Making the unobservable, unobservable

Julian Rathke

ECS, *University of Southampton*

Paweł Sobociński [1]

ECS, *University of Southampton*

Abstract

Behavioural equivalences of various calculi for modelling distributed systems differ significantly because the properties which can be observed through interaction depend heavily upon their mode of communication. A typical approach to describing the semantics of communicating processes is to provide a labelled transition system (LTS) which captures the interaction potential of the individual processes within a larger system. In many cases, a natural rendering of this LTS leads to too fine a semantics as unobservability of certain communications is not accounted for.
We propose that a standard approach to augmenting LTSs allows morally unobservable communications to actually be modelled as unobservables in the semantics. This approach derives from a rule initially given by Honda and Tokoro to account for unobservability of reception in the asynchronous π-calculus. We examine the implications of adding such rules to LTS with respect to the proving behavioural equivalences for various synchronisation mechanisms.

Keywords: process algebra, structural operational semantics, observability, simulation, bisimulation

1 Introduction

Semantic descriptions of processes in the process algebra/calculus tradition have come to be dominated by the use of labelled transition systems (LTSs) to describe the nature of the interaction contributed by each participant in a communication. This approach was inspired by early work of Plotkin [11] on structural operational semantics and Milner's definition of LTS semantics for CCS [8].

The use of labels has an interesting artefact. Labelled transition systems now not only capture reduction but also allow programs to be compared for equivalence without the need for examining their behaviour within all enclosing contexts. The *dual purpose* nature of these transition systems does have an unfortunate side-effect in that the labelled transitions, which are useful for describing structurally defined reduction relations, do not always coincide with observability properties of language with respect to program equivalence. This problem is particularly acute

[1] Research partially supported by EPSRC grant EP/D066565/1.

when the underlying communication mechanism between processes is anything less than fully synchronous. In a nutshell, the "natural looking" structural LTSs for a given language may yield too fine an equivalence relation if the labels are used for comparing programs.

A well-known example of this lies in the unobservability of reception in languages featuring asynchronous communication. This issue is studied thoroughly in [5,1] in the setting of the asynchronous π-calculus in which it is demonstrated that (weak) bisimilarity on the structural LTS for the π-calculus does not correctly capture a notion of (weak) observable equivalence in this language. There are two different, but related, approaches to resolving this problem: [1] proposes a variant definition of bisimilarity while [5] proposes an additional, non-structural labelled transition rule to model the unobservability of reception. For technical reasons concerning uniformity of definition and difficulties in establishing transitivity of the variant bisimilarity, we prefer the latter approach by Honda and Tokoro [5]. Their innovation was to blur the observability of reception actions in π-calculus by augmenting the standard LTS with (essentially) the following rule:

$$P \xrightarrow{a?b} P \parallel a!b$$

This rule allows a process which genuinely performs a receive action $P \xrightarrow{a?b} P'$ to be identified with a process that may not be able to perform any actions, let alone a receive action. In fact, augmenting the LTS with the above rule makes the contextually unobservable receive actions in the asynchronous world actually become unobservable in their LTS model. We will refer to rules such as the above as Honda Tokoro ($\mathcal{HT}$) style rules.

In this paper we pursue the idea of Honda and Tokoro by generalising their approach of using additional LTS rules to guarantee the correct unobservability properties in a structurally defined LTS. We do this by identifying how we can routinely add $\mathcal{HT}$ rules to an LTS without compromising the soundness and completeness of its associated (bi)similarity with respect to a simple notion of contextual equivalence. We demonstrate this approach in three different scenarios: fully asynchronous, output asynchronous and fully synchronous communication in CCS-like calculi. Furthermore, we discuss how to specialise the resulting LTSs by only adding the necessary additional unobservability rules – in some calculi some actions are inherently observable and do not require a $\mathcal{HT}$ rule. This analysis is useful because it results in more manageable LTSs. Indeed, it is important to emphasise that this paper does not address the issue of characterising when particular actions are or are not contextually unobservable, as studied in [9]. The focus here is on the general applicability and correctness of the unobservability rules.

2 Structural rules, LTSs and (bi)simulations

We need to recall the standard notion of a labelled transition system. In all of our examples the set of states coincides with the set of processes of the calculus at hand.

Definition 2.1 Suppose that A is a set of *observations* and S is a set of states.

An (A, S)-*labelled transition system* (LTS) is a ternary relation $\mathcal{T} \subseteq S \times A \times S$. We will write $s \xrightarrow{\alpha} s'$ for $(s, \alpha, s') \in \mathcal{T}$.

Traditionally, LTSs defined by a set of SOS rules have been used to:

- define the reduction relation compositionally by considering the structure of terms;
- define the possible "interactions" and reason about the resulting (labelled) pre-orders/ equivalences;
- characterise observational preorders/equivalences by using the aforementioned labelled preorders/equivalences.

There is a natural tension between the first and the third point. The first may require us to observe the intensional structure of the terms in order to characterise the possible reductions. The third requires us to characterise the structure that can be *observed*. Examples of and techniques for relaxing this tension are the remit of this paper.

