



HAL
open science

On convergence-sensitive bisimulation and the embedding of CCS in timed CCS

Roberto Amadio

► **To cite this version:**

Roberto Amadio. On convergence-sensitive bisimulation and the embedding of CCS in timed CCS. 2008. hal-00285337v1

HAL Id: hal-00285337

<https://hal.science/hal-00285337v1>

Preprint submitted on 5 Jun 2008 (v1), last revised 18 Sep 2008 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On convergence-sensitive bisimulation and the embedding of CCS in timed CCS

Roberto M. Amadio*
 Université Paris Diderot†

5th June 2008

Abstract

We propose a notion of convergence-sensitive bisimulation as a suitable semantic framework for a fully abstract embedding of untimed processes into timed ones.

1 Introduction

Our motivation for this work is two fold. On one hand, we wish to understand in a simple and general setting the notion of bisimulation we developed for a timed/synchronous π -calculus in [2]. On the other hand, we look for an intuitive semantic framework that spans both untimed/asynchronous and timed/synchronous models.

We assume the reader is familiar with CCS [10]. Timed CCS (TCCS) is a ‘timed’ version of CCS whose basic principle is that *time passes exactly when no internal computation is possible*. This notion of ‘time’ is inspired by early work on the ESTEREL synchronous language [3], and it has been formalised in various dialects of CCS [14, 12, 6]. Here we shall follow the formalisation in [6].

As in CCS, one models the internal computation with an action τ while the passage of (discrete) time is represented by an action `tick` that implicitly synchronizes all the processes and moves the computation to the next instant.¹

In this framework, the basic principle we mentioned is formalised as follows:

$$P \xrightarrow{\text{tick}} \cdot \text{ iff } P \not\rightarrow \cdot$$

where we write $P \xrightarrow{\mu} \cdot$ if P can perform an action μ . TCCS is designed so that if P is a process built with the usual CCS operators and P cannot perform τ actions then $P \xrightarrow{\text{tick}} P$.

*Work partially supported by ANR-06-SETI-010-02.

†PPS, UMR-7126.

¹There seems to be no standard terminology for this action. It is called ϵ in [14], χ in [12], σ in [6], and sometimes ‘next’ in ‘synchronous’ languages à la ESTEREL [2].

$\frac{}{a.P \xrightarrow{a} P}$	$\frac{P \xrightarrow{a} P' \quad Q \xrightarrow{\bar{a}} Q'}{(P \mid Q) \xrightarrow{\tau} (P' \mid Q')}$
$\frac{P \xrightarrow{\alpha} P'}{(P \mid Q) \xrightarrow{\alpha} (P' \mid Q)}$	$\frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'}$
$\frac{A(\mathbf{a}) = P}{A(\mathbf{b}) \xrightarrow{\tau} [\mathbf{b}/\mathbf{a}]P}$	$\frac{P \xrightarrow{\alpha} P'}{P \triangleright Q \xrightarrow{\alpha} P'}$
$\frac{}{0 \xrightarrow{\text{tick}} 0}$	$\frac{}{a.P \xrightarrow{\text{tick}} a.P}$
$\frac{P \xrightarrow{\bar{\tau}} \cdot}{P \triangleright Q \xrightarrow{\text{tick}} Q}$	$\frac{(P_1 \mid P_2) \xrightarrow{\bar{\tau}} \cdot \quad P_i \xrightarrow{\text{tick}} Q_i \quad i = 1, 2}{(P_1 \mid P_2) \xrightarrow{\text{tick}} (Q_1 \mid Q_2)}$
$\frac{P_i \xrightarrow{\text{tick}} Q_i \quad i = 1, 2}{P_1 + P_2 \xrightarrow{\text{tick}} Q_1 + Q_2}$	$\frac{P \xrightarrow{\mu} Q \quad a, \bar{a} \neq \mu}{\nu a P \xrightarrow{\mu} \nu a Q}$

Table 1: Labelled transition system

In other terms, CCS processes are *time insensitive*. To compensate for this property, one introduces a new binary operator $P \triangleright Q$, called *else_next*, that tries to run P in the current instant and, if it fails, runs Q in the following instant.

