcal{L}_a} A \ \wedge\ \mathcal{FN}(A) \subset \mathcal{F}\} \qquad \blacksquare
\end{aligned}$$

Characterisation Results

Before going into characterisation of respective bisimilarities by logics, some basic facts concerning satisfaction relation are stated below. The following two are direct from the definition.

Lemma 5.8 $P \models_{\mathcal{L}_{a,s}} \langle t_1 t_2 \rangle A \quad \Leftrightarrow \quad P \models_{\mathcal{L}_{a,s}} \langle t_1 \rangle \langle t_2 \rangle A$.

Lemma 5.9 *Either* $P \models_{\mathcal{L}_{a,s}} A$ *or* $P \models_{\mathcal{L}_{a,s}} \neg A$. $\blacksquare$

Another lemma (cf. Lemma 3 of [14]) is a variant theorem in the case of statement.

Lemma 5.10 *Variant lemma for statement.*

$$P \models_{\mathcal{L}_{a,s}} A \ \wedge\ v \notin \mathcal{FN}(P) \cup \mathcal{FN}(A) \quad \Rightarrow \quad P[v/x] \models_{\mathcal{L}_{a,s}} A[v/x] \quad \blacksquare$$

The preceding lemmas lead us to the following property, which connects formulae and corresponding transition. In a sense it says that α-convertibility does not let us lose operational content of satisfaction relation[9].

Lemma 5.11

$$P \models_{\mathcal{L}_{a,s}} \langle t \rangle A \ \wedge\ \mathcal{BN}(t) \cap \mathcal{FN}(P) = \emptyset \ \wedge\ t \textit{ is regular} \quad \Rightarrow \quad P \stackrel{t}{\Longrightarrow}_{a,s} P' \models_{\mathcal{L}_{a,s}} A \qquad \blacksquare$$

Using these properties, the standard method [7] let us prove the characterisation of bisimilarities by respective logics.

Theorem 5.12

(i) $P \approx_s Q \quad \Leftrightarrow \quad [P]_{\mathcal{L}_s} = [Q]_{\mathcal{L}_s}$.

(ii) $P \approx_a Q \quad \Leftrightarrow \quad [P]_{\mathcal{L}_a} = [Q]_{\mathcal{L}_a}$. $\blacksquare$

We can make the right-hand side weaker than above, via the result in Remarks 3.26 and 4.15, as follows. Note $\Longrightarrow$ in the proposition is immediate from the theorem.

Proposition 5.13

(i) $P \approx_s Q \quad \Leftrightarrow \quad [P]^{\mathcal{FN}(P,Q)}_{\mathcal{L}_s} = [Q]^{\mathcal{FN}(P,Q)}_{\mathcal{L}_s}$.

(ii) $P \approx_a Q \quad \Leftrightarrow \quad [P]^{\mathcal{FN}(P,Q)}_{\mathcal{L}_a} = [Q]^{\mathcal{FN}(P,Q)}_{\mathcal{L}_a}$. $\blacksquare$

The result is important regarding the second embedding result.

[9] The reverse implication does not hold in the lemma, since e.g. $\leftarrow ac, |a|(\leftarrow ba, P) \models_{\mathcal{L}_{a,s}} \langle \uparrow ac \uparrow b(a) \rangle A$.

Embedding of $\mathcal{L}_s$ in $\mathcal{L}_a$

Logics we formulated so far are convenient tools to compare $\approx_a$ and $\approx_s$. One essential result we can gain, which we will discuss subsequently, is that there are sound and complete embeddings between two logics. By *sound and complete embedding* we mean that, in our setting, a formula is satisfied by a certain term in a logic if and only if the image of the former is also satisfied by the image of the latter in another logic. In this subsection we are going to deal with an embedding of $\mathcal{L}_s$ in $\mathcal{L}_a$, showing that there exist a pair of mappings $\Upsilon : \mathbf{C} \to \mathbf{C}$ and $\Psi : \mathbf{F} \to \mathbf{F}$ such that

$$P \models_{\mathcal{L}_s} A \quad \Leftrightarrow \quad \Upsilon(P) \models_{\mathcal{L}_a} \Psi(A).$$

The result says that (in this translation) $\mathcal{L}_s$ is a subsystem of $\mathcal{L}_a$, so that $\mathcal{L}_a$ can be regarded at least no less strong than $\mathcal{L}_a$.

The basic idea of embedding $\mathcal{L}_s$ in $\mathcal{L}_a$ is to transform a term so that its "synchronous" observables can be observed as "asynchronous" observables. The following shows the transformation. This corresponds to Υ above. The coding is adopted from [10].

Definitions 5.14 *We define a mapping* $[\![P]\!]$ *from* $\boldsymbol{C}$ *to* $\boldsymbol{C}$ *as follows. We assume that newly introduced names are always fresh.*

$$\begin{array}{rcl}
[\![\leftarrow av]\!] & \stackrel{def}{=} & az.(\leftarrow zv) \\
[\![ax.P]\!] & \stackrel{def}{=} & |c|(\leftarrow ac, cx.[\![P]\!]) \\
[\![|x|P]\!] & \stackrel{def}{=} & |x|[\![P]\!] \\
[\![P, Q]\!] & \stackrel{def}{=} & [\![P]\!], [\![Q]\!] \\
[\![!\, ax.P]\!] & \stackrel{def}{=} & |e|(\leftarrow e,\ !\, e.|c|(\leftarrow ac, cx.([\![P]\!], \leftarrow e))) \\
[\![\Lambda]\!] & \stackrel{def}{=} & \Lambda\ .
\end{array} \quad \blacksquare$$

What we did above is, basically speaking, to let the asynchronous message passing involve *synchronization*, but preserving the overall behaviour of interaction. How this transformation of terms results in transformation of observables, is shown in the following mapping.

Definitions 5.15 *A mapping* $t^\star$ *from* $(\boldsymbol{L}\backslash\tau)^*$ *to* $(\boldsymbol{L}\backslash\tau)^*$ *is given as follows. We assume newly introduced names are always fresh and the symbol "$\star$" associates stronger than other constructors.*

$$\begin{array}{rcl}
\varepsilon^\star & = & \varepsilon \\
(\downarrow av \cdot t)^\star & = & \uparrow a(c) \downarrow cv \cdot t^\star \\
(\downarrow a(v) \cdot t)^\star & = & \uparrow a(c) \downarrow c(v) \cdot t^\star \\
(\uparrow av \cdot s)^\star & = & \downarrow a(c) \uparrow cv \cdot t^\star \\
(\uparrow a(v) \cdot s)^\star & = & \downarrow a(c) \uparrow c(v) \cdot t^\star
\end{array}$$

We also define $\mathcal{N}^\star(t) \stackrel{def}{=} \mathcal{N}(t^\star)\backslash\mathcal{N}(t)$ *e.g.* $\mathcal{N}^\star(\uparrow a(c) \downarrow cv) = \{c\}$. $\blacksquare$

One small but important fact, used in proofs of e.g. Theorem 5.20, is:

Lemma 5.16 *In* $(lt)^\star \stackrel{def}{=} l_1 \cdot l_2 \cdot (t^\star)$, $\pi_2(l_1)(= \pi_1(l_2))$ *never occurs again in* t. $\blacksquare$

Now the mapping $t^\star$ naturally leads us to a mapping of logical formulae. This corresponds to Ψ before.

Definitions 5.17 *With $(\cdot)^\star$ as given before, the mapping of a formula, denoted by $(\cdot)^\circ$, is defined as follows. We assume the symbol "$\circ$" associates stronger than other constructors.*

$$\begin{aligned} (\langle t \rangle A)^\circ &\stackrel{def}{=} \langle t^\star \rangle A^\circ \\ (\neg A)^\circ &\stackrel{def}{=} \neg A^\circ \\ (\bigwedge_{i\in I} A_i)^\circ &\stackrel{def}{=} \bigwedge_{i\in I} A_i^\circ \qquad \blacksquare \end{aligned}$$

Lemma 5.16 extends to the formulae.

Lemma 5.18 *In $(\langle lt \rangle A)^\circ \stackrel{def}{=} \langle l_1 \cdot l_2 \cdot t^\star \rangle A^\circ$, $\pi_2(l_1) = \pi_1(l_2)$ never occur in $t^\star$ and A°.* $\blacksquare$

The following property of the mapping is essential for the embedding result. Remember $\overline{\mathcal{F}} \stackrel{def}{=} \mathbf{N}\backslash\mathcal{F}$.

Proposition 5.19

(i) $P \stackrel{t}{\Longrightarrow}_s P' \quad \Rightarrow \quad \exists P''.\ [\![P]\!] \stackrel{t^\star}{\Longrightarrow}_a P'' \approx_a [\![P']\!]$.

(ii) Suppose $[\![P]\!] \stackrel{t^\star}{\Longrightarrow}_a Q$. *Then there exists some R with the following property.*

- *(1)* $Q \stackrel{\varepsilon}{\Longrightarrow}_a R \approx_a [\![P']\!]$ *for some* P' *with* $P \stackrel{t}{\Longrightarrow}_s P'$.
- *(2)* $[Q]_{\mathcal{L}_a}^{\overline{\mathcal{N}^\star(t)}} = [R]_{\mathcal{L}_a}^{\overline{\mathcal{N}^\star(t)}}$ $\blacksquare$

The main result follows, which shows $\mathcal{L}_s$ has a sound and complete embedding in $\mathcal{L}_a$. It uses induction on the structure of A to prove bidirectional implication at the same time. See Appendix.

Theorem 5.20 *The embedding of $\mathcal{L}_s$ in $\mathcal{L}_a$.*

$$P \models_{\mathcal{L}_s} A \quad \Leftrightarrow \quad [\![P]\!] \models_{\mathcal{L}_a} A^\circ \ . \qquad \blacksquare$$

Remarks 5.21

(i) Note that there are many formulae satisfied by $[\![P]\!]$, which are not *in the form stipulated above, i.e. $[\![P]\!] \models_{\mathcal{L}_a} A$ does not imply there is some formula A' such that $A' = A^\circ$ and $P \models_{\mathcal{L}_a} A'$. Those stipulated formulae represent the "correct" interaction sequence of the term while those which are not represent "incorrect" sequences. Indeed the following does* not *hold (cf. [10]):*

$$P \approx_s Q \ \Rightarrow \ [\![P]\!] \approx_a [\![Q]\!] \tag{1}$$

thus the theorem strongly depends on the power of $\mathcal{L}_a$ *to* separate *the set of statements for correct interaction sequences* from *the set of statements for incorrect interaction sequences. The discernibility of correctness of sequences in logical formulae (cf. usage of Lemma 5.18 in the proof of the theorem) is of independent interest, along with its representation scheme using bindings in an essential fashion. Also of interest is to find a simple compositional encoding which has the property (1), though we only have negative results so far.*

(ii) The method of the embedding can be applied to embeddings of various calculi in $\mathcal{L}_a$ *as well. For example, with the almost identical encoding, we can show that* $\mathcal{L}_a$ *can embed a version of a logic for the fragmentary* π*-calculus in [16]. A challenging work in this respect is embedding of a calculus with the general summation, may it be CCS or* π*-calculus, in the logic* $\mathcal{L}_a$*. Note that* $\approx_a$ *is a congruence relation while once we add the general summation we lose the congruence: this suggests that the embedding may be a non-trivial task (though is far from implying it is impossible).* ∎

Embedding of $\mathcal{L}_a$ in $\mathcal{L}_s$

We next show we can have a converse embedding, i.e. representing statements of $\mathcal{L}_a$ in $\mathcal{L}_s$. The embedding is sound, but is complete only relative to a finite set of names (which correspond to names occurring free in logical formulae). This does not let us lose any generality due to Proposition 5.13.