Our LTSs will be defined using simple SOS rules. Given an LTS $\mathcal{T}$ and a rule ψ, let $\psi(\mathcal{T})$ be the LTS obtained by adding the extra transitions which can be derived using (possibly several applications of) ψ from the existing transitions of $\mathcal{T}$. This is easily defined as a least fixed point. Seen as an endofunction on the class of LTSs, ψ is idempotent. This notation can be extended to sets of rules $\Psi = \{\psi_i\}_{i \in I}$ in the obvious way; proof-theoretically, $\Psi(\mathcal{T})$ denotes the set of transitions that result from the derivations that use the rules in Ψ, assuming the transitions of $\mathcal{T}$ as axioms. When we speak about *the* LTS defined by a set of rules Ψ, we mean the LTS $\Psi(\varnothing)$ where $\varnothing$ is the empty LTS.

Definition 2.2 *Given two (A, S)-LTSs $\mathcal{T}_1$ and $\mathcal{T}_2$, a simulation from $\mathcal{T}_1$ to $\mathcal{T}_2$ is a binary relation $\mathcal{R} \subseteq S \times S$ such that if $P \mathrel{\mathcal{R}} Q$ and $P \xrightarrow{\alpha} P'$ in $\mathcal{T}_1$ then $\exists Q'$ such that $Q \xrightarrow{\alpha} Q'$ in $\mathcal{T}_2$ and $P' \mathrel{\mathcal{R}} Q'$. A simulation $\mathcal{R}$ is a* bisimulation *whenever R^{-1} is also a simulation.*

Given an LTS $\mathcal{T}$, $\precsim_{\mathcal{T}}$ is similarity, the largest simulation, while $\sim_{\mathcal{T}}$ is bisimilarity, the largest bisimulation.

3 Full asynchrony

Consider the following simple language $\mathcal{L}_f$ of finite processes generated by:

$$P \ ::= \ 0 \mid a! \mid a? \mid P \parallel Q \mid \tau$$

We can think of a typical term as a textual representation of a "soup" of processes [2] which interact with each other. We thus ask that parallel composition behave appropriately: a *process* is an equivalence class of terms quotiented by the smallest congruence in which $\parallel$ is associative, commutative and has 0 as identity.

The execution semantics (*i.e.* what a program *does*) is captured by the smallest (unlabelled) transition system $\rightarrow$ that contains $\tau \rightarrow 0$ as well as $a! \parallel a? \rightarrow 0$ (for any a) and is closed under parallel composition. We will refer to the edges in the above transition system as *interactions* or *reductions*.

Assume that an interaction has an observable effect, for concreteness let us say that each interaction generates a certain amount of heat which can be detected by the observer's sophisticated sensors. The external observer cannot see the internals of a soup (the identity and number of its components) but can experiment with it only by:

- introducing new ingredients into the soup;
- measuring the change in heat, *i.e.* observing interactions.

Our most controversial assumption is that time can be slowed down to such an extent that no two reductions can ever happen truly concurrently, that is, the observer can observe the reductions one at a time. This is normally referred to as *interleaving*.

For example, the observer can distinguish between 0 and $a!$; the experiment runs as follows: the observer introduces $a?$ into the both the soups and observes that while the first soup stays at constant temperature, the second soup gets a little bit hotter.

The above ingredients induce a canonical preorder and equivalence on the set of processes: the preorder being the largest (reduction) simulation which is a precongruence (wrt to $\|$) and the equivalence being the largest reduction bisimulation which is a congruence. For us, these two relations capture the greatest reasonable power of an external observer.

Definition 3.1 *Reduction precongruence* is the largest relation $\lesssim$ which:

(i) is a simulation with respect to reduction, *i.e.* if $P \lesssim Q$ and $P \to P'$ then there exists Q' such that $Q \to Q'$ and $P' \lesssim Q'$;

(ii) is stable wrt $\|$, *i.e.* if $P \lesssim Q$ then for any R we have $P \parallel R \lesssim Q \parallel R$.

Definition 3.2 *Reduction congruence* is the largest relation $\simeq$ which:

(i) satisfies conditions (i) and (ii) of Definition 3.1;

(ii) is symmetric.

In the following we will attempt to reason about the contextually defined reduction (pre)congruence by characterising the possible "experiments" (*cf.* the discussion before Definition 3.1) as *labelled* transitions; the procedure (initial state, experiment, final state) will be a transition in an LTS. Roughly, this means that we are in effect replacing "contexts" of Definitions 3.1 and 3.2 with "labels" in a LTS. The goal is to use labelled preorders and equivalences, such as bisimulation, as proof methods for reasoning about reduction (pre)congruence.

The first kind of experiment is the simplest, the observer does not need to do anything and can observe a rise in temperature due to the presence of a τ within the term. Let us call this experiment τ. We can characterise inductively some of the processes for which this experiment is successful:

$$\frac{}{\tau \xrightarrow{\tau} 0}\ (\textsc{Tau}) \qquad \frac{P \xrightarrow{\tau} P'}{P\|Q \xrightarrow{\tau} P'\|Q}\ (\textsc{Tau}\|)$$

Another experiment is the following: the observer introduces an output on a ($a!$) and observes that there is a rise in temperature. Let us name (slightly counterintuitively) this kind of experiment $a?$. The reason for this label is that this is the capability (input) that has been observed. Also this kind of experiment can be characterised inductively:

$$\frac{}{a? \xrightarrow{a?} 0}\ (\textsc{In}) \qquad \frac{P \xrightarrow{a?} P'}{P \| Q \xrightarrow{a?} P' \| Q}\ (\textsc{In}\|)$$