We denote with α, β, \dots the usual CCS actions which are composed of either an internal action τ or of a communication action a (each communication action a having a co-action \bar{a}). We denote with μ, μ', \dots either an action α or the distinct action tick .

The TCCS processes P, Q, \dots are specified by the following grammar

$$P ::= 0 \mid a.P \mid P + P \mid P \mid P \mid \nu a P \mid A(\mathbf{a}) \mid P \triangleright P .$$

We denote with $fn(P)$ the names free in P . We adopt the usual convention that for each thread identifier A there is a unique defining equation $A(\mathbf{b}) = P$ where the parameters \mathbf{b} include the names in $fn(P)$. The related labelled transition system is specified in table 1.

Say that a process is a CCS process if it does not contain the *else_next* operator. The reader can easily verify that:

- (1) $P \xrightarrow{\text{tick}} \cdot$ if and only if $P \xrightarrow{\bar{\tau}} \cdot$.
- (2) If $P \xrightarrow{\text{tick}} Q_i$ for $i = 1, 2$ then $Q_1 = Q_2$. One says that the passage of time is *deterministic*.
- (3) If P is a CCS process and $P \xrightarrow{\text{tick}} Q$ then $P = Q$. Hence CCS processes are closed under labelled transitions.

The labelled transition system in table 1 relies on negative conditions of the shape $P \not\downarrow$. These conditions can be replaced by a condition $\exists L. P \downarrow L$, where L is a finite set of communication actions. This predicate can be defined as follows:

$$\begin{array}{c} \frac{}{0 \downarrow \emptyset} \quad \frac{}{a.P \downarrow \{a\}} \quad \frac{P_i \downarrow L_i, \quad i = 1, 2}{(P_1 + P_2) \downarrow L_1 \cup L_2} \\ \\ \frac{P \downarrow L}{P \triangleright Q \downarrow L} \quad \frac{P \downarrow L}{(\nu a. P) \downarrow L \setminus \{a, \bar{a}\}} \quad \frac{P_i \downarrow L_i, \quad i = 1, 2 \quad L_1 \cap \bar{L}_2 = \emptyset}{(P_1 | P_2) \downarrow L_1 \cup L_2} \end{array}$$

It will be convenient to write $\tau.P$ for $\nu a. (a.P | \bar{a}.0)$ where $a \notin \text{fn}(P)$, $\text{tick}.P$ for $0 \triangleright P$, and Ω for the diverging process $\tau.\tau.\dots$

1.1 Signals and a deterministic fragment

As already mentioned, the TCCS model has been inspired by the notion of time available in the ESTEREL model [4] and its relatives such as SL [5]. These models rely on *signals* as the basic communication mechanism. Unlike a channel, a signal persists within the instant and disappears at the end of instant. It turns out that a signal can be defined recursively in TCCS as:

$$\text{emit}(a) = \bar{a}.\text{emit}(a) \triangleright 0$$

The ‘present’ statement of SL that either reads a signal and continues the computation in the current instant or reacts to the absence of the signal in the following instant can be coded as follows:

$$\text{present } a \text{ do } P \text{ else } Q = a.P \triangleright Q$$

Modulo these encodings, the resulting fragment of TCCS is specified as follows:

$$P ::= 0 \mid \text{emit}(a) \mid \text{present } a \text{ do } P \text{ else } P \mid (P \mid P) \mid \nu a. P \mid A(\mathbf{a}) .$$

Notice that, unlike in (T)CCS, communication actions have an input or output polarity. The most important property of this fragment is that its processes are *deterministic* [5, 1].