The mapping of terms onto terms is given below. It uses $\mathcal{I}_2(a)$ of Definition 2.19. As noted, it is parameterized by a finite set of names. A notable thing is we do not need any encoding of formulae: if an encoded term satisfies some formula in $\mathcal{L}_s$, the original term satisfies that formula as it is in $\mathcal{L}_a$.

Definitions 5.22 Suppose A is a finite set of names, e.g. $\mathcal{A} = \{a_1, a_2, .., a_n\}$. Then we define $\mathcal{I}_2(\mathcal{A}) \stackrel{def}{=} \mathcal{I}_2(a_1), .., \mathcal{I}_2(a_n)$. ∎

Definitions 5.23 $\mathcal{I}$ *Completion.* With the function $\psi : \mathbf{C} \rightarrow \mathbf{C}$,

$$\begin{aligned}
\psi(\Lambda) &\stackrel{def}{=} \Lambda \\
\psi(\leftarrow av) &\stackrel{def}{=} \leftarrow av \\
\psi(ax.P) &\stackrel{def}{=} ax.(\mathcal{I}_2(x), \psi(P)) \\
\psi(!\, ax.P) &\stackrel{def}{=} !\, \psi(ax.P) \\
\psi(|x|P) &\stackrel{def}{=} |x|(\mathcal{I}_2(x),\ \psi(P)) \\
\psi(P,\ Q) &\stackrel{def}{=} \psi(P),\ \psi(Q)
\end{aligned}$$

we define, for some finite set $\mathcal{F}$,

$$\Phi_{\mathcal{F}}(P) \stackrel{def}{=} \mathcal{I}_2(\mathcal{F}),\ \psi(P)$$

We also define $\Phi^*_{\mathcal{F}}(P)$ as the smallest set generated by the rules:

(i) $\Phi_{\mathcal{F}}(P) \in \Phi^*_{\mathcal{F}}(P)$.

(ii) $|\tilde{w}|(\mathcal{I}_2(\tilde{a}),\ \partial) \in \Phi^*_{\mathcal{F}}(P) \quad \Rightarrow \quad |\tilde{w}|(\mathcal{I}_2(\{\tilde{a}\}),\ \partial) \in \Phi^*_{\mathcal{F}}(P)$.

(iii) $|\tilde{w}|(\mathcal{I}_2(A),\ \partial) \in \Phi^*_{\mathcal{F}}(P) \ \wedge\ \{\tilde{a}\} \subset A \quad \Rightarrow \quad |\tilde{w}|(\mathcal{I}_2(\tilde{a}),\ \mathcal{I}_2(A),\ \partial) \in \Phi^*_{\mathcal{F}}(P)$.

(iv) $|\tilde{w}|(\mathcal{I}_2(\tilde{a}),\ \partial) \in \Phi^*_{\mathcal{F}}(P) \ \wedge\ \mathcal{I}_2(\tilde{a}) \longrightarrow\!\!\!\!\!\rightarrow R \quad \Rightarrow \quad |\tilde{w}|(R,\ \partial) \in \Phi^*_{\mathcal{F}}(P)$. ∎

Now it is easy to see, using $\mathcal{I}_2(\tilde{a}) \longrightarrow\!\!\!\!\!\rightarrow R \quad \Rightarrow \quad R \approx_a \Lambda$,

Proposition 5.24 *For any $\mathcal{F}$,*

$$(P \in \Phi^*_{\mathcal{F}}(P_0) \ \wedge\ Q \in \Phi^*_{\mathcal{F}}(P_0)) \quad \Rightarrow \quad P \approx_a Q \approx_a P_0. \qquad ∎$$

Hence, by the modal characterisation result,

Proposition 5.25 *For any $\mathcal{F}$,* $\quad (P \in \Phi^*_{\mathcal{F}}(P_0) \ \wedge\ Q \in \Phi^*_{\mathcal{F}}(P_0)) \quad \Rightarrow \quad (P \models_{\mathcal{L}_a} A \quad \Leftrightarrow \quad Q \models_{\mathcal{L}_a} A)$. ∎

Now the following property is essential in understanding how the encoding represent the original asynchronous transition in the synchronous transition. The proof is long but mechanical.

Proposition 5.26 Let us write $\mathcal{FI}(t)$ for names of subjects of input labels occurring free in t. Then with $P \in \Phi^*_{\mathcal{F}}(P_0)$ for $\mathcal{F} \supset \mathcal{FN}(P_0)$, we have

(i) $P \stackrel{t}{\Longrightarrow}_s P' \quad \Rightarrow \quad P_0 \stackrel{t}{\Longrightarrow}_a P'_0$ for some P'_0 with $P' \in \Phi^*_{\mathcal{F}\cup\mathcal{N}(t)}(P'_0)$.

(ii) $P_0 \stackrel{t}{\Longrightarrow}_a P'_0 \ \wedge\ \mathcal{FI}(t) \subset \mathcal{F} \ \wedge\ \mathcal{BN}(t) \cap \mathcal{F} = \emptyset \quad \Rightarrow \quad P \stackrel{t}{\Longrightarrow}_s P'$ for some $P' \in \Phi^*_{\mathcal{F}\cup\mathcal{N}(t)}(P'_0)$. ∎

This leads us to the following essential lemma. Though this is stated as a lemma, in fact it is the characterisation of $\approx_a$ by $\approx_s$ relative to a set of names.

Lemma 5.27 $\mathcal{F} \supset \mathcal{FN}(P, Q) \ \wedge\ P \approx_a Q \quad \Leftrightarrow \quad \Phi_{\mathcal{F}}(P) \approx_s \Phi_{\mathcal{F}}(Q)$. ∎

With the above property, it is not so difficult to establish the main result.

Theorem 5.28 *Embedding of $\mathcal{L}_a$ in $\mathcal{L}_s$ Suppose $\mathcal{F} \supset \mathcal{FN}(P)$ below.*

$$P \models_{\mathcal{L}_a} A \ \wedge\ \mathcal{FN}(A) \subset \mathcal{F} \quad \Leftrightarrow \quad \Phi_{\mathcal{F}}(P) \models_{\mathcal{L}_s} A \qquad ∎$$

The result suggests the important role of identity receptors in measuring difference between $\approx_s$ and $\approx_a$. The further line of study in this context is axiomatization of $\approx_a$ *via* $\approx_s$. This will convey finer information about relationship between two bisimilarities. The study in this line would also involve topics on some efficiency notion in the setting of asynchronous communication.

6 Conclusion

We thus have two weak bisimilarities for a single calculus, each of which has internal consistency and the special notion of observation, and moreover with equivalent expressive power. While intuitive argument for or against each semantic theory is possible, it seems hard to give decisive argument that solely selects one and not the other as far as we rely on the notion of observables. For example, Sangiorgi's "barbed bisimulation" [17] in our context would become the theory based on "synchronous" convergence (which, therefore, does not equate e.g. $\mathcal{I}(a)$ and Λ), and justification of using *this* notion of convergence and not others would be difficult to give. Accordance between Propositions 2.20 (in reduction) and Proposition 4.13 (in bisimulation) seems to tell us $\approx_a$ is more appropriate in capturing term's behaviour. But the argument again is far from decisive, and rather we can take more relativistic standpoint, which is certainly useful (for example, it opens the way to use the asynchronous observables for synchronous calculi such as CCS). But the lack of canonical semantics (not to say models) leaves us in a difficult situation when, e.g. we compare two calculi in their expressive power: surely *some* canonical semantic framework is essential in such a situation. Specifically we feel it embarrassing that ν-calculus, which we intend to study as a possible basic formalism of concurrency, can have two different semantics (note that our development so far does not tell us whether $\approx_a$, which indeed seems *better* than $\approx_s$, is actually *the best* or not, whatever the criteria might be). Thus we regard the pursuit for a new universal construction in this respect is called for, which, we hope, our sequel to the present exposition to appear soon, will give at least a partial answer. We especially would like to emphasize that in some cases it is quite difficult to decide what is the observable behaviour of a given program, e.g. parallel procedural languages. In such cases, the relative view based on observables may tend to give us an *ad hoc* answer, at least in the first place.

Acknowledgements

Discussions with Professor Robin Milner concerning concurrency in July 1991 meant him much for which he is grateful. Thanks to Nobuko Yoshida for suggestions.

References

[1] Abramsky, S., Observational Equivalence as a Testing Equivalence, *Theoretical Computer Science*, 53, 1987.

[2] Bergstra, J., Klop, J., and Tucker, J., Process Algebra with Asynchronous Communication Mechanisms, In *Seminar on Concurrency*, LNCS 197, 1984, Springer Verlag.

[3] de Boer, F., Kok, J., Palamidessi, C., and J.J.M.M. Rutten. The Failure of failures in a paradigm for asynchronous communication. In *CONCUR '91*, LNCS, Springer-Verlag, 1991.

[4] Boudol, Towards the Lambda Calculus for Concurrent Computation. In *Proc. TAPSOFT 1989*, LNCS 351, Springer Verlag, 1989.

[5] Boudol, *Asynchrony and π-calculus*, a typescript, March 1992.

[6] Hewitt, C., *Viewing Control Structures as Patterns of Passing Messages.* Artificial Intelligence, 1977.

[7] Hennssy and Milner, *Algebraic Laws for Nondeterminism and Concurrency*, Journal of ACM, Vol 32, No. 1, 1985.

[8] Hindley,J.R., and Seldin, J.P., *Introduction to Combinators and λ-calculus.* Campbridhe university Press, 1986.

[9] Honda, K. and Tokoro, M., An Object Calculus for Asynchronous Communication, In: *Proc. of European Conference on Object-Oriented Programming*, LNCS, Springer-Verlag, July 1991. Extended version to appear as a Keio CS report, Novemver 1992.

[10] Honda, K., *Notes on Soundness of Mapping from π-calculus to ν-calculus*, A typescript, April 1992.

[11] Honda, K., *On Interaction Types*, Submitted, September 1992.

[12] Milner, R., *Calculus of Communicating Systems.* LNCS 92, Springer-Verlag, 1980.

[13] Milner, R., Parrow, J.G. and Walker, D.J., *A Calculus of Mobile Processes. Part I/II.* ECS-LFCS-89-85/86, Edinburgh University, 1989

[14] Milner, R., Parrow, J.G. and Walker, D.J., *Modal Logics for Mobile Processes.* ECS-LFCS-91, Edinburgh University, 1991.

[15] Milner, R., *Communication and Concurrency.* Prentice Hall, 1989.

[16] Milner, R., Functions as Processes. In *Automata, Language and Programming*, LNCS 443, 1990. The extended version under the same title as *Rapports de Recherche No.1154*, INRIA-Sophia Antipolis, February 1990.

[17] Milner, R., *Polyadic π-calculus: A Tutorial.* ECS-LFCS-91, Edinburgh University, 1991.

[18] Nielsen and Engberg, *A Calculus of Communicating Systems with Label Passing.* Research Report DAIMI PB-208, Computer Science Department, University of Aarhus, 1986.

[19] Park, D., *Concurrency and Automata on Infinite Sequences.* LNCS 104, Springer-Verlag, 1980.

Appendix: Proofs

In this appendix, proofs or proof outlines of propositions, lemmas, and theorems (except the trivial ones) in the main part of the paper are shown, section by section.

Section 2

All proofs are easy. For Lemmas 2.10 and 2.14, and Proposition 2.15, the proofs are mechanical. For Proposition 2.20 use a normal form and reason in terms of one-step reduction.

Section 3

Before Lemma 3.13, all proofs are easy and omitted. We begin with the proof for Lemma 3.13.