Dually, the observer can introduce an input to observe an output.

$$\frac{}{a! \xrightarrow{a!} 0}\ (\textsc{Out}) \qquad \frac{P \xrightarrow{a!} P'}{P \| Q \xrightarrow{a!} P' \| Q}\ (\textsc{Out}\|)$$

Finally, if the observer's experiments lead her to conclude that there is an input on a in soup A and an output on a in soup B, then a heating should be observed in the combined soup without additional ingredients. Note that the label is the same as our first experiment: this is because, from the observer's point of view, the same (empty) process is provided to $P \parallel Q$ for a change in heat to be observed.

$$\frac{P \xrightarrow{a?} P' \quad Q \xrightarrow{a!} Q'}{P \| Q \xrightarrow{\tau} P' \| Q'}\ (\textsc{Comm})$$

Let $\Phi_f \stackrel{\text{def}}{=} \{(\textsc{Tau}), (\textsc{Tau}\|), (\textsc{In}), (\textsc{In}\|), (\textsc{Out}), (\textsc{Out}\|), (\textsc{Comm})\}$. Consider the LTS $\mathcal{C}_f \stackrel{\text{def}}{=} \Phi_f(\varnothing)$. The following lemma shows that our intuition of this labelled transition system as a "logbook" of certain experiments is correct: for each kind of label α there is a process χ_α which is added by the experimenter to observe a reduction.

Lemma 3.3 *$\chi_{a!} \stackrel{\text{def}}{=} a?$, $\chi_{a?} \stackrel{\text{def}}{=} a!$ and $\chi_\tau \stackrel{\text{def}}{=} 0$. $P \xrightarrow{\alpha} P'$ implies $P \parallel \chi_\alpha \rightarrow P'$.* □

For τ-labelled transitions, the other direction also holds.

Lemma 3.4 *$P \xrightarrow{\tau} P'$ iff $P \rightarrow P'$.* □

Indeed, reasoning on $\mathcal{C}_f$ is sound wrt to observation:

Lemma 3.5 (Soundness of $\mathcal{C}_f$) *$\precsim_{\mathcal{C}} \subseteq \lesssim$ and $\sim_{\mathcal{C}} \subseteq \simeq$.*

Proof. It suffices to show that $\precsim$ is a reduction simulation and stable under parallel composition. It is a reduction simulation because it is a τ-simulation and τs coincide with reductions (Lemma 3.4). To show that it is stable under $\|$, it suffices to show that $\{(P \parallel R, Q \parallel R) \mid P \precsim Q\}$ is a simulation. This is easily checked by cases, the most interesting being $P \parallel R \xrightarrow{\tau} P' \parallel R'$ where, for some a, $P \xrightarrow{a!} P'$ and $R \xrightarrow{a?} R'$. Then, by assumption, $\exists Q'. Q \xrightarrow{a!} Q' \wedge P' \precsim Q'$. But then $Q \parallel R \xrightarrow{\tau} Q' \parallel R'$ which is in the relation. The proof for the case of bisimulation and reduction congruence proceeds similarly. □

However, the reverse inclusion (completeness) does not hold. The LTS allows us to distinguish processes which are observationally not distinguishable. For example:

Lemma 3.6 *Let $P_1 \stackrel{\text{def}}{=} a? \parallel a!$ and $P_2 \stackrel{\text{def}}{=} \tau$. Then $P_1 \lesssim P_2$ but $P_1 \not\precsim_{\mathcal{C}} P_2$.*

Proof. Clearly $P_1 \not\precsim_{\mathcal{C}} P_2$ since $P_1 \xrightarrow{a!} a?$ which cannot be matched by P_2.

We will show that $\mathcal{R} \stackrel{\text{def}}{=} \{(P_1 \parallel R, P_2 \parallel R) \mid R \text{ a process}\} \cup \Delta$ (where Δ is the identity relation) is closed under reduction and stable under parallel composition. Suppose that $P_1 \parallel R \rightarrow Q$. Then:

(i) $R \rightarrow R'$ and $Q = P_1 \parallel R'$. Then also $P_2 \parallel R \rightarrow P_2 \parallel R'$ and $Q \mathrel{\mathcal{R}} (P_2 \parallel R')$ by construction;

(ii) $P_1 \rightarrow 0$ and $Q = R$. But also $\tau \rightarrow 0$ so $\tau \parallel R \rightarrow R$;

(iii) $R = a! \parallel R'$ and $P_1 \parallel R \rightarrow a! \parallel R' = R$. But also $\tau \parallel R \rightarrow R$.

(iv) $R = a? \parallel R'$ which is similar to the previous case. □

We also have $P_2 \lesssim P_1$ and, moreover, $P_1 \simeq P_2$; the two processes are reduction congruent. In fact, already $a! \lesssim P_2$ and $a? \lesssim P_2$, although, in each of these cases the other direction does not hold since P_2 can reduce. Of course, neither $a? \not\lesssim a!$ nor $a! \not\lesssim a?$.

We have shown that there is a mismatch between our logbook (the LTS) and the actual power of experiments. This can be expressed succinctly by noting that the converse of Lemma 3.3 does not hold and, more than this, no characterisation of the labels by contexts is possible. This is implied by the following result.