1.2 The usual labelled bisimulation

As usual, one can define a notion of *weak transition* as follows:

$$\stackrel{\mu}{\Rightarrow} = \begin{cases} (\overset{\tau}{\rightarrow})^* & \text{if } \mu = \tau \\ (\overset{\tau}{\rightarrow})^* \circ \overset{\mu}{\rightarrow} \circ (\overset{\tau}{\rightarrow})^* & \text{otherwise} \end{cases}$$

where the notation X^* stands for the reflexive and transitive closure of a binary relation X . When focusing just on internal reduction, we shall write \rightarrow for $\overset{\tau}{\rightarrow}$ and \Rightarrow for $\stackrel{\tau}{\Rightarrow}$.

With respect to this notion of weak transition, one can define the usual notion of bisimulation as the largest symmetric relation \mathcal{R} such that if $(P, Q) \in \mathcal{R}$ and $P \stackrel{\mu}{\Rightarrow} P'$ then for some $Q', Q \stackrel{\mu}{\Rightarrow} Q'$ and $(P', Q') \in \mathcal{R}$. We denote with \approx^u the largest labelled bisimulation (u for *usual*). When looking at CCS processes, one may focus on CCS actions (thus excluding the tick action). We denote with \approx_{ccs}^u the resulting labelled bisimulation.

Remark 1 *The `else_next` operator suffers from the same compositionality problems as the `sum` operator. For instance, $0 \approx^u \tau.0$ but $0 \triangleright Q = \text{tick}.Q$ while $\tau.0 \triangleright Q \approx^u 0$. As for the `sum` operator, one may remark that in practice we are interested in a guarded form of the `else_next` operator. Namely, the `else_next` operator is only introduced as an alternative to a communication action (the `present` operator discussed in section 1.1 is such an example). One can check that in this form, the `else_next` operator preserves bisimulation equivalence.*

1.3 CCS vs. TCCS

As we already noticed, TCCS has been designed so that CCS can be regarded as a transition closed subset of TCCS. A natural question is whether two CCS processes which are equivalent with respect to an untimed environment are still equivalent in a timed one. For instance, Milner [9] discusses a similar question when comparing CCS to SCCS.²

In the context of TCCS and of a testing semantics, the question has been answered negatively by Hennessy and Reagan [6]. For instance, they notice that the processes $P = a.(b + c.b) + a.(d + c.d)$ and $Q = a.(b + c.d) + a.(d + c.b)$ are ‘untimed’ testing equivalent but ‘timed’ testing inequivalent. The relevant test is the one that checks that if an action b cannot follow an action a in the current instant then an action b will happen in the following instant just after an action c (process P will not pass this test while process Q does). This remark motivated the authors to develop a notion of ‘timed’ testing semantics.

What is the situation with the usual labelled bisimulation semantics recalled in section 1.2? Things are fine for *reactive* processes which are defined as follows.

Definition 2 *A process P is reactive if whenever $P \xrightarrow{\mu_1} \dots \xrightarrow{\mu_n} Q$, for $n \geq 0$, we have the property that all sequences of τ reductions starting from Q terminate.*

Proposition 3 *Suppose P, Q are CCS reactive processes. Then $P \approx^u Q$ if and only if $P \approx_{ccs}^u Q$.*

PROOF. Clearly, \approx^u is a CCS bisimulation, hence $P \approx^u Q$ implies $P \approx_{ccs}^u Q$. To show the converse, we prove that \approx_{ccs}^u is a timed bisimulation. So suppose $P \approx_{ccs}^u Q$ and $P \xrightarrow{\text{tick}} P'$. This means $P \xrightarrow{\tau} P_1 \xrightarrow{\text{tick}} P_1 \xrightarrow{\tau} P'$. Then for some Q_1 , $Q \xrightarrow{\tau} Q_1$ and $P_1 \approx_{ccs}^u Q_1$. Further, because Q_1 is reactive there is a Q_2 such that $Q_1 \xrightarrow{\tau} Q_2$ and $Q_2 \downarrow$. By definition of bisimulation and the fact that $P_1 \downarrow$, we have that $P_1 \approx_{ccs}^u Q_2$. So for some Q' , $Q_2 \xrightarrow{\tau} Q'$ and $P' \approx_{ccs}^u Q'$. Thus we have shown that there is a Q' such that $Q \xrightarrow{\text{tick}} Q'$ and $P' \approx_{ccs}^u Q'$. \square