Lemma 3.13 (α-conversion on weak synchronous transition).
We use induction on t.

Base step. We have $t = \varepsilon$. Then $t' = \varepsilon$ hence $Q \stackrel{def}{=} Q\{t\backslash\backslash t'\} \stackrel{def}{=} P$.

Inductive Step. As the trailing τ transitions are not substantial by Lemma 3.9, we suppose $tl \equiv_\alpha t'l'$ with $P \stackrel{t}{\Longrightarrow}_s Q_0 \stackrel{l}{\longrightarrow}_s Q$ and $P \stackrel{t'}{\Longrightarrow}_s Q_0\{t\backslash\backslash t'\} \doteq Q_0$. We note:

a: $l' \equiv_\alpha l\{t\backslash\backslash t'\}$
b: $\mathcal{BN}(t'l') \cap (\mathcal{FN}(P) \cup \mathcal{N}(tl)) = \emptyset$.
c: $\mathcal{BN}(l') \not\subset \mathcal{FN}(Q_0\{t\backslash\backslash t'\})$.

Note **c** follows from **b**, regularity of $t'l'$, and Lemma 3.6. If $\mathcal{BN}(l') = \emptyset$ the result follows easily. Suppose $\mathcal{BN}(l) \neq \emptyset$. Now we have the following reasoning.

$$\begin{array}{lcll}
Q_0 \stackrel{l}{\longrightarrow}_s Q & \Rightarrow & Q_0\{t\backslash\backslash t'\} \stackrel{l\{t\backslash\backslash t'\}}{\longrightarrow}_s Q[z/\pi_2(l)]\{t\backslash\backslash t'\}[\pi_2(l)/z] & (\mathbf{b}, \text{Lemma 3.9 (ii)}) \\
 & \Rightarrow & Q_0\{t\backslash\backslash t'\} \stackrel{l'}{\longrightarrow}_s Q[z/\pi_2(l)]\{t\backslash\backslash t'\}[\pi_2(l)/z][\pi_2(l')/\pi_2(l)] & (\mathbf{a}, \mathbf{c}, \text{Lemma 3.8}) \\
 & \Rightarrow & Q_0\{t\backslash\backslash t'\} \stackrel{l'}{\longrightarrow}_s Q[z/\pi_2(l)]\{t\backslash\backslash t'\}[\pi_2(l')/z] & (\pi_2(l) \notin \mathcal{BN}(t') \text{ by } \mathbf{b}) \\
 & \Rightarrow & Q_0\{t\backslash\backslash t'\} \stackrel{l'}{\longrightarrow}_s Q\{tl\backslash\backslash t'l'\} & (\text{Definition 3.11}) \\
 & \Rightarrow & P \stackrel{t'l'}{\Longrightarrow}_s Q\{tl\backslash\backslash t'l'\} & (P \stackrel{t'}{\Longrightarrow}_s Q_0\{t\backslash\backslash t'\})
\end{array}$$

Note $Q\{tl\backslash\backslash t'l'\} \doteq Q$ is immediate from **b**. ■

For the next Lemma 3.14, apply Lemma 3.9 repeatedly.

Then we show the proof of Proposition 3.21, which is more cumbersome than might have been imagined.

Proposition 3.21 (characterisation of $\approx_s$).

(1) For "if" direction, we show $\approx_s$ enjoys the closure property defined in Proposition 3.21. Suppose $P \approx_s Q$. If $P \stackrel{l}{\longrightarrow}_s P'$ then $P \stackrel{\hat{l}}{\Longrightarrow}_s P'$ where $\hat{l}$ is sound w.r.t. Q (regularity is vacuously satisfied), hence $Q \stackrel{\hat{l}}{\Longrightarrow}_s Q' \approx_s P'$, concluding, with the symmetry of $\approx_s$, the proof.

(2) To prove "only if" direction, we show $\approx'_s$ is a synchronous bisimulation. Suppose $P \approx'_s Q$ and $P \stackrel{t}{\Longrightarrow}_s P'$ which is regular, and is sound w.r.t. P and Q. If $t = l_1..l_n$ we can write:

$$P \stackrel{\varepsilon}{\Longrightarrow}_s P_{1a} \stackrel{l_1}{\longrightarrow}_s P_{1b} ... \stackrel{\varepsilon}{\Longrightarrow}_s P_{na} \stackrel{l_n}{\longrightarrow}_a P_{nb} \stackrel{\varepsilon}{\Longrightarrow}_s P'$$

Then we show, for any $i \leq n$, we can construct a sequence

$$Q \stackrel{\varepsilon}{\Longrightarrow}_s Q_{1a} \stackrel{l_1}{\Longrightarrow}_s Q_{1b} ... \stackrel{\varepsilon}{\Longrightarrow}_s Q_{ia} \stackrel{l_i}{\Longrightarrow}_s Q_{ib}$$

where $P_{ib} \approx_s Q_{ib}$ and $l_{i+1}...l_n$ is regular and is sound w.r.t. P_{ib} and Q_{ib}. This establishes, when $i = n$, the desired result by considering the final transition to P', $P_{nb} \stackrel{\varepsilon}{\Longrightarrow}_s P'$.

Initially P and Q, by assumption, satisfy the condition.

For the inductive step, suppose for some $m-1$ with $m \geq 1$ we can construct such a sequence which ends with $Q_{(m-1)b}$. Then $P_{(m-1)b} \approx_s Q_{(m-1)b}$ and w.r.t. these terms $l_m...l_n$ is sound. Now from

$$P_{(m-1)b} \stackrel{\varepsilon}{\Longrightarrow}_s P_{ma}$$

we can easily have some Q_{ma} such that

$$Q_{(m-1)b} \stackrel{\varepsilon}{\Longrightarrow}_s Q_{ma} \approx_s P_{ma}$$

But $\mathcal{FN}(P_{ma}) \subset \mathcal{FN}(P_{(m-1)a})$ by Lemma 3.6 and similarly $\mathcal{FN}(Q_{ma}) \subset \mathcal{FN}(Q_{(m-1)a})$. Hence l_m is sound w.r.t P_{ma} and Q_{ma} so that we have, for some Q_{mb},

$$Q_{ma} \stackrel{l_m}{\Longrightarrow}_s Q_{mb} \approx_s P_{mb}.$$

Now we show $l_{m+1}...l_n$ is sound w.r.t. P_{mb} and Q_{mb}. By Lemma 3.6 and also by regularity of t, we know

$$\mathcal{FN}(P_{mb}) \subset \mathcal{FN}(P_{(m-1)b}) \cup \mathcal{N}(l_m) \ \wedge \ \mathcal{FN}(Q_{mb}) \subset \mathcal{FN}(Q_{(m-1)b}) \cup \mathcal{N}(l_m)$$

from which we have $\mathcal{FN}(P_{mb}, Q_{mb}) \cap \mathcal{BN}(l_{m+1}...l_n) = \emptyset$, concluding the proof. ∎

Propositions 3.22 and 3.24 and Corollary 3.25 are easy. For the statement in Remark 3.26, we note that the only transition which matters is free input (because for others $P \stackrel{l}{\longrightarrow}_s P' \Rightarrow \mathcal{FN}(l) \subset \mathcal{FN}(P)$). For input, if $P \approx''_s Q$ then

$$\begin{array}{rll} P \stackrel{\downarrow av}{\longrightarrow}_s P' \wedge v \notin \mathcal{FN}(P,Q) & \Rightarrow & P \stackrel{\downarrow a(v)}{\longrightarrow}_s P' \qquad \text{(Remark 3.5)} \\ & \Rightarrow & Q \stackrel{\downarrow a(v)}{\Longrightarrow}_s Q' \approx''_s P' \\ & \Rightarrow & Q \stackrel{\downarrow av}{\Longrightarrow}_s Q' \approx''_s P' \quad \text{(Remark 3.5)} \end{array}$$

from this it is easy to verify the statement (by showing $\approx''_s$ is a synchronous bisimulation).

We now go into the proofs of congruence of $\approx_s$. *Proofs for all the lemmas and propositions are listed.*

Proposition 3.27 (*Variant Property.*).
With

$$
\begin{array}{llll}
\mathcal{R}_0 : & \approx_s & & \\
\mathcal{R}_1 : & P\,\mathcal{R}_0\,Q \,\wedge\, v \notin \mathcal{FN}(P,Q) & \Rightarrow & P[v/x]\,\mathcal{R}_1\,Q[v/x] \\
& \vdots & \vdots & \\
\mathcal{R}_{n+1} : & P\,\mathcal{R}_n\,Q \,\wedge\, v \notin \mathcal{FN}(P,Q) & \Rightarrow & P[v/x]\,\mathcal{R}_{n+1}\,Q[v/x].
\end{array}
$$

and take their union. This union turns out to be a bisimulation. We use induction on n and prove:

> if $\langle P,\ Q\rangle \in \mathcal{R}_n$ then whenever $P \xrightarrow{l}_s P'$ which is sound w.r.t. P and Q we have $Q \overset{\hat{l}}{\Longrightarrow}_s Q'$ with $\langle P,\ Q\rangle \in \mathcal{R}_{n+1}$.

Then we use the characterization of $\approx_s$ by Proposition 3.21. Let us write $\sigma \overset{def}{=} [v/x]$ and $\sigma^{-1} \overset{def}{=} [x/v]$.

The initial case is trivial. We assume the case for n and prove the case for $n+1$. Suppose $\langle P,\ Q\rangle \in \mathcal{R}_n$ below.

Case $P\sigma \xrightarrow{\downarrow aw}_s P'$. We have the following reasoning. Note $P\sigma\sigma^{-1} \overset{def}{=} P$.

$$
\begin{array}{llll}
P\sigma \xrightarrow{\downarrow aw}_s P' & \Rightarrow & P\sigma\sigma^{-1} \xrightarrow{(\downarrow aw)\sigma^{-1}}_s P'\sigma^{-1} & \text{(by Lemma 3.9 (ii))} \\
& \Rightarrow & P \xrightarrow{(\downarrow aw)\sigma^{-1}}_s P'\sigma^{-1} & \\
& \Rightarrow & Q \overset{(\downarrow aw)\sigma^{-1}}{\Longrightarrow}_s Q''\mathcal{R}_n P'\sigma^{-1} & \text{(by hypothesis)} \\
& \Rightarrow & Q\sigma \overset{\downarrow aw}{\Longrightarrow}_s Q''\sigma\,\mathcal{R}_{n+1}\,P' & \text{(by Lemma 3.9 (ii))}
\end{array}
$$

Case $P\sigma \xrightarrow{\downarrow a(w)}_s P'$ with w sound with respect to $P\sigma$ and $Q\sigma$. If $w \neq x, v$ we have the same reasoning as above. Else we take some $w' \notin \mathcal{FN}(P,Q) \cup \{x, v\}$. Let $\rho \overset{def}{=} [w'/w]$ and $\rho^{-1} \overset{def}{=} [w/w']$. Then

$$
\begin{array}{llll}
P\sigma \xrightarrow{\downarrow a(w)}_s P' & \Rightarrow & P\sigma \xrightarrow{\downarrow a(w')}_s P'\rho & \text{(by Lemma 3.8)} \\
& \Rightarrow & P \xrightarrow{(\downarrow a(w'))\sigma^{-1}}_s P'\rho\sigma^{-1} & \text{(by Lemma 3.9)} \\
& \Rightarrow & Q \overset{(\downarrow a(w'))\sigma^{-1}}{\Longrightarrow}_s Q'' \approx_s P'\rho\sigma^{-1} & \text{(by } P \approx_s Q\text{)} \\
& \Rightarrow & Q\sigma \overset{\downarrow a(w')}{\Longrightarrow}_s Q''\sigma & \text{(by Lemma 3.9)} \\
& \Rightarrow & Q\sigma \overset{\downarrow a(w)}{\Longrightarrow}_s Q''\sigma\rho^{-1} & \text{(by } w \notin \mathcal{FN}(Q\sigma) \text{ and Lemma 3.9)}
\end{array}
$$

hence the proof is done.