Theorem 3.7 *If, for an* LTS $\mathcal{X}$, *there exist processes* χ_α *such that*

$$P \xrightarrow{\alpha}_{\mathcal{X}} P' \quad \textit{iff} \quad P \parallel \chi_\alpha \rightarrow P'$$

then $\lesssim \subseteq \precsim_{\mathcal{X}}$ *and* $\simeq \subseteq \sim_{\mathcal{X}}$.

Proof. We will show that $\lesssim$ is a $(\mathcal{X})$-simulation. Suppose that $P \lesssim Q$ and $P \xrightarrow{\alpha} P'$. Then $P \parallel \chi_\alpha \rightarrow P'$ and since $\lesssim$ is stable under $\parallel$ and closed under reduction we have that there exists Q' such that $P' \lesssim Q'$ and $Q \parallel \chi_\alpha \rightarrow Q'$. But, by assumption, the final part implies that $Q \xrightarrow{\alpha} Q'$. The same reasoning shows that $\simeq$ is a bisimulation. □

Corollary 3.8 *For* $\alpha \in \{a?, a!\}$, *there do not exist* χ_α *such that* $P \xrightarrow{\alpha} P'$ *iff* $P \parallel \chi_\alpha \rightarrow P'$.

Proof. If such a set existed then by Theorem 3.7 we would have $\lesssim \subseteq \precsim_{\mathcal{C}}$ which we know is not true by the conclusion of Lemma 3.6. □

What has gone wrong? A moment's reflection about the processes of Lemma 3.6 confirms that, for instance, our tentative idea that $P \xrightarrow{a!} P'$ tests for the presence of an output (and dually, an input) on a in P is unimplementable. Indeed, the experimenter which provides an $a?$ process and observes an interaction can conclude that either P has an output on a or that *a reduction was already possible in* P. The following rules[2] take into account the possibility of the latter:

$$\frac{P \xrightarrow{\tau} P'}{P \xrightarrow{a?} P' \| a!}\ (\textsc{InHT}) \qquad \frac{P \xrightarrow{\tau} P'}{P \xrightarrow{a!} P' \| a?}\ (\textsc{OutHT})$$

[2] These rules are named after Honda and Tokoro's rule for asynchronous π-calculus, although here they are modified for strong equivalences.

The extra component in the results is just the process that the experimenter provided. Since the reduction was already present in P, this process was not consumed. Letting $\Psi_f \stackrel{\text{def}}{=} \{(\textsc{InHT}), (\textsc{OutHT})\}$, we will consider the LTS $\mathcal{HT}_f \stackrel{\text{def}}{=} \Psi_f \mathcal{C}_f$. In other words, $\mathcal{HT}_f$ is the LTS with derivations that can be split into two (possibly empty) parts - the first which features rules from Φ_f and the second with rules from Ψ_f. We shall sometimes refer to those transitions which are in $\mathcal{HT}$ but not in $\mathcal{C}$ as Honda-Tokoro transitions.

The following lemma shows the relationship between the $a?$- and $a!$-labelled transitions in the two LTSs.

Lemma 3.9 *If $\alpha \in \{a!, a?\}$: $P \xrightarrow{\alpha}_{\mathcal{HT}} P' \Leftrightarrow P \xrightarrow{\alpha}_{\mathcal{C}} P' \vee (P \xrightarrow{\tau} P'' \wedge P' = P'' \parallel \chi_\alpha)$.* □

The τ-labelled transitions are unchanged. We use these observations to show that simulation (bisimulation) on $\mathcal{HT}$ remains sound for reduction precongruence (congruence).

Lemma 3.10 (Soundness of $\mathcal{HT}_f$) $\precsim_{\mathcal{HT}} \subseteq \lesssim$ *and* $\sim_{\mathcal{HT}} \subseteq \simeq$.

Proof. It suffices to show that $\precsim$ is stable under $\parallel$. This is done by showing that $\{(P \parallel R, Q \parallel R) \mid P \precsim Q\}$ is a simulation. One of the two symmetric interesting cases is again $P \parallel R \xrightarrow{\tau} P' \parallel R'$ for $P \xrightarrow{a!}_{\mathcal{C}} P'$ and $R \xrightarrow{a?}_{\mathcal{C}} R'$. Note that $R = R' \parallel a?$. We have, by assumption, that $\exists Q'. Q \xrightarrow{a!} Q'$ such that $P' \precsim Q'$. Now, using the conclusion of Lemma 3.9, either $Q \xrightarrow{a!}_{\mathcal{C}} Q'$, in which case also $Q \parallel R \xrightarrow{\tau} Q' \parallel R'$, or $Q \xrightarrow{\tau} Q''$ such that $Q' = Q'' \parallel a?$. But then $Q \parallel R \xrightarrow{\tau} Q'' \parallel R = Q'' \parallel a? \parallel R' = Q' \parallel R'$. □

The rules of Ψ_f are designed so that the 'if' direction of the following statement holds, provided that the 'only if' direction holds.