Proposition 3 fails when we look at *non-reactive* processes. For instance, 0 and Ω are regarded as untimed equivalent but they are obviously timed inequivalent since the second process does not allow time to pass. On the other hand, one can argue that in a

²The notion of instant in SCCS is quite different from the one considered in TCCS/ESTEREL. In the former one declares explicitly what each thread does at each instant while in the latter the duration of an instant is the result of an arbitrarily complex interaction among the different threads.

timed framework no observation is possible unless the instant terminates and therefore, for instance, $(a \mid \Omega)$ should be regarded as equivalent to Ω , while they are usually regarded as (untimed) inequivalent.

These examples suggest that if we want to extend proposition 3 to non-reactive processes, then the notion of bisimulation has to be *convergence sensitive*. A number of authors have considered the notion of convergence (or divergence) in a bisimulation framework. An early example is [15] and a more recent systematic study of the related axiomatisations is in [8]. However all these works *assume* a labelled transition system whose definition is not necessarily compelling. To cope with this problem, a well-known approach started in [11] and elaborated in [7] is to look at ‘internal’ reductions and at a basic notion of ‘barb’ and then to close under contexts thus producing a notion of ‘contextual’ bisimulation. However, even the notion ‘barb’ is not always easy to define and justify (an attempt based on the concept of *bi-orthogonality* is described in [13]). It seems to us that a natural approach which applies to a wide variety of formalisms is to regard convergence (may-termination) as the ‘intrinsic’ basic observable automatically provided by the internal reduction relation.

Following these preliminary considerations, we are now in a position to describe our contribution.

1. We introduce a notion of contextual bisimulation for (T)CCS whose basic observable (or barb) is the may-termination predicate (section 2).
2. We provide various characterisations of this equivalence culminating in one based on a suitable ‘convergence-sensitive’ labelled bisimulation (section 3).
3. We derive from this characterisation that (section 4):
 - (a) the proposed equivalence coincides with the usual one on reactive processes.
 - (b) the embedding of CCS in TCCS is fully abstract (even for non-reactive processes).

The development will take place in the context of so called *weak* bisimulation [10] which is more interesting and challenging than *strong* bisimulation.

2 Convergence sensitive bisimulation

We write $P \rightarrow \cdot$ if $\exists P' (P \rightarrow P')$, otherwise we say that P has converged and write $P \downarrow$. We write $P \Downarrow$ if $\exists Q (P \Rightarrow Q \text{ and } Q \downarrow)$. Thus $P \Downarrow$ means that P may converge, *i.e.*, there is a reduction sequence to a process that has converged. Because $P \downarrow$ iff $P \xrightarrow{\text{tick}} \cdot$, we have that $P \Downarrow$ iff $P \xRightarrow{\text{tick}}$.

We denote with C, D, \dots one hole *static contexts* specified by the following grammar:

$$C ::= [] \mid C \mid P \mid \nu a C$$

We require that the notion of bisimulation we consider is preserved by the static contexts in the sense of [7].

Definition 4 (bisimulation) A symmetric relation \mathcal{R} on processes is a bisimulation if PRQ implies:

cxt for any static context C , $C[P]\mathcal{R}C[Q]$.

red $P \xrightarrow{\mu} P'$, $\mu \in \{\tau, \text{tick}\}$ implies $\exists Q' (Q \xrightarrow{\mu} Q' \text{ and } P'\mathcal{R}Q')$.