Case $P[v/x] \xrightarrow{\uparrow aw}_s P'$. Similar to the case $\downarrow aw$.
Case $P[v/x] \xrightarrow{\uparrow a(w)}_s P'$. Similar to the case $\downarrow a(w)$. ∎

Proposition 3.28 ($\approx_s$ is an equivalence relation).
Symmetry is by definition, while reflexivity is immediate. So let us check the transitivity. We

construct a hierarchy of relations:

$$\begin{array}{lccc}
\mathcal{R}_0: & \exists Q.\ P \approx_s Q \approx_s R & \Rightarrow & P\mathcal{R}_0 R \\
\mathcal{R}_1: & P\ \mathcal{R}_0\ R\ \wedge\ v \notin \mathcal{FN}(P,R) & \Rightarrow & P[v/x]\ \mathcal{R}_1\ R[v/x] \\
 & \vdots & \vdots & \\
\mathcal{R}_{n+1}: & P\ \mathcal{R}_n\ R\ \wedge\ v \notin \mathcal{FN}(P,R) & \Rightarrow & P[v/x]\ \mathcal{R}_{n+1}\ R[v/x].
\end{array}$$

and show that their union, denoted $\mathcal{R}$, is a bisimulation. The proof proceeds like the variant lemma. Note $\mathcal{R}$ is obviously symmetric and includes $\approx_s$. We use induction on n and prove that, if $P\mathcal{R}_n R$ then $P \xrightarrow{l}_s P'$ where l is sound w.r.t. P and R, implies, for some R', $R \xRightarrow{\hat{l}}_s R'$ with $P'\mathcal{R}_{n+1}R'$.

In the base step, we show the case when $\mathcal{BN}(l) \neq \emptyset$. Let l be sound w.r.t. P and R, $l' \equiv_\alpha l$, and l' is sound w.r.t. all of P, Q, and R.

$$\begin{array}{llll}
P \xrightarrow{l}_s P' & \Rightarrow & P \xRightarrow{l'}_s P'\{l\backslash\backslash l'\} \stackrel{def}{=} P'' & \text{(Lemma 3.8)} \\
 & \Rightarrow & Q \xRightarrow{l'}_s Q'' \approx_s P'' & (P \approx_s Q) \\
 & \Rightarrow & R \xRightarrow{l'}_s R'' \approx_s Q'' & (Q \approx_s R) \\
 & \Rightarrow & R \xRightarrow{l}_s R''\{l'\backslash\backslash l\} \stackrel{def}{=} R' & \text{(Lemma 3.8)}
\end{array}$$

(note the use of Lemma 3.8 in the last step is possible by l being sound w.r.t. P and R). But $P'' \approx_s Q'' \approx_s R''$ and $P' \stackrel{def}{=} P''\{l'\backslash\backslash l\}$ and $R' \stackrel{def}{=} R''\{l'\backslash\backslash l\}$ (the latter again by the soundness of l) hence $P'\mathcal{R}_1 Q'$, concluding the case. Other cases are similar or easier, using the fact that $\mathcal{R}_1 \supset \mathcal{R}_0$ (note in general we have $\mathcal{R}_{i+1} \supset \mathcal{R}_i$).

The inductive step is just similar to the inductive step of the proof of Proposition 3.27 (the variant lemma), so safely omitted. ∎

Lemma 3.29 (parallel composition and restriction).

(i) Take the relation $\{(|x|P,\ |x|Q) \mid P \approx_s Q\} \cup \approx_s$ and check the transition (we use the characterisation of $\approx_s$ in Proposition 3.21). For output and τ transition the reasoning is easy. For input, we have the following reasoning.

$$\begin{array}{llll}
|x|P \xrightarrow{\downarrow av}_s P'' & \Rightarrow & |z|(P[z/x]) \xrightarrow{\downarrow av}_s |z|P' \equiv P''\ \ (z \text{ fresh}) & (\text{by } z \notin \{av\}) \\
 & \Rightarrow & P[z/x] \xrightarrow{\downarrow av}_s P' & \\
 & \Rightarrow & Q[z/x] \xRightarrow{\downarrow av}_s Q' \approx_s P' & \text{(Proposition 3.27)} \\
 & \Rightarrow & |z|(Q[z/x]) \xRightarrow{\downarrow av}_s |z|Q' & (\text{by } z \notin \{av\}) \\
 & \Rightarrow & |x|Q \xRightarrow{\downarrow av}_s |z|Q' & (\mathit{STRUCT} \text{ in Definition 2.17})
\end{array}$$

where $\langle |z|P',\ |z|Q'\rangle \in \mathcal{R}$ as required.

(ii) Take the relation $\{\langle |\tilde{w}|(P,R),\ |\tilde{w}|(Q,R)\rangle \mid P \approx_s Q\}$ where R is an arbitrary term and show that this is a synchronous bisimulation by checking transitions. Input and output are easy. We only show the case for τ transition where a term from P and a term from R forms a redex pair. For example, we assume $P \equiv |v|(\leftarrow av,\ P')$ and $R \equiv |\tilde{y}|(ax.R_0,\ R_1)$ with v fresh and $a \notin \{\tilde{y}\}$. Let $|\tilde{y}|(R'[v/x],\ R'')) \stackrel{def}{=} R'$. Then

$$|\tilde{w}|(P,R)\ \xrightarrow{\tau}_s\ |\tilde{w}v|(P',\ R')$$

now we have the following inferences.

$$\begin{array}{rcll} P \overset{\uparrow a(v)}{\Longrightarrow}_s P' & \Rightarrow & Q \overset{\uparrow a(v)}{\Longrightarrow}_s Q' \approx_s P' & (P \approx_s Q) \\ & \Rightarrow & Q \overset{\varepsilon}{\Longrightarrow}_s |v|(\leftarrow av,\ Q'') \overset{\uparrow a(v)}{\longrightarrow}_s Q'' \overset{\varepsilon}{\Longrightarrow}_s Q' & (\text{since } v \text{ is fresh}) \\ & \Rightarrow & |\tilde{w}|(Q,\ R) \overset{\varepsilon}{\Longrightarrow}_s |\tilde{w}v|(Q',\ R') & \end{array}$$

thus as required. Other cases are easy. ∎

Lemma 3.30 (substitution lemma for $\approx_s$).
We take the relation

$$\mathcal{R} \overset{def}{=} \bigcup_{i \geq 0} \mathcal{R}_i$$

where

$$\begin{array}{lccc} \mathcal{R}_0 : & \approx_s & & \\ \mathcal{R}_1 : & P\ \mathcal{R}_0\ Q & \Rightarrow & P[v/x]\ \mathcal{R}_1\ Q[v/x] \\ & \vdots & & \\ \mathcal{R}_{n+1} : & P\ \mathcal{R}_n\ Q & \Rightarrow & P[v/x]\ \mathcal{R}_{n+1}\ Q[v/x]\ . \end{array}$$

and prove that this is a synchronous bisimulation. We show this by proving that, for any $n \geq 0$, if $(P, Q) \in \mathcal{R}_n$,

$$P \overset{l}{\longrightarrow}_s P' \quad \Rightarrow \quad \exists\, Q' \ s.t.\ (Q \overset{l}{\Longrightarrow}_s Q')$$

with $\langle P,\ Q\rangle \in \mathcal{R}_{n+1}$ and l sound w.r.t. P' and Q'. This implies, via Proposition 3.21, the required result. We omit the symmetric case.

We use the following construction. For $(P, Q) \in \mathcal{R}_n$, let P_0 and Q_0 denote two expressions such that

$$(P_0, Q_0) \in \mathcal{R}_0 (=\approx_s) \ \wedge\ P \overset{def}{=} P_0[v_1/x_1]..[v_n/x_n] \ \wedge\ Q \overset{def}{=} Q_0[v_1/x_1]..[v_n/x_n]\ .$$

We often write $\tilde{\sigma}$ etc. for the substitution.

Case $P \overset{\downarrow av}{\longrightarrow}_s P'$ By Proposition 2.15 $P \equiv |\tilde{w}|(\partial_0, ax.P, \partial_1)$ with $a \notin \{\tilde{w}\}$. Hence $P_0 \equiv |\tilde{w}|(\partial_0', a'x.P_0'', \partial_1')$ where $a', v' \notin \{\tilde{w}\}$ and $P_0\tilde{\sigma} \equiv P$. Hence (using Lemma 3.9)

$$\begin{array}{rcl} P_0 \overset{\downarrow a'v'}{\longrightarrow}_s R & \Rightarrow & Q_0 \overset{\downarrow a'v'}{\Longrightarrow}_s S \ \approx_s\ R \\ & \Rightarrow & Q_0\tilde{\sigma} \overset{(\downarrow a'v')\tilde{\sigma}}{\Longrightarrow}_s S\tilde{\sigma}\ \mathcal{R}_n\ R\tilde{\sigma}\ \equiv\ P' \end{array}$$

But $Q_0\tilde{\sigma} \equiv Q$ and $(R,\ S) \in \mathcal{R}_n$ so the proof is done.

Case $P \overset{\downarrow a(v)}{\longrightarrow}_s P'$ Let $P \equiv |\tilde{w}|(\partial_0, ax.P'', \partial_1)$ with $a \notin \{\tilde{w}\}$. Hence $P_0 \equiv |\tilde{w}|(\partial_0', a'x.P_0'', \partial_1')$ where $a' \notin \{\tilde{w}\}$ and $P_0\tilde{\sigma} \equiv P$. We let $v' \notin \mathcal{N}(\tilde{\sigma})$ with $\mathcal{N}(\tilde{\sigma})$ denoting the set of names appearing in $\tilde{\sigma}$.

$$\begin{array}{rcl} P_0 \overset{\downarrow a'(v')}{\longrightarrow}_s R & \Rightarrow & Q_0 \overset{\downarrow a'(v')}{\Longrightarrow}_s S \ \approx_s\ R \\ & \Rightarrow & Q_0\tilde{\sigma} \overset{(\downarrow a'(v'))\tilde{\sigma}}{\Longrightarrow}_s S\tilde{\sigma}\ \mathcal{R}_n\ R\tilde{\sigma} \end{array}$$

But then $P' \equiv R\tilde{\sigma}[v/v']$ and $Q \equiv Q_0\tilde{\sigma} \overset{\downarrow a(v)}{\Longrightarrow}_s S\tilde{\sigma}[v/v']$, hence as desired.

Case $P \xrightarrow{\uparrow av}_s P'$. Immediately $P \equiv |\tilde{w}|(\partial_0, \leftarrow av, \partial_1)$ with names in $\tilde{w}$ fresh and $\{av\} \cap \{\tilde{w}\} = \emptyset$. Hence $P_0 \equiv |\tilde{w}|(\partial_0', \leftarrow a'v', \partial_1')$ where $a', v' \notin \{\tilde{w}\}$ and $|\tilde{w}|(\partial_0', \leftarrow a'v', \partial_1')\tilde{\sigma} \equiv P$. Therefore

$$\begin{array}{lll} P_0 \xrightarrow{\uparrow a'v'}_s R & \Rightarrow & Q_0 \xRightarrow{\uparrow a'v'}_s S \approx_s R \qquad\qquad (P_0 \approx_s Q_0) \\ & \Rightarrow & Q_0\tilde{\sigma} \xRightarrow{(\uparrow a'v')\tilde{\sigma}}_s S\tilde{\sigma} \ \mathcal{R}_n \ R\tilde{\sigma} \equiv P' \quad \text{(Lemma 3.14)} \end{array}$$

so the proof is done.