Lemma 3.11 *Let χ_α be defined as in Lemma 3.3. Then*

$$P \xrightarrow{\alpha}_{\mathcal{HT}} P' \quad \textit{iff} \quad P \parallel \chi_\alpha \rightarrow P'$$

Proof. Lemma 3.4 takes care of τ. ($\Rightarrow$) is implied by Lemma 3.3 and the fact that if $P \rightarrow P'$ then $P \parallel \chi_\alpha \rightarrow P' \parallel \chi_\alpha$. For ($\Leftarrow$) if $P \parallel \chi_\alpha \rightarrow P'$ then either the χ_α is consumed in which case $P \xrightarrow{\alpha}_{\mathcal{C}} P'$ or it is not, in which case $P \xrightarrow{\alpha}_{\mathcal{HT}} P'$. □

The above is enough to derive completeness.

Corollary 3.12 (Completeness of $\mathcal{HT}_f$) $\lesssim \subseteq \precsim_{\mathcal{HT}}$ *and* $\simeq \subseteq \sim_{\mathcal{HT}}$

Proof. Consequence of the conclusions of Lemma 3.11 and Theorem 3.7. □

4 Asynchrony

Here we will consider a language $\mathcal{L}_a$ where, an $a?$ capability can guard another process.

$$P \;::=\; 0 \mid a! \mid a?P \mid P \parallel Q \mid \tau P$$

The reduction relation is the smallest transition system that contains $a! \parallel a?P \rightarrow P$ as well as $\tau P \rightarrow P$ and is closed by parallel composition.

As before, we have the experiment for the τ prefix where nothing needs to be provided in order to observe a heating:

$$\frac{}{\tau P \xrightarrow{\tau} P}\ (\textsc{Tau}) \qquad \frac{P \xrightarrow{\tau} P'}{P\|Q \xrightarrow{\tau} P'\|Q}\ (\textsc{Tau}\|)$$

Also, the observer can experiment with a process by providing an output:

$$\frac{}{a?P \xrightarrow{a?} P}\ (\textsc{In}) \qquad \frac{P \xrightarrow{a?} P'}{P\|Q \xrightarrow{a?} P'\|Q}\ (\textsc{In}\|)$$

The experiment for outputs is more involved because our language now allows more powerful tests: the observer introduces an input on a followed by some other process R of the observer's choosing ($a?R$) and observes a rise in temperature consistent with one interaction. We will denote[3] this kind of experiment $a!{\downarrow}\, R$. The inductive presentation is the following:

$$\frac{}{a! \xrightarrow{a!\downarrow R} R}\ (\textsc{Out}) \qquad \frac{P \xrightarrow{a!\downarrow R} P'}{P\|Q \xrightarrow{a!\downarrow R} P'\|Q}\ (\textsc{Out}\|)$$

The final rule is needed to characterise those internal reductions which come from two interacting parallel components:

$$\frac{P \xrightarrow{a?} P' \quad Q \xrightarrow{a!\downarrow 0} Q'}{P\|Q \xrightarrow{\tau} P'\|Q'}\ (\textsc{Comm})$$

Let $\Phi_a \overset{\text{def}}{=} \{(\textsc{Tau}), (\textsc{Tau}\|), (\textsc{In}), (\textsc{In}\|), (\textsc{Out}), (\textsc{Out}\|), (\textsc{Comm})\}$ and $\mathcal{C}_a = \Phi_a(\varnothing)$. The following is the counterpart of Lemma 3.3 for our current setting.

Lemma 4.1 *Let $\chi_{a!\downarrow R} = a?R$, $\chi_{a?} = a!$, $\chi_\tau = 0$. Then $P \xrightarrow{\alpha} P'$ implies $P \parallel \chi_\alpha \to P'$.* □

It is also easy to check that the conclusion of Lemma 3.4 holds. We easily obtain soundness; the proof of the following is essentially the same as the proof of Lemma 3.5.

Lemma 4.2 (Soundness of $\mathcal{C}_a$) $\precsim_{\mathcal{C}} \subseteq \lesssim$ *and* $\sim_{\mathcal{C}} \subseteq \simeq$. □

Analogously to the fully synchronous case, simulation on $\mathcal{C}_a$ is too strong: there exist processes which are distinguished by similarity but which *are not* distinguished by observational precongruence.

Lemma 4.3 *Let $P_1 \overset{\text{def}}{=} a?a!$ and $P_2 \overset{\text{def}}{=} \tau$. Then $P_1 \lesssim P_2$ but $P_1 \not\precsim_{\mathcal{C}} P_2$.*

Proof. Again, it is easy to see that $P_1 \not\precsim_{\mathcal{C}} P_2$ as $P_1 \xrightarrow{a?} a!$ which cannot be matched by P_2.

[3] The motivation for this notation is our work on deriving structural LTSs from reduction rules [12, 13]. Roughly, the '$\downarrow$' in the label separates the information provided by the process (here an output capability) from the data provided by the environment (here R).

On the other hand $\mathcal{R} \stackrel{\text{def}}{=} \{(P_1 \parallel R, P_2 \parallel R) \mid R \text{ a process}\} \cup \Delta$ is closed under reduction and stable under parallel composition: Suppose that $P_1 \parallel R \rightarrow Q$. Then:

(i) the reduction comes from R, so $Q = P_1 \parallel R'$, and $P_2 \parallel R \rightarrow P_2 \parallel R'$;

(ii) $R = a! \parallel R'$ and $P_1 \parallel R \rightarrow a! \parallel R' = R$. But also $\tau \parallel R \rightarrow R$.