We denote with \approx the largest bisimulation.

Remark 5 For CCS processes, if $P \xrightarrow{\text{tick}} Q$ then $P = Q$. It follows that in the definition above, the condition **[red]** when $\mu = \text{tick}$ can be replaced by $P \Downarrow$ implies $Q \Downarrow$. This is obviously false for processes including the *else-next* operator.

In view of the previous remark, the definition of bisimulation is specialised to CCS processes by simply restricting the condition **[cxt]** to CCS static contexts. We denote with \approx_{ccs} the resulting largest bisimulation.

Next we remark that the observability of a particular kind of ‘barb’ or ‘commitment’ is entailed by the observation of convergence.

Definition 6 We say that P commits on a , and write $P \Downarrow_a$, if $P \Rightarrow P'$, $P' \Downarrow$, and $P' \xrightarrow{a} \cdot$.³

Proposition 7 If $P \approx Q$ and $P \Downarrow_a$ then $Q \Downarrow_a$.

PROOF. Suppose $P \Downarrow_a$ and $P \approx Q$. Then $P \Rightarrow P'$, $P' \Downarrow$, and $P' \xrightarrow{a} \cdot$. By definition of bisimulation, $Q \Rightarrow Q''$ and $P' \approx Q''$. Moreover, $Q'' \Rightarrow Q'$, $Q' \Downarrow$, $Q' \approx P' \approx Q''$. To show that $Q' \xrightarrow{a} \cdot$, consider the context $C = ([] \mid \bar{a}.\Omega)$. Then we have $C[P'] \Downarrow$, while $C[Q'] \Downarrow$ if and only if $Q' \xrightarrow{a} \cdot$. \square

Another interesting notion is that of *contextual convergence*.

Definition 8 We say that a process P is contextual convergent, and write $P \Downarrow_C$, if $\exists C (C[P] \Downarrow)$.

Clearly, $P \Downarrow$ implies $P \Downarrow_C$ but the converse fails taking, for instance, $(a + b) \mid \bar{a}.\Omega$. Contextual convergence, can be characterised as follows.

Proposition 9 The following conditions are equivalent:

- (1) $P \xrightarrow{\alpha_1} \dots \xrightarrow{\alpha_n} P'$ and $P' \Downarrow$.
- (2) $\exists Q$ CCS process $(P \mid Q) \Downarrow$.
- (3) $P \Downarrow_C$.

³Note that in this definition the process ‘commits’ on action a only when it has converged.

PROOF. (1 \Rightarrow 2) Suppose $P_0 \xrightarrow{\alpha_1} P_1 \cdots \xrightarrow{\alpha_n} P_n$ and $P_n \downarrow$. We build the process Q in (2) by induction on n . If $n = 0$ we can take $Q = 0$. Otherwise, suppose $n > 0$. By inductive hypothesis, there is Q_1 such that $(P_1 \mid Q_1) \downarrow$. We proceed by case analysis on the first action α_1 . If $\alpha_1 = \tau$ take $Q = Q_1$ and if $\alpha_1 = a$ take $Q = \bar{a}.Q_1$.

(2 \Rightarrow 3) Taking the static context $C = [] \mid Q$.

(3 \Rightarrow 1) First, check by induction on a static context C that $P \xrightarrow{\tau} \cdot$ implies $C[P] \xrightarrow{\tau} \cdot$. Hence $C[P] \downarrow$ implies $P \downarrow$. Second, show that $C[P] \xrightarrow{\alpha} Q$ implies that $Q = C'[P']$ where C' is a static context and either $P = P'$ or $P \xrightarrow{\alpha'} P'$. Third, suppose $C[P] \xrightarrow{\tau} Q_1 \cdots \xrightarrow{\tau} Q_n$ with $Q_n \downarrow$. Show by induction on n that P can make a series of labelled transitions and reach a process which has converged. \square

Remark 10 *As shown by the characterisation above, the notion of contextual convergence is unchanged if we restrict our attention to contexts composed of CCS processes.*

We notice that a bisimulation never identifies a process which is contextual convergent with one which is not while identifying all processes which are not contextual convergent.