Case $P \xrightarrow{\uparrow a(v)}_s P'$. Similarly to the case above, $P \equiv |\tilde{w}|(\partial_0, \leftarrow au, \partial_1)$ with names in $\tilde{w}$ fresh and $a \notin \{\tilde{w}\}$ and $u \in \{\tilde{w}\}$). Hence
$P_0 \equiv |\tilde{w}|(\partial_0', \leftarrow a'u, \partial_1')$ where $a' \notin \{\tilde{w}\}$ and $|\tilde{w}|(\partial_0', \leftarrow a'u, \partial_1')\tilde{\sigma} \equiv P$. Therefore

$$\begin{array}{lll} P_0 \xrightarrow{\uparrow a'(u)}_s R & \Rightarrow & Q_0 \xRightarrow{\uparrow a'(u)}_s S \approx_s R \qquad\qquad (P_0 \approx_s Q_0) \\ & \Rightarrow & Q_0\tilde{\sigma} \xRightarrow{(\uparrow a'(u))\tilde{\sigma}}_s S\tilde{\sigma} \ \mathcal{R}_n \ R\tilde{\sigma} \equiv P'[u/v] \quad \text{(Lemma 3.14)} \\ & \Rightarrow & Q_0\tilde{\sigma} \xRightarrow{\uparrow a(v)}_s (S\tilde{\sigma})[v/u] \ \mathcal{R}_{n+1} \ P' \qquad \text{(Lemma 3.13)} \end{array}$$

So the proof is done.

Case $P \xrightarrow{\tau}_s P'$. Since we can write:

$$P \equiv |\tilde{w}|(\partial_0, \leftarrow av, ay.P, \partial_1) \xrightarrow{\tau}_s |\tilde{w}|(\partial_0, P[v/y], \partial_1) \equiv P' \ .$$

where we assume the pair in the middle is the concerned redex for the τ transition. Hence $P_0 \equiv |\tilde{w}|(\partial_0', \leftarrow a_1v', a_2y.P', \partial_1')$ with $P_0\tilde{\sigma} \equiv P$. Now note, in P_0 above, if $a_1 \in \{\tilde{w}\}$ and $a_2 \notin \{\tilde{w}\}$ then after substitution they never coincide. We have either $a_1 = a_2 = a \in \{\tilde{w}\}$ or $a_1, a_2 \notin \{\tilde{w}\}$.

(1) $a_1 = a_2 = a \in \{\tilde{w}\}$. By Lemma 3.9 easily there is Q' such that $Q \xRightarrow{\varepsilon}_s Q'$ with $\langle P', Q'\rangle \in \mathcal{R}_n$.

(2) $a_1, a_2 \notin \{\tilde{w}\}$. If $v' \in \{\tilde{w}\}$, from P_0 we have:

$$|\tilde{w}|(\partial_0', \leftarrow a_1v', a_2y.P', \partial_1') \xrightarrow{\uparrow a_1(v')}_s \xrightarrow{\downarrow a_2v'}_s |\tilde{w}\backslash v'|(\partial_0', P'[v'/y], \partial_1') \stackrel{def}{=} R \ .$$

This tells us

$$Q_0 \xRightarrow{\uparrow a_1(v')}_s \xRightarrow{\downarrow a_2v'}_s S \approx_s R$$

W.l.o.g. we let $v' \notin \mathcal{N}(\tilde{\sigma}) \cup \mathcal{N}(P, Q)$. Then we have

$$Q_0\tilde{\sigma} \xRightarrow{\uparrow a(v')}_s \xRightarrow{\downarrow av'}_s S\tilde{\sigma} \approx_s R\tilde{\sigma}$$

Hence, coupled with Lemma 3.29,

$$Q \xRightarrow{\varepsilon}_s |v'|S\tilde{\sigma} \approx_s |v'|R\tilde{\sigma}$$

which establishes the desired result. Similarly in the case $v \notin \{\tilde{w}\}$. ∎

Proposition 3.31 ($\approx_s$ under prefix).
We take the relation

$$\mathcal{R} = \{\langle ax.P, \ ax.Q\rangle \mid P \approx_s Q\} \cup \approx_s$$

Then the only case we should consider is input, and the result is immediately from Lemma 3.30. ∎

Lemma 3.32 (replication lemma for $\approx_s$).
We show the symmetric closure of $\{\langle(!\,ax.P, R),\ (!\,ax.P, ax.P, R)\rangle \ \mid R \text{ arbitrary}\} \cup \equiv$ is a synchronous bisimulation. The case for the output is obvious. Among other cases, we only show the case for free input. Remaining cases are similar. Assume $(!\,ax.P, R) \xrightarrow{\downarrow av}_s (!\,ax.P, R')$. But if $R \xrightarrow{\downarrow av}_s R'$ the result trivially holds. If not, we should have

$$(!\,ax.P, R) \equiv (ax.(P, !\,ax.P), R) \ \xrightarrow{\downarrow av}_s \ (!\,ax.P, P[v/x], R)$$

but then

$$(!\,ax.P, ax.P, R) \ \xrightarrow{\downarrow av}_s \ (!\,ax.P, P[v/x], R)$$

hence done. ∎

Proposition 3.33 (under replication).
We show the relation $\{\langle(!\,ax.P, R),\ (!\,by.Q, S)\ \rangle\rangle \mid ax.P \approx_s by.Q \wedge R \approx_s S\}$ is a synchronous bisimulation. Output transitions are easy. For τ-transition, only the one between $!\,ax.P$ and R is not immediate. Hence assume $(!\,ax.P, R) \xrightarrow{\tau}_s (!\,ax.P, R')$. Then

$$(ax.P, R) \ \xrightarrow{\tau}_s \ R'$$

therefore (by $(ax.P, R) \approx_s (by.Q, S)$ and Lemma 4.18) there is S' such that

$$(by.Q, S) \ \overset{\varepsilon}{\Longrightarrow}_s \ S' \ \approx_a \ R'$$

which means

$$(!\,by.Q, S) \ \overset{\varepsilon}{\Longrightarrow}_s \ (!\,by.Q, S')$$

as required. For input, suppose $(!\,ax.P, R) \xrightarrow{\downarrow av}_s (!\,ax.P, P[v/x], R)$ but then

$$by.Q \ \overset{\downarrow av}{\Longrightarrow}_s \ Q' \ \approx_s \ P[v/x]$$

hence

$$!\,by.Q, S \ \overset{\downarrow av}{\Longrightarrow}_s \ !\,by.Q, Q', S$$

but by Proposition 3.29, $\langle(!\,ax.P, P[v/x], R),\ (!\,by.Q, Q', S)\rangle$ as desired. The case for bound input is similar. ∎

This concludes the proof of congruence of $\approx_s$.

Section 4

Proposition 4.3.
Direct from the definition. For the last subset relation, we have a transition e.g.

$$|\tilde{w}|(\Gamma, ax.P, \Delta) \ \xrightarrow{\downarrow av}_a \xrightarrow{\tau}_a \ |\tilde{w}|(\Gamma, P[v/x], \Delta)$$

which effectively captures $\xrightarrow{\downarrow av}_s$. Similarly for bound input. ∎

Lemmas 4.8, 4.9 can be proved analogously to Lemmas 3.13, 3.14. Similarly the proof for Proposition 4.11 parallels that of Proposition 3.21. Now we prove Proposition 4.12.

Proposition 4.12 (*Inclusion of* $\approx_s$ *in* $\approx_a$).
We show $\approx_s$ is an asynchronous bisimulation. Suppose $P \approx_s Q$. Then by Proposition 4.3 the only transition we should check is input. Suppose $P \xrightarrow{\downarrow av}_a P'$. Then $P' \equiv (P, \leftarrow av)$. But we have $Q \xrightarrow{\downarrow av}_a (Q, \leftarrow av) \approx_s (P, \leftarrow av)$. Similarly for bound input. ∎

Example 4.13 is proved by taking the relation e.g. $\{\langle(\mathcal{I}(a), \partial), (\Lambda, \partial)\rangle\}$ where ∂ is composition of arbitrary messages and show that its symmetric closure is an asynchronous bisimulation. Proposition 4.14 is direct from Proposition 4.12 and Example 4.13. For Remark 4.15, we construct the relation

$$\mathcal{R} \stackrel{def}{=} \{\langle(P,\ \partial),\ (Q,\ \partial)\rangle \mid P \approx''_a Q\}$$

where ∂ is an arbitrary composition of messages, and show that $\mathcal{R}$ is an asynchronous bisimulation. The rest is easy.

Now we prove the congruence of $\approx_a$, sometimes referring to proofs for $\approx_s$.

Proposition 4.16 (composition with arbitrary messages).
By proving that the relation $\mathcal{R} \stackrel{def}{=} \{\langle(P,\partial),\ (Q,\partial)\rangle \mid P{\approx_a}Q\}$ is an asynchronous bisimulation. The only non-trivial case is when $(P,\partial) \xrightarrow{\tau}_a P''$ and one of the redex for this τ transition is in ∂. Then there should be a message, say $\leftarrow av$, in ∂, so that $\partial \equiv (\partial', \leftarrow av)$ and that $P \xrightarrow{\downarrow av}_a (P, \leftarrow av) \xrightarrow{\tau}_a P'$ with $P'' \equiv (P', \partial')$. But because $P \approx_a Q$, we have, for some $Q' \approx_a P'$, $Q \xrightarrow{\downarrow av}_a (Q, \leftarrow av) \xrightarrow{\tau}_a Q'$. Thus again $\langle(P', \partial'),\ (Q', \partial')\rangle \in \mathcal{R}$ as required. ∎

Proposition 4.17 ($\approx_a$ is an equivalence relation). Symmetry is by definition, while reflexivity is immediate. The transitivity is checked in the similar way as in Proposition 3.28, taking a hierarchy of relations:

$$\begin{array}{lccc}
\mathcal{R}_0 : & \exists Q.\ P \approx_a Q \approx_a R & \Rightarrow & P \mathcal{R}_0 R \\
\mathcal{R}_1 : & P\ \mathcal{R}_0\ R\ \wedge\ v \notin \mathcal{FN}(P,R) & \Rightarrow & P[v/x]\ \mathcal{R}_1\ R[v/x] \\
 & \vdots & \vdots & \\
\mathcal{R}_{n+1} : & P\ \mathcal{R}_n\ R\ \wedge\ v \notin \mathcal{FN}(P,R) & \Rightarrow & P[v/x]\ \mathcal{R}_{n+1}\ R[v/x].
\end{array}$$

and show that their union, denoted $\mathcal{R}$, is a bisimulation.

The base case is just as in Proposition 3.28 (we use Lemma 4.5 and Lemma 4.6 instead of Lemma 3.8 and Lemma 3.9, respectively).

For the inductive step, the cases for input are easily proved by showing $P\mathcal{R}_n Q$ implies $(P, \leftarrow av)\mathcal{R}_{n+2}(Q, \leftarrow av)$, using Lemma 2.10 (vi), while the case for τ transition is easy by Lemma 4.6 (i). The output cases can be proved exactly as we did in Proposition 3.27 (done for input but reasoning is just similar), using corresponding lemmas. ∎

Proposition 4.18 (parallel composition and restriction).

(i) Take the relation $\{(|x|P,\ |x|Q) \mid P \approx_a Q\} \cup \approx_a$. For input, we use Proposition 4.16 above.

(ii) Take the relation $\{(|\tilde{w}|(P, R),\ |\tilde{w}|(Q, R)) \mid P \approx_a Q\}$ where R is an arbitrary term. We use (i) above for τ transitions. Finally we take the case $\tilde{w} = \varepsilon$. ∎

The following proof, though with many similarities, still has subtle differences from that of Lemma 3.30.