□

To obtain completeness, we can again generate a new LTS $\mathcal{HT}_a$ from $\mathcal{C}_a$ by applying rules:

$$\frac{P \xrightarrow{\tau} P'}{P \xrightarrow{a?} P' \| a!}\ (\text{InHT}) \qquad \frac{P \xrightarrow{\tau} P'}{P \xrightarrow{a!\downarrow R} P' \| a?R}\ (\text{OutHT})$$

Corollary 4.4 (Completeness of $\mathcal{HT}_a$) *$\lesssim \subseteq \precsim_{\mathcal{HT}}$ and $\simeq \subseteq \sim_{\mathcal{HT}}$.*

Proof. The presence of the Honda-Tokoro rules allows us to establish the counterpart to Lemma 3.11 and completeness follows from Theorem 3.7. □

The fact that we are applying *both* (InHT) and (OutHT) should come as a surprise. In fact, in the fully asynchronous case, it was intuitively clear that both the labels should be unobservable. Here, while the $a?$ transition should be unobservable, our intuition tells us that $a!$ should be observable. The crucial observation is that while (InHT) really does make the inputs unobservable, the (OutHT) does not make outputs unobservable, it only accounts for the fact that the experiment for $a!$ can fail. In fact, in the fully asynchronous setting we had $a! \precsim_{\mathcal{HT}} \tau$ and this does not hold here:

Example 1 $a! \not\precsim_{\mathcal{HT}} \tau$.

Proof. $a! \xrightarrow{a!\downarrow\tau} \tau$. The τ process must match this with the Honda-Tokoro transition $\tau \xrightarrow{a!\downarrow\tau} a?\tau$. But clearly $\tau \not\precsim_{\mathcal{HT}} a?\tau$, since the first process can do a τ labelled transition. □

Indeed, $\mathcal{HT}$ remains sound.

Lemma 4.5 (Soundness of $\mathcal{HT}_a$) *$\precsim_{\mathcal{HT}} \subseteq \lesssim$ and $\sim_{\mathcal{HT}} \subseteq \simeq$.*

Proof. We need to show that $\precsim$ is stable under $\|$. This is done by showing that $\mathcal{R} = \{(P \parallel R, Q \parallel R) | P \precsim_{\mathcal{HT}} Q\}$ is a simulation.

The interesting case is $P \parallel R \xrightarrow{\tau} P' \parallel R'$ for $P \xrightarrow{a!\downarrow 0}_{\mathcal{C}} P'$ and $R \xrightarrow{a?}_{\mathcal{C}} R'$. Suppose that Q does not have an output on a available and will be forced to match the $a!\downarrow 0$ transition with a Honda-Tokoro transition.

We know that, for some S and R'', $R = R'' \parallel a?S$ and $R' = R'' \parallel S$. The key observation is that $P \xrightarrow{a!\downarrow S}_{\mathcal{C}} P' \parallel S$. Then Q will have to match this with a Honda-Tokoro transition, hence $\exists Q'$ such that $Q \xrightarrow{\tau} Q'$ and $Q \xrightarrow{a!\downarrow S}_{\mathcal{HT}} Q' \parallel a?S$ with $P' \parallel S \precsim Q' \parallel a?S$ (†). But then $Q \parallel R \xrightarrow{\tau} Q' \parallel R = Q' \parallel a?S \parallel R''$. But $P' \parallel R' = P' \parallel S \parallel R''$, hence by (†), we remain in $\mathcal{R}$.

Similar reasoning goes through for the case of bisimilarity and reduction congruence. □

In Section 6 we will show that, when reasoning about reduction congruence, the rule (OutHT) is not actually necessary.

5 Synchrony

We can recycle our results for the synchronous language $\mathcal{L}_s$, where both inputs and outputs guard other processes:

$$P \ ::= \ 0 \mid a!P \mid a?P \mid P \parallel Q \mid \tau P$$

The reduction relation $\to$ is the smallest relation which, for any P, Q contains $a!P \parallel a?Q \to P \parallel Q$ as well as $\tau P \to P$ and is closed under parallel composition. In this case, our $\mathcal{C}_s$ LTS is generated by:

$$\frac{}{a?P \xrightarrow{a?\downarrow R} P\|R}\ (\textsc{In}) \qquad \frac{P \xrightarrow{a?\downarrow R} P'}{P\|Q \xrightarrow{a?\downarrow R} P'\|Q}\ (\textsc{In}\|) \qquad \frac{}{a!P \xrightarrow{a!\downarrow R} P\|R}\ (\textsc{Out}) \qquad \frac{P \xrightarrow{a!\downarrow R} P'}{P\|Q \xrightarrow{a!\downarrow R} P'\|Q}\ (\textsc{Out}\|)$$

$$\frac{P \xrightarrow{a?\downarrow 0} P' \quad Q \xrightarrow{a!\downarrow 0} Q'}{P\|Q \xrightarrow{\tau} P'\|Q'}\ (\textsc{Comm}) \qquad \frac{}{\tau P \xrightarrow{\tau} P}\ (\textsc{Tau}) \qquad \frac{P \xrightarrow{\tau} P'}{P\|Q \xrightarrow{\tau} P'\|Q}\ (\textsc{Tau}\|)$$

The set Ψ_s of Honda-Tokoro rules is the set containing the two rules:

$$\frac{P \xrightarrow{\tau} P'}{P \xrightarrow{a?\downarrow R} P'\|a!R}\ (\textsc{InHT}) \qquad \frac{P \xrightarrow{\tau} P'}{P \xrightarrow{a!\downarrow R} P'\|a?R}\ (\textsc{OutHT})$$

and we automatically obtain a sound and complete LTS (for both reduction precongruence and reduction congruence) $\mathcal{HT}_s = \Psi_s(\mathcal{C}_s)$.