Proposition 11 (1) *If $P \approx Q$ and $P \downarrow_C$ then $Q \downarrow_C$.*

(2) *If $P \not\downarrow_C$ and $Q \not\downarrow_C$ then $P \approx Q$.*

PROOF. (1) If $P \downarrow_C$ then for some context C , $C[P] \downarrow$. By condition [cxt], we have that $C[P] \approx C[Q]$ and by condition [red] we derive that $C[Q] \downarrow$. Hence $Q \downarrow_C$.

(2) We notice that the relation $S = \{(P, Q) \mid P, Q \not\downarrow_C\}$ is a bisimulation. Indeed: (i) if $P \not\downarrow_C$ then $C[P] \not\downarrow_C$, (ii) if $P \Rightarrow P'$ and $P \not\downarrow_C$ then $P' \not\downarrow_C$, and (iii) if $P \not\downarrow_C$ then $P \not\xrightarrow{\text{tick}} \cdot$. \square

3 Characterisation

We characterise the (contextual and convergence sensitive) bisimulation introduced in definition 4 by means of a labelled bisimulation. The latter is obtained from the former by replacing condition [cxt] with a suitable condition [lab] on labelled transitions as defined in table 1.

Definition 12 (labelled bisimulation) *A symmetric relation \mathcal{R} on processes is a labelled bisimulation if $P\mathcal{R}Q$ implies:*

lab *if $P \downarrow_C$ and $P \xrightarrow{\alpha} P'$ then $Q \xrightarrow{\alpha} Q'$ and $P'\mathcal{R}Q'$ where $\alpha \in \{a, \tau\}$ and $\alpha = a$ if $P' \downarrow_C$.*

red *$P \xrightarrow{\mu} P'$, $\mu \in \{\tau, \text{tick}\}$ implies $\exists Q' (Q \xrightarrow{\mu} Q' \text{ and } P'\mathcal{R}Q')$.*

We denote with \approx^ℓ the largest labelled bisimulation.

Remark 13 (1) *In view of remarks 5 and 10, there is no need to specialise the definition of labelled bisimulation to CCS processes. By definition, the (timed) labelled bisimulation restricted to CCS processes is the same as the labelled bisimulation on (untimed) CCS processes.*

(2) *The predicate of contextual convergence \Downarrow_C plays an important role in the condition [lab]. To see why, suppose we replace it with the predicate \Downarrow and assume we denote with $\approx^{\ell\Downarrow}$ the resulting largest bisimulation. The following example shows that $\approx^{\ell\Downarrow}$ is not preserved by parallel composition. Consider*

$$P_1 = a.(b + c) \quad P_2 = a.b + a.c \quad Q = \bar{a}.(d + \Omega)$$

Then $(P_1 \mid Q) \approx^{\ell\Downarrow} (P_2 \mid Q)$ because both processes fail to converge. On the other hand, $(P_1 \mid Q) \mid \bar{d} \not\approx^{\ell\Downarrow} (P_2 \mid Q) \mid \bar{d}$ because the first may converge to $(b + c)$ which cannot be matched by the second process.

(3) *One may consider an asymmetric and equivalent definition of labelled bisimulation where strong transitions are matched by weak transitions. To check the equivalence, it is useful to note that $P \Downarrow_C$ and $P \xrightarrow{\alpha} P'$ implies $P' \Downarrow_C$.*

We provide a rather standard proof that bisimulation and labelled bisimulation coincide.