Lemma 4.19 (substitution lemma for $\approx_a$).
We take the relation

$$\mathcal{R} \stackrel{def}{=} \bigcup_{i\geq 0} \mathcal{R}_i$$

where

$$\begin{array}{llcl}
\mathcal{R}_0 : & \approx_a & & \\
\mathcal{R}_1 : & P\,\mathcal{R}_0\,Q & \Rightarrow & P[v/x]\,\mathcal{R}_1\,Q[v/x] \\
& \vdots & & \\
\mathcal{R}_{n+1} : & P\,\mathcal{R}_n\,Q & \Rightarrow & P[v/x]\,\mathcal{R}_{n+1}\,Q[v/x]\ .
\end{array}$$

and prove that this is an asynchronous bisimulation. Note for any n we have $\mathcal{R}_{n+1} \supset \mathcal{R}_n$. As $(P,Q) \in \mathcal{R}_1 \subset \mathcal{R}$, this shows that
$(P,Q) \in \mathcal{R}_1$ implies $P \approx_a Q$. We show this by proving that, for any $n \geq 0$, if $(P,Q) \in \mathcal{R}_n$,

$$P \stackrel{l}{\longrightarrow}_a P' \quad \Rightarrow \quad \exists\, Q'\ s.t.\ (Q \stackrel{l}{\Longrightarrow}_a Q')$$

where l is sound w.r.t. P and Q, with $\langle P,\ Q\rangle \in \mathcal{R}_{n+2}$. W.l.o.g. we can omit the symmetric case.

Similarly to Lemma 3.30, we use the following construction: for $\langle P,\ Q\rangle \in \mathcal{R}_n$, let P_0 and Q_0 denote two expressions such that

$$\langle P_0,\ Q_0\rangle \in \mathcal{R}_0(=\approx_a) \ \wedge\ P \stackrel{def}{=} P_0[v_1/x_1]..[v_n/x_n] \ \wedge\ Q \stackrel{def}{=} Q_0[v_1/x_1]..[v_n/x_n]\ .$$

We sometimes write $P_0\tilde{\sigma}$ etc. for above construction, $\tilde{\sigma}$ denoting the sequence of substitutions.
Case $P \stackrel{\downarrow av}{\longrightarrow}_a P'$ **or** $P \stackrel{\downarrow a(v)}{\longrightarrow}_a P'$ **.** Immediately $P' \equiv (P, \leftarrow av)$. Similarly $Q \stackrel{\downarrow av}{\longrightarrow}_a (Q, \leftarrow av)$. Using (iv) and (vi) of Lemma 2.10 repeatedly, we have

$$\begin{array}{rcl}
(P, \leftarrow av) & \stackrel{def}{=} & (P_0[v_1/x_1]..[v_n/x_n], \leftarrow av) \\
& \stackrel{def}{=} & (P_0, \leftarrow av[z_1/x_i][z_2/x_j])[v_1/x_1]..[v_n/x_n][x_i/z_1][x_j/z_2]
\end{array}$$

where $z_1, z_2 \notin \mathcal{FN}(P) \cup \mathcal{FN}(Q) \cup \{x_1,..,x_n\}$ and that for any x_k with $1 \leq k \leq n$ and $k \neq i,j$, $x_k \notin \{a, v\}$, and similarly for Q (note we have $n+2$ substitutions outside the parenthesis above). But by Proposition 4.18 we have

$$(P_0, \leftarrow av[z_1/x_i][z_2/x_j]) \ \approx_a \ (Q_0, \leftarrow av[z_1/x_i][z_2/x_j])$$

which shows $((P, \leftarrow av),\ (Q, \leftarrow av))$ is in $\mathcal{R}_{n+2}$.

Case $P \stackrel{\uparrow av}{\longrightarrow}_a P'$. By Proposition 2.15 $P \equiv |\tilde{w}|(\partial_0, \leftarrow av, \partial_1)$ with $a \notin \{\tilde{w}\}$. Hence $P_0 \equiv |\tilde{w}|(\partial_0', \leftarrow a'v', \partial_1')$ where $a', v' \notin \{\tilde{w}\}$ and $|\tilde{w}|(\partial_0', \leftarrow a'v', \partial_1')\tilde{\sigma} \equiv P$. Hence

$$\begin{array}{rcl}
P_0 \stackrel{\uparrow a'v'}{\longrightarrow}_a R & \Rightarrow & Q_0 \stackrel{\uparrow a'v'}{\Longrightarrow}_a S \ \approx_a\ R \\
& \Rightarrow & Q_0\tilde{\sigma} \stackrel{(\uparrow a'v')\tilde{\sigma}}{\Longrightarrow}_a S\tilde{\sigma}\,\mathcal{R}_n\,R\tilde{\sigma} \ \equiv\ P'
\end{array}$$

so the proof is done.

Case $P \xrightarrow{\uparrow a(v)}_a P'$. Let $P \equiv P_0\tilde{\sigma}$ and $Q \equiv Q_0\tilde{\sigma}$ with $P_0 \approx_a Q_0$. By Lemma 4.5, we let $v \notin \mathcal{N}(\tilde{\sigma}) \cup \mathcal{N}(P, Q)$ without loss of generality. The rest exactly follows the path in the case of $\uparrow av$.

Case $P \xrightarrow{\tau}_a P'$. Almost similar to the corresponding proof of Lemma 3.30 except we need τ transition in some places in the way: $P \xrightarrow{\downarrow av}_s P' \;\Rightarrow\; P \xrightarrow{\downarrow av}_a \xrightarrow{\tau}_a P'$. ∎

Proposition 4.20 ($\approx_a$ under prefix).
We take the relation

$$\mathcal{R} = \{\langle (ax.P, \partial),\ (ax.Q, \partial)\rangle \mid P \approx_s Q\} \cup \approx_a$$

For input and output the resulting pair is still in the closure. For τ transition we should consider the pair $\langle (P[v/x], \partial'),\ (Q[v/x], \partial')\rangle$ where $P \approx_a Q$. But we know by Lemma 4.19 and 4.18 that they are in $\approx_a$, hence in $\mathcal{R}$. Thus $\mathcal{R}$ is a bisimulation. ∎

Proposition 4.22 ($\approx_a$ under replication).
We show the relation $\{\langle (!\,ax.P, R),\ (!\,by.Q, S)\ \rangle \mid ax.P \approx_a by.Q \wedge R \approx_a S\}$ is an asynchronous bisimulation. Input and output are obvious, and the only non-trivial case which is τ transition between $!\,ax.P$ and R, is just the same as in the proof of Lemma 3.33. ∎

This concludes the proofs of congruence of $\approx_a$.

Section 5

In Section 5, the first non-trivial proof is the variant lemma. This we prove in the following.

Lemma 5.10 (Variant lemma for statement).
We argue by induction on the structure of A. We will prove the case for $\models_{\mathcal{L}_s}$; The proof for $\models_{\mathcal{L}_a}$ precisely follows the same line.

(i) **Case** $P \models_{\mathcal{L}_s} \langle t\rangle A'$. By assumption, there are some t_0, A_0, P_0 with $\langle t_0\rangle A_0 \equiv_\alpha \langle t\rangle A$ such that $P \overset{t_0}{\Longrightarrow}_s P_0 \models_{\mathcal{L}_a} A_0$. Now we have the following sequence of inferences. Suppose below that $t' \equiv_\alpha t_0 \wedge \mathcal{BN}(t') \cap \{v, x\} = \emptyset$.

$$\begin{array}{lll}
P \overset{t_0}{\Longrightarrow}_s P_0 \models_{\mathcal{L}_s} A_0 & & \\
\quad\Rightarrow & P \overset{t'}{\Longrightarrow}_s P_0\{t_0\backslash\backslash t'\}(\overset{def}{=} P'') \models_{\mathcal{L}_s} A_0\{t_0\backslash\backslash t'\}(\overset{def}{=} A'') & \text{(Lemma 3.13)} \\
\quad\Rightarrow & P[v/x] \overset{t'[v/x]}{\Longrightarrow}_s P''[v/x] \models_{\mathcal{L}_s} A''[v/x] & \text{(Lemma 3.14, } \mathcal{BN}(t') \cap \{v, x\} = \emptyset) \\
\quad\Rightarrow & P[v/x] \models_{\mathcal{L}_s} (\langle t'\rangle A'')[v/x] \equiv_\alpha (\langle t\rangle A')[v/x] &
\end{array}$$

(ii) **Case** $P \models_{\mathcal{L}_s} \neg A'$. Suppose, in spite of the assumption, we have not $P[v/x] \models_{\mathcal{L}_s} \neg A'[v/x]$. But then

$$\begin{array}{llll}
\text{not } P[v/x] \models_{\mathcal{L}_s} \neg A'[v/x] & \Rightarrow & P[v/x] \models_{\mathcal{L}_s} A'[v/x] & \text{(by definition)} \\
& \Rightarrow & P[v/x][x/v] \models_{\mathcal{L}_s} A'[v/x][x/v] & \text{(induction hypothesis)} \\
& \Rightarrow & P \models_{\mathcal{L}_s} A' & (v \notin \mathcal{FN}(P) \cup \mathcal{FN}(A'))
\end{array}$$

hence contradiction.

(iii) **Case** $A \overset{def}{=} \bigwedge_{i\in I} A_i$. Straightforward. ∎

Now we prove the following result.

Lemma 5.11
We have the following inference.

$$\begin{array}{llll} P \models_{\mathcal{L}_{a,s}} \langle t\rangle A & \Rightarrow & P \stackrel{t'}{\Longrightarrow}_{a,s} P' \models_{\mathcal{L}_{a,s}} A' \ \wedge\ \langle t'\rangle A' \equiv_\alpha \langle t\rangle A & \text{(by definition)} \\ & \Rightarrow & P \stackrel{t}{\Longrightarrow}_{a,s} P'\{t\backslash\backslash t'\} \models_{\mathcal{L}_{a,s}} A'\{t\backslash\backslash t'\} & \text{(Lemmas 3.13,3.14,4.8,4.9,5.10)} \end{array}$$

hence as required. ∎

The proof for modal characterisation theorem is shown incorporating the proof for Proposition 5.13 (i.e. for $\Longleftarrow$ we prove that of Proposition 5.13).

Theorem 5.12 (modal characterisation) and **Proposition** 5.13.
We only show the case for (i). (ii) exactly follows using corresponding lemmas.

($\Longrightarrow$) We use induction on the structure of formulae.

Base step. The only formula we have is the empty conjunction, and all terms satisfy this.

Inductive step. Suppose the proposition holds for formulae with no more than a certain length, yet a certain formula, say A, whose proper sub formulae are all within the assumed length, violates the property, i.e. $P \approx_s Q$ but

$$P \models_{\mathcal{L}_s} A \ \text{ but not } \ Q \models_{\mathcal{L}_s} A.$$

We inspect according to the structure of A.
If $A = \neg A'$, then not $P \models_{\mathcal{L}_s} A'$ but $Q \models_{\mathcal{L}_s} A'$. This is contradiction.
If $A = \bigwedge_{i\in I} A_i$ then similarly contradiction.
If $A = \langle t\rangle A'$ then for some t' and A'' with $\langle t\rangle A' \equiv_\alpha \langle t'\rangle A''$ we have $P \stackrel{t'}{\Longrightarrow}_s P' \models_{\mathcal{L}_s} A'$. Thus for some Q' $Q \stackrel{t'}{\Longrightarrow}_s Q' \approx_s P'$. But then by inductive hypothesis $Q' \models_{\mathcal{L}_s} A'$, hence $Q \models_{\mathcal{L}_s} A$, contradicting our assumption. This concludes the proof for $\Longrightarrow$.