Just as in the asynchronous case, when reasoning about reduction congruence we can be more efficient. In fact, the results of the proceeding section imply that the transitions generated by the rules in Ψ_s are not actually necessary for completeness and, in fact, bisimilarity on $\mathcal{C}_s$ is already sound and complete for reduction congruence.

6 Refining rules

We have shown that closing wrt (InHT) and (OutHT) is an "automatic" way of obtaining an LTS on which similarity characterises reduction precongruence and indeed bisimilarity characterises reduction congruence. It is clear that this procedure is essentially the same in our three settings and is sufficient for completeness. What we have not addressed in closing the LTSs with these rules is whether it was *necessary* to do so. Indeed, for the synchronous language one would expect that both input and output actions are observable and that there should be no need for additional $\mathcal{HT}$ rules. This is indeed the case when considering bisimilarity and reduction congruence and in this section we shall demonstrate how we can safely remove some of the $\mathcal{HT}$ rules in certain circumstances.

To start, consider the language $\mathcal{L}_a$ of asynchronous communication (*cf.* Section 4). Interestingly, although the exclusion of the (OutHT) rule breaks completeness of the LTS for similarity, this is not the case for bisimilarity. To show this requires more work but the reward is a refined LTS in the sense that bisimulations in specific cases can be made smaller, and therefore, reasoning about reduction congruence is easier.

The key to showing the redundancy of (OutHT) for the asynchronous language lies in the fact that strong output actions are preserved by reduction congruence as given by the following result, which is similar in spirit to Theorem 2 of [9]. However, the notion of observational equivalence is different (ours is "dynamic" in the sense of [10]).

Theorem 6.1 *Suppose that $a! \parallel Q \simeq R$, then $R = a! \parallel R'$ for some R'.*

Proof. Omitted. □

Indeed, let $\sim_{in\mathcal{HT}}$ denote bisimilarity over the LTS given by extending with only rule (inHT). The characterisation of reduction congruence does not break:

Theorem 6.2 $\sim_{in\mathcal{HT}} = \simeq$.

Proof. It is easy to check that soundness ($\sim_{in\mathcal{HT}} \subseteq \simeq$) still holds. For the reverse inclusion, completeness, we establish that $\simeq$ is in fact a bisimulation relation. Suppose then that $P \simeq Q$ and that $P \xrightarrow{\alpha} P_0$. The interesting case is $\alpha = a!R$: choose a name c which is fresh for both P and Q and let $\chi \stackrel{\text{def}}{=} c! \parallel a?c?$; P must be able to engage in output on a so

$$P \parallel \chi \to (P'' \parallel c! \parallel c?) \to P'$$

for some P', P'' where P_0 is $P' \parallel R$. We see that because $P \simeq Q$, we must also have some Q', Q'' and

$$Q \parallel \chi \to Q'' \to Q'$$

such that $(P'' \parallel c! \parallel c?) \simeq Q''$ and $P' \simeq Q'$. The freshness of c and Theorem 6.1 tells us that c is not in P' and thus Q' cannot be of the form $c! \parallel Q'''$ for any Q'''. But this means that the $c!$ in χ must have been consumed. The only possibility for this is if $Q'' = Q' \parallel c! \parallel c?$ and this could have only arisen if $Q = a! \parallel Q'$. From this we can see that $Q \xrightarrow{a!R} Q' \parallel R$ and, by congruence of $\simeq$ we have $P' \parallel R \simeq Q' \parallel R$ as required. □

The key property used in the above proof is the preservation of strong output actions given by Theorem 6.1. This property is a useful one for characterising the observability of particular actions and has in fact been exploited in the literature in the form of barbed equivalences [9,6]. In fact, as a corollary of Theorem 6.1 we can easily check that reduction barbed congruence in the finite asynchronous calculus with output barbs coincides with our reduction congruence.

For the synchronous language $\mathcal{L}_s$ it is in fact possible to prove that bisimilarity over $\Phi_s(\varnothing)$ without the addition of any Honda Tokoro rules is already complete for reduction congruence. The proof of this relies on establishing results that correspond to Theorems 6.1 and 6.2 but for both input and output actions separately. The point is that it is sound to blindly add the $\mathcal{HT}$ rules to $\Phi_s(\varnothing)$ in order to automatically obtain completeness. We do not need to know whether inputs and outputs are observable or not.

The $\mathcal{HT}$ rules certainly increase the size of the LTS and introduce some undesirable properties such as infinite branching. Indeed, when $\mathcal{HT}$ rules can be avoided, one often obtains a more useful LTS in the sense that bisimulations are easier to construct. We finish this section with a trivial but illustrative example to show the impact of the (removal of the) rule (outHT) on reasoning with bisimilarity. Consider the processes of $\mathcal{L}_a$

$$P \stackrel{\text{def}}{=} a! \parallel a?0 \qquad \text{and} \qquad Q \stackrel{\text{def}}{=} \tau 0$$

and suppose that we wish to demonstrate that $P \not\sim Q$. Theorem 6.2 would immediately distinguish these processes as there is no $\sim_{in\mathcal{HT}}$ which relates these due to the presence of an output action on a in P which is clearly absent from Q. Compare this the use of the $\sim_{\mathcal{HT}}$ relation: consider how Q could match

$$P \xrightarrow{a!\downarrow c!} \xrightarrow{c!\downarrow 0} a?$$

for some c. The rule (outHT) would allow $Q \xrightarrow{a!\downarrow c!} a?c!$ as an attempt to match P's initial move but there can be no subsequent matching transition from $a?c!$.

The interesting point in the above example is that there is a non-trivial use of the continuation process in the experiment $a!\!\downarrow c!$. Indeed, these continuation processes are crucial to the soundness of using $\mathcal{HT}$ rules. Note that in the more traditional approach to LTS semantics that one finds in, say, [8], there is no room for specifying the continuation process in an experiment — effectively it is always just the nil process. For the synchronous language this turns out to be sufficient as no $\mathcal{HT}$ rules are necessary, however, what is significant that, if the continuation processes are restricted to be the nil process, it is actually *unsound* to add them. The previous example suffices to demonstrate this because P and Q would in fact become bisimilar if we restricted to $a!\!\downarrow 0$ labels and admitted rule (outHT).

7 Conclusions and related work

Sewell's paper [16] about the derivation of LTS has stimulated considerable interest (*e.g.* [7,14,15,4]) in the relationship between labelled transition systems and underlying reduction semantics. Our simple "fully asynchronous" language (*cf.* Section 3) was considered already by Sewell but observability was not taken into account in his derived LTS. Bonchi [3] has also considered this example: reduction congruence agrees with the closely related concept of saturated semantics. In this paper, we did not consider the derivation process as such but it is usually the case that such derivations (*cf.* [12,13]) yield LTSs which are sound but not complete, here we have illustrated a method by which one "completes" them. This fits in with our general research programme that aims at developing techniques (such as derivation of LTSs) which are applicable across several languages for concurrency.

Throughout the present paper we have considered finite languages. A natural question is whether the techniques that we have studied extend to languages with infinite behaviour. To answer it, we must first divulge our reasons for considering finiteness: by assuming it we were able to use very simple notions of observational preorders/equivalences (*cf.* Definitions 3.1 and 3.2). In particular, we were able to avoid talking about *barbs* which specify "moral observability", and such notions become less clear for more involved languages [9]. In particular, extending the ideas of [9] may lead to a more clear understanding of the general infinite case.

Acknowledgement.

We thank the anonymous referees for their helpful remarks, which have helped us to improve the presentation of the paper.

References

[1] R. Amadio, I. Castellani, and D. Sangiorgi. On bisimulations for the asynchronous pi-calculus. *Theor. Comput. Sci.*, 195(2):291–324, 1998.

[2] G. Berry and G. Boudol. The chemical abstract machine. *Theor. Comput. Sci.*, 96:217–248, 1992.

[3] F. Bonchi. *Abstract Semantics by Observable Contexts.* PhD thesis, Università degli studi di Pisa, 2008.

[4] F. Bonchi, B. König, and U. Montanari. Saturated semantics for reactive systems. In *LiCS '06*, pages 69–80. IEEE Press, 2006.

[5] K. Honda and M. Tokoro. An object calculus for asynchronous communication. In *Proc. ECOOP*, volume 512 of *LNCS*, pages 133–147. Springer, 1991.

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

[7] J. Leifer and R. Milner. Deriving bisimulation congruences for reactive systems. In *Proc. Concur*, volume 1877 of *LNCS*, pages 243–258. Springer, 2000.

[8] R. Milner. *A Calculus of Communicating Systems*, volume 92 of *LNCS*. Springer, 1980.

[9] R. Milner and D. Sangiorgi. Barbed bisimulation. In *ICALP '92*, volume 623 of *LNCS*, pages 685–695. Springer, 1992.

[10] U. Montanari and V. Sassone. Dynamic congruence vs. progressing bisimulation for CCS. *Fundamenta Informaticae*, XVI:171–199, 1992.

[11] G. Plotkin. A structural approach to operational semantics. Technical Report FN-19, DAIMI, Computer Science Department, Aarhus University, 1981.

[12] J. Rathke and P. Sobociński. Deconstructing behavioural theories of mobility. In *TCS '08*. Springer Science and Business Media, 2008. To appear.

[13] J. Rathke and P. Sobociński. Deriving structural labelled transitions for mobile ambients. In *Concur '08*, LNCS. Springer, 2008. To appear.

[14] V. Sassone and P. Sobociński. Deriving bisimulation congruences using 2-categories. *Nordic Journal of Computing*, 10(2):163–183, 2003.

[15] V. Sassone and P. Sobociński. Locating reaction with 2-categories. *Theoretical Computer Science*, 333(1-2):297–327, 2005.

[16] P. Sewell. From rewrite rules to bisimulation congruences. *Theor. Comput. Sci.*, 274(1-2):183–230, 2002. Extended version of Concur '98 conference paper.