($\Longleftarrow$) We show

$$\mathcal{R} \stackrel{def}{=} \{\langle P,\ Q\rangle \mid P \models_{\mathcal{L}_s} A \ \Leftrightarrow\ Q \models_{\mathcal{L}_s} A \ \text{ for } A \text{ with } \mathcal{FN}(A) \subset \mathcal{FN}(P,Q)\}$$

is a weak synchronous bisimulation using the characterisation in Remark 3.26. Suppose $\langle P,\ Q\rangle$ is in the relation. We then assume

$$P \stackrel{l}{\longrightarrow}_s P'$$

where l is sound w.r.t. P and Q, and $\mathcal{FN}(l) \subset \mathcal{FN}(P,Q)$. Suppose

$$P' \models_{\mathcal{L}_s} A \ \wedge\ \mathcal{FN}(\langle\hat{l}\rangle A) \subset \mathcal{FN}(P,Q)\ .$$

Then by assumption and Lemma 3.13 we have for some Q' $Q \stackrel{\hat{l}}{\Longrightarrow}_s Q'$. Now we show that there is some Q' as above such that $\langle P',\ Q'\rangle$ is again in the relation. But suppose not. Enumerate Q_i, $i = 1, 2, ..$, such that $Q \stackrel{\hat{l}}{\Longrightarrow}_s Q_i$ (the enumeration is always possible by enumerating the underlying set $\mathbf{N}$). If there is any formula, say A', with $\mathcal{FN}(A') \subset \mathcal{FN}(P', Q_i)$, which is satisfied by Q_i but not by P' then its negation is satisfied by P' but not by Q_i by Lemma 5.9, with $\mathcal{FN}(\neg A') \subset \mathcal{FN}(P', Q_i)$. In this way there should be at least one formula for Q_i such that it is satisfied by P' but not by Q_i. Then we take their conjunction, $\bigwedge_{i \in I} A_i$ (note that no Q_i can satisfy the formula). Then a formula $\langle \hat{l} \rangle \bigwedge_{i \in I} A_i$ cannot be satisfied by Q but can be by P. But by $\mathcal{FN}(l) \subset \mathcal{FN}(P, Q)$ we have:

$$\mathcal{FN}(A_i) \subset \mathcal{FN}(P', Q_i) \subset \mathcal{FN}(P, Q) \cup \mathcal{BN}(l)$$

(the last inclusion is from Lemma 3.6) hence $\mathcal{FN}(\langle \hat{l} \rangle \bigwedge_{i \in I} A_i) \subset \mathcal{FN}(P, Q)$. Therefore it should be satisfied also by Q, which is a contradiction. Hence Q' satisfying the condition can always be found. As the symmetry of $\mathcal{R}$ is obvious, we know $\mathcal{R}$ is a synchronous bisimulation, concluding the proof. ■

Proposition 5.19.

(i) It is enough to show $P \xrightarrow{l}_s P' \ \Rightarrow\ \exists P''.\ [\![P]\!] \stackrel{\hat{l}^\star}{\Longrightarrow}_a P'' \approx_a [\![P']\!]$. with $l = \tau, \downarrow av, \downarrow a(v), \uparrow av, \uparrow a(v)$.

Case $l = \tau$. We show $P \longrightarrow P' \ \Rightarrow\ [\![P]\!] \longrightarrow\longrightarrow |c|[\![P']\!]$ with c fresh. But then

$$\leftarrow av,\ ax.P \ \longrightarrow\ P[v/x]$$

is transformed into

$$az. \leftarrow zv,\ |c|(\leftarrow ac,\ cx.P) \ \longrightarrow\ |c|[\![P[v/x]]\!] \approx_a [\![P[v/x]]\!]$$

as required.

Case $t = \downarrow av$. Then in a normal form

$$P \equiv |\tilde{w}|(\partial,\ ax.R,\ \partial') \xrightarrow{\downarrow av}_s |\tilde{w}|(\partial,\ R[v/x],\ \partial') \equiv P'$$

which can be emulated by

$$P \equiv |\tilde{w}|([\![\partial]\!],\ |c|(\leftarrow ac, cx.[\![R]\!],\ [\![\partial']\!]) \xrightarrow{\uparrow a(c)}_a \xrightarrow{\downarrow cv}_a |\tilde{w}|([\![\partial]\!],\ [\![R]\!][v/x],\ [\![\partial']\!])$$

as required.

The rest is similar.

(ii) The proof needs more detailed analysis of the term structure than (i). We can write:

$$P \ \equiv\ |\tilde{w}|(ay.S^n_, \ \leftarrow bv^m)$$

where $ay.Q^n$ means $a_1 y_1.Q_1, .., a_n y_n.Q_n$ (in the above we omit Λ's without affecting the whole proof). By definition we have:

$$[\![P]\!] \ \equiv\ |\tilde{w}c^n|(\leftarrow ac^n_, \ cy.[\![S]\!]^n_, \ by.(\leftarrow yv)^m)$$

Now by Lemma 5.8 and freshness of $\mathcal{N}^\star(t)$, it is enough to check cases for $len(t) \le 1$.

Case $[\![P]\!] \overset{\varepsilon^\star}{\Longrightarrow}_a Q$. With P' as

$$P \overset{\varepsilon}{\Longrightarrow}_s P' \overset{def}{=} |\tilde{w'}|(ay.{S_1}^{n'}_{,} \ \leftarrow b'{v'}^{m'}_{,} \ S_2^k) \ .$$

It is easy to show

$$\begin{array}{rcl} Q & \equiv & |\tilde{w'}{c'}^{n'}c^k\tilde{g}|(\leftarrow a'{c'}^{n'}_{,} \ c'y.[\![S_1]\!]^{n'}_{,} \ b'y.(\leftarrow yv')^{m'}_{,} \ cy.[\![S_2]\!]^{k}_{,} \ \leftarrow cz^k) \\ & \equiv & |\tilde{w'}{c'}^{n'}\tilde{g}|(\leftarrow a'{c'}^{n'}_{,} \ c'y.[\![S_1]\!]^{n'}_{,} \ b'y.(\leftarrow yv')^{m'}_{,} \ (\lambda y.[\![S_2]\!] \odot z)^k) \\ & \approx_a & [\![P']\!] \end{array}$$

where newly introduced names should be fresh in base expressions, and also names in $\tilde{g}$ do not occur in its scope (to be exact the order of occurrences in scope restriction may not be in the order depicted above but this is irrelevant because of $|xy|P \approx_a |yx|P$). Then just regard Q as R (note $Q \overset{\varepsilon}{\Longrightarrow}_a R$), which, together with Theorem 5.12, concludes the case. Note the result is stronger than the statement itself.

Case $[\![P]\!] \overset{\downarrow av^\star}{\Longrightarrow}_a Q$. If $\mathcal{N}^\star(\downarrow av) \cap \mathcal{FN}(Q) = \emptyset$, the form of Q is the same as the case for $t = \varepsilon$. Suppose not. Then we can write:

$$Q \equiv |\tilde{w'}{c'}^{n'}c^k\tilde{g}|(\leftarrow ev, \ ex.[\![S_1]\!], \ \leftarrow a'{c'}^{n'}_{,}c'y.[\![S_2]\!]^{n'}_{,} \ b'y.(\leftarrow yv')^{m'}_{,} \ cy.[\![S_3]\!]^{k}_{,} \ \leftarrow cz^k)$$

where $\{e\} = \mathcal{N}^\star(\downarrow av)$. Let R be

$$R \equiv |\tilde{w'}{c'}^{n'}c^k\tilde{g}|([\![S_1[v/x]]\!], \ \leftarrow a'{c'}^{n'}_{,}c'y.[\![S_2]\!]^{n'}_{,} \ b'y.(\leftarrow yv')^{m'}_{,} \ cy.[\![S_3]\!]^{k}_{,} \ \leftarrow cz^k)$$

Now we are going to establish:

$$[Q]_{\mathcal{L}_a}^{\overline{\{e\}}} = [R]_{\mathcal{L}_a}^{\overline{\{e\}}}$$

For this purpose we prove the following (more general) claim.

Claim: Suppose

$$U \equiv |\tilde{w}|(ex.W, \leftarrow ev, \partial)$$

where ∂ is a multiset of base terms and $e \notin \{\tilde{w}\}$ is fresh, i.e. occur only in two places explicitly specified. Let U' be

$$U' \equiv |\tilde{w}|(W[v/x], \partial)$$

Then we have

$$[U]_{\mathcal{L}_a}^{\overline{\{e\}}} = [U']_{\mathcal{L}_a}^{\overline{\{e\}}}$$

Proof of the claim.
Take the set of all such pairs of $\langle U, \ U'\rangle$, denoted by $\mathcal{R}$. We argue that, for all such pairs, $U \models_{\mathcal{L}_a} A$ with $e \notin \mathcal{FN}(A)$ if and only if $U' \models_{\mathcal{L}_a} A$, using induction on the structure of A (that is, we suppose that up to a certain size of A, the above holds for all such pairs, and argue for the next step). The base case, the empty conjunction, is trivial.

If $A \stackrel{def}{=} \langle t \rangle A'$, with $e \notin \mathcal{FN}(t) \cup \mathcal{FN}(A')$, w.l.o.g. we assume t is regular and $\mathcal{BN}(t) \cap \mathcal{FN}(U) = \emptyset$. Now it is easy to show (strictly speaking by induction on t)

$$U \stackrel{t}{\Longrightarrow}_a V \wedge e \notin \mathcal{FN}(t) \quad \Rightarrow \quad U' \stackrel{t}{\Longrightarrow}_a V'$$

with *either* $V \equiv V'$ *or* $\langle V,\ V' \rangle \in \mathcal{R}$. If $V \equiv V'$ then easily $V \models_{\mathcal{L}_a} A' \quad \Rightarrow \quad V' \models_{\mathcal{L}_a} A'$. Else, i.e. $\langle V,\ V' \rangle \in \mathcal{R}$, we use induction, so that again $V \models_{\mathcal{L}_a} A \quad \Rightarrow \quad V' \models_{\mathcal{L}_a} A$. This shows $U \models_{\mathcal{L}_a} A \ \Rightarrow\ U' \models_{\mathcal{L}_a} A$. In the reverse case, we have

$$U' \stackrel{t}{\Longrightarrow}_a V' \wedge e \notin \mathcal{FN}(t) \quad \Rightarrow \quad U \stackrel{t}{\Longrightarrow}_a V'$$

hence immediately $U' \models_{\mathcal{L}_a} A \ \Rightarrow\ U \models_{\mathcal{L}_a} A$.

If $A \stackrel{def}{=} \neg A'$, with $e \notin \mathcal{FN}(A')$, we have:

$$\begin{array}{lcll} (U \models_{\mathcal{L}_a} A' \ \Leftrightarrow\ U' \models_{\mathcal{L}_a} A') & \Rightarrow & (\text{not } U \models_{\mathcal{L}_a} A' \ \Leftrightarrow\ \text{not } U' \models_{\mathcal{L}_a} A') & \text{(by hypothesis)} \\ & \Rightarrow & (U \models_{\mathcal{L}_a} \neg A' \ \Leftrightarrow\ U' \models_{\mathcal{L}_a} \neg A') & \text{(by definition)} \end{array}$$

as required.

If $A \stackrel{def}{=} \bigwedge_{i \in I} A_i$, easy by induction.
end of the proof of the claim.

We immediately have: $[Q]^{\overline{\{e\}}}_{\mathcal{L}_a} = [R]^{\overline{\{e\}}}_{\mathcal{L}_a}$. Since $R \approx_a [\![P']\!]$ with

$$P \stackrel{\downarrow av}{\Longrightarrow}_s P' \stackrel{def}{=} |\tilde{w'}|(S_1[v/x],\ \ ay.S_2{}^{n'}_{,} \ \leftarrow b'v'{}^{m'}_{,} \ \ S_3^k)\ .$$

the proof is done.

Case $[\![P]\!] \stackrel{\downarrow a(v)^\star}{\Longrightarrow}_s Q$. Just similar to the case $t = \downarrow av$.

Case $[\![P]\!] \stackrel{\uparrow av^\star}{\Longrightarrow}_s Q$. Then we always have $\mathcal{N}^\star(\downarrow av) \cap \mathcal{FN}(Q) = \emptyset$ and the form of Q is the same as the case for $t = \varepsilon$. The result is immediate.

Case $[\![P]\!] \stackrel{\uparrow a(v)^\star}{\Longrightarrow}_s P''$. Similarly to the case $t = \uparrow av$. ∎

Finally the embedding result is proved.

Theorem 5.20 (The embedding of $\mathcal{L}_s$ in $\mathcal{L}_a$).
We prove the bidirectional implication simultaneously. We use induction on the structure of A.

Case $A \stackrel{def}{=} \langle t \rangle A'$.
$(\Longrightarrow)$. By Lemma 5.11 we safely assume $P \stackrel{t}{\Longrightarrow}_s P' \models_{\mathcal{L}_s} A'$. By Proposition 5.19 we have, for some $R \approx_a [\![P']\!]$ $[\![P]\!] \stackrel{t^\star}{\Longrightarrow}_a R$. By inductive hypothesis $[\![P']\!] \models_{\mathcal{L}_a} A'^\circ$ but by Proposition 5.12 this tells us $R \models_{\mathcal{L}_a} A'^\circ$ therefore we have $[\![P]\!] \models_{\mathcal{L}_a} A^\circ$ as desired.

$(\Longleftarrow)$ We assume $[\![P]\!] \stackrel{t^\star}{\Longrightarrow}_a Q \models_{\mathcal{L}_a} A'^\circ$. By Proposition 5.19 (ii) and by $\mathcal{N}^\star(t) \cap \mathcal{FN}(A^\circ) = \emptyset$, there is some R such that

$$Q \stackrel{\varepsilon}{\Longrightarrow}_a R \ \wedge\ R \models_{\mathcal{L}_a} A'^\circ \wedge R \approx_a [\![P']\!]$$

with $P \overset{t}{\Longrightarrow}_s P'$. By Theorem 5.12 we have $[\![P']\!] \models_{\mathcal{L}_a} A'^{\circ}$. But then the induction hypothesis tells us that $P' \models_{\mathcal{L}_s} A'$, hence $P \models_{\mathcal{L}_s} \langle t \rangle A'$ as required.
Case $A \stackrel{def}{=} \neg A'$.
($\Longrightarrow$) Then not $P \models_{\mathcal{L}_s} A'$. By induction for $\Longleftarrow$ and by Lemma 5.9 this tells us it is not the case that $[\![P]\!] \models_{\mathcal{L}_a} A'^{\circ}$, hence $[\![P]\!] \models_{\mathcal{L}_a} \neg A'^{\circ} \stackrel{def}{=} A^{\circ}$.
($\Longleftarrow$) Similarly.

Case $A \stackrel{def}{=} \bigwedge_{i \in I} A_i$.
($\Longrightarrow$) Then $P \models_{\mathcal{L}_s} A_i$ for $i \in I$. By induction this tells us $[\![P]\!] \models_{\mathcal{L}_a} A_i^{\circ}$ for $i \in I$
($\Longleftarrow$) Similarly. ∎

Now we prove a proposition and a lemma before proving another embedding theorem. The first is the mutual emulation result.

Proposition 5.26.
First we prove:

(i) $\mathcal{I}_2(\tilde{a}) \longrightarrow\!\!\!\!\!\rightarrow R \quad \Rightarrow \quad R \approx_s \mathcal{I}_2(\tilde{a})$.
(ii) $\mathcal{I}_2(\tilde{a}\tilde{b}) \approx_s (\mathcal{I}_2(\tilde{a}), \mathcal{I}_2(\tilde{b}))$

For (i) take a closure $\{\langle P_1,\ P_2\rangle \mid \mathcal{I}_2(\tilde{a}) \longrightarrow\!\!\!\!\!\rightarrow P_i\} \cup \equiv$ while, in (ii), take a closure $\{\langle P_1,\ P_2\rangle \mid P_i \in \Phi^*_{\{\tilde{a}\}}(\mathcal{I}_2(\tilde{a}))$ for some $\tilde{a}\}$. Using them, we prove the following two lemmas. The proofs are mechanical by checking each transition so we omit them.

Lemma 1. With $P \in \Phi^*_{\mathcal{F}}(P_0)$ and $\mathcal{F} \supset \mathcal{FN}(P_0)$,

$$P \overset{l}{\longrightarrow}_s P' \quad \Rightarrow \quad (P_0 \overset{l}{\Longrightarrow}_a P_0') \text{ with } P' \in \Phi^*_{\mathcal{F} \cup \mathcal{N}(l)}(P_0').$$

Lemma 2. With $P \in \Phi^*_{\mathcal{F}}(P_0)$ for any P_0 and $\mathcal{F} \supset \mathcal{FN}(P_0)$,

$$P_0 \overset{l}{\longrightarrow}_a P_0' \ \wedge\ \mathcal{FI}(l) \subset \mathcal{F} \quad \Rightarrow \quad P \overset{l}{\Longrightarrow}_s P' \text{ with } P' \in \Phi^*_{\mathcal{F} \cup \mathcal{N}(l)}(P_0')$$

Then (i) of the main proposition is proved using Lemma 1 above, by induction on structure of t. Here properties of the $\mathcal{I}$ completed term is essential in repeatedly applying the Lemma, i.e. $P \in \Phi^*_{\mathcal{F}}(P_0)$ with $\mathcal{F} \supset \mathcal{FN}(P_0)$, we have $\mathcal{FI}(P) = \mathcal{FN}(P) = \mathcal{F}$ and also

$$P \overset{l}{\longrightarrow}_a P' \quad \Rightarrow \quad P' \in \Phi^*_{\mathcal{F} \cup \mathcal{N}(l)}(P_0')$$

for some P_0' with $\mathcal{F} \cup \mathcal{N}(l) \supset \mathcal{FN}(P_0')$.
Similarly we use Lemma 2 to prove (ii). ∎

The proof below establishes a characterisation of $\approx_a$ by $\approx_s$ relative to a certain set of names.

Lemma 5.27.
($\Longrightarrow$). For some finite set of names $\mathcal{F}$, let us have:

$$\mathbf{R}_{\mathcal{F}} \stackrel{def}{=} \{\langle P,\ Q\rangle \mid P \in \Phi^*_{\mathcal{F}}(P_0),\ Q \in \Phi^*_{\mathcal{F}}(Q_0) \text{ for } P_0, Q_0 \in \mathbf{C}_{\mathcal{F}} \text{ and } P_0 \approx_a Q_0.\}$$

where $\mathbf{C}_{\mathcal{F}} = \{P \mid \mathcal{FN}(P) \subset \mathcal{F}\}$. Then we show that the union of all such relations, $\bigcup_{\mathcal{F} \subset_f \mathbf{N}} \mathbf{R}_{\mathcal{F}}$, where "$\subset_f$" stands for "is a finite subset of", is a synchronous bisimulation. Clearly the pair

$\langle \Phi_{\mathcal{FN}(P,Q)}(P),\ \Phi_{\mathcal{FN}(P,Q)}(Q)\rangle$ is in the closure. Assuming $(P,\ Q) \in \mathbf{R}_{\mathcal{F}}$ for some $\mathcal{F}$, we have a following sequence of inferences. Assume t is regular and sound w.r.t. P and Q.

$$
\begin{array}{lcll}
P \stackrel{t}{\Longrightarrow}_s P' & \Rightarrow & P_0 \stackrel{t}{\Longrightarrow}_a P_0' \text{ with } P' \in \Phi^*_{\mathcal{F}\cup\mathcal{N}(t)}(P_0') & \text{(by Proposition5.26 (i))} \\
 & \Rightarrow & Q_0 \stackrel{t}{\Longrightarrow}_a Q_0' \text{ with } P_0' \approx_a Q_0' & \text{(by } P_0 \approx_a Q_0) \\
 & \Rightarrow & Q \stackrel{t}{\Longrightarrow}_s Q' \text{ with } Q' \in \Phi^*_{\mathcal{F}\cup\mathcal{N}(t)}(Q_0') & \text{(by Proposition5.26 (ii))}
\end{array}
$$

Note that automatically t is sound w.r.t. P_0 and Q_0 because $\mathcal{F} \supset \mathcal{FN}(P_0) \cup \mathcal{FN}(Q_0)$. Finally $(P',\ Q') \in \mathbf{R}_{\mathcal{F}\cup\mathcal{N}(t)} \subset \bigcup_{\mathcal{F}\subset_f \mathbf{N}} \mathbf{R}_{\mathcal{F}}$, and by symmetry of the relation is a synchronous bisimulation, concluding the proof.
$(\Longleftarrow)$. Immediate from Lemma 5.24 and transitivity of $\approx_a$. ∎

Finally we prove the main theorem.

Theorem 5.28 (The embedding of $\mathcal{L}_a$ in $\mathcal{L}_s$).
We again prove simultaneously $\Longrightarrow$ and $\Longleftarrow$ directions. We use induction on the structure of A.

Case $A \stackrel{def}{=} \langle t\rangle A'$.
$(\Longrightarrow)$. Let $P \models_{\mathcal{L}_a} A$. W.l.o.g. (cf. Lemma 5.11), we let $\mathcal{BN}(t) \cap \mathcal{FN}(P) = \emptyset$. Hence we have

$$P \stackrel{t}{\Longrightarrow}_a P' \models_{\mathcal{L}_a} A'.$$

By $\mathcal{FI}(t) \subset \mathcal{FN}(t) \subset \mathcal{FN}(P) \subset \mathcal{F}$, we have, for some $P'' \in \Phi^*_{\mathcal{F}\cup\mathcal{N}(t)}(P')$,

$$\Phi_{\mathcal{F}}(P) \stackrel{t}{\Longrightarrow}_s P''$$

Clearly $\mathcal{FN}(A') \subset \mathcal{F} \cup \mathcal{N}(t)$. Thus by inductive hypothesis we have

$$\Phi_{\mathcal{F}\cup\mathcal{N}(t)}(P') \models_{\mathcal{L}_s} A'$$

which, together with Lemma 5.27, leads to

$$P'' \models_{\mathcal{L}_s} A'$$

hence as desired.
$(\Longleftarrow)$. Easy by Lemma 5.25.

Case $A \stackrel{def}{=} \neg A'$.
$(\Longrightarrow)$. We have the following inference.

$$
\begin{array}{lcll}
P \models_{\mathcal{L}_a} A & \Rightarrow & \text{not } P \models_{\mathcal{L}_a} A' & \\
 & \Rightarrow & \text{not } \Phi_{\mathcal{F}}(P) \models_{\mathcal{L}_s} A' & \text{(by induction)} \\
 & \Rightarrow & \Phi_{\mathcal{F}}(P) \models_{\mathcal{L}_s} \neg A' &
\end{array}
$$

$(\Longleftarrow)$. Similarly.

Case $A \stackrel{def}{=} \bigwedge_{i\in I} A_i'$.
For both directions, the proof is easy by induction. ∎

This concludes Appendix.