Proposition 14 *If $P \approx Q$ then $P \approx^{\ell} Q$.*

PROOF. We show that a bisimulation is a labelled bisimulation. We denote with $P \oplus Q$ the internal choice between P and Q which is definable, *e.g.*, as $\tau.P + \tau.Q$. Suppose $P \Downarrow_C$ and $P \xrightarrow{\alpha} P'$. We consider a context $C = [] \mid T$ where $T = \bar{a}((b \oplus 0) \oplus c)$ and b, c are fresh names (not occurring in P, Q). We know $C[P] \approx C[Q]$ and $C[P] \Rightarrow (P' \mid (b \oplus 0))$. Thus $C[Q] \Rightarrow (Q' \mid T')$ where either $Q \xrightarrow{\alpha} Q'$ and $T \xrightarrow{\bar{a}} T'$ or $Q \Rightarrow Q'$ and $T = T'$.

- Suppose $P' \Downarrow_C$. Then $(P' \mid (b \oplus 0)) \Downarrow_C$ and, by proposition 11, $(Q' \mid T') \Downarrow_C$. The latter implies that $Q' \Downarrow_C$. By contradiction, suppose $Q' \Downarrow_C$, that is $(Q' \mid R) \Downarrow$. Then $(Q' \mid T') \mid R \mid \bar{T}' \Downarrow$ (contradiction!) where we take $\bar{T}' = \bar{a}$ if $T' = T$ and $\bar{T}' = 0$ otherwise. Hence, $P' \approx Q'$ as required.

- Suppose $P' \Downarrow_C$. If $Q \xrightarrow{\alpha} Q'$ and $T \xrightarrow{\bar{a}} T'$ then we show that it must be that $T' = (b \oplus 0)$. This is because if $P' \Downarrow_C$ then $P' \mid (b \oplus 0) \Downarrow_C$ which in turn implies that for some R (not containing the names b or c), $(P' \mid (b \oplus 0) \mid R) \Downarrow_b$. By proposition 7, we must have $Q'' = (Q' \mid T') \mid R \Downarrow_b$. Thus T' cannot be 0 and it cannot be $(b \oplus 0) \oplus c$, for otherwise $Q'' \Downarrow_c$ which cannot be matched by $(P' \mid (b \oplus 0) \mid R)$. Further, we have $P' \mid (b \oplus 0) \xrightarrow{\tau} P' \mid 0 (= P')$. So $(Q' \mid (b \oplus 0)) \xrightarrow{\tau} (Q' \mid T'')$ and $P' \approx (Q' \mid T'')$. The latter entails that $T'' = 0$.

On the other hand, we show that $Q \xrightarrow{\tau} Q'$ and $T = T'$ is impossible. Reasoning as above, we have $(P' \mid (b \oplus 0) \mid R) \Downarrow_b$. But then if $(Q' \mid T) \mid R \Downarrow_b$ we shall also have $(Q' \mid T) \mid R \Downarrow_c$. \square

The following lemma relates contextual convergence to labelled bisimulation (cf. the similar proposition 11).

Lemma 15 (1) *If $P \approx^\ell Q$ and $P \Downarrow_C$ then $Q \Downarrow_C$.*

(2) *If $P \not\Downarrow_C$ and $Q \not\Downarrow_C$ then $P \approx^\ell Q$.*

PROOF. (1) By proposition 9, if $P \Downarrow_C$ then $P \xrightarrow{\alpha_1} \dots \xrightarrow{\alpha_n} P'$ and $P' \downarrow$. By definition of labelled bisimulation we should have $Q \xrightarrow{\alpha_1} \dots \xrightarrow{\alpha_n} Q'$ and $P' \approx^\ell Q'$. Then $P' \xrightarrow{\text{tick}} \cdot$ entails $Q' \xrightarrow{\text{tick}}$, and therefore $Q \Downarrow_C$.

(2) By proposition 11, $P, Q \not\Downarrow_C$ implies $P \approx Q$, and by proposition 14 we conclude that $P \approx^\ell Q$. \square

Proposition 16 *If $P \approx^\ell Q$ then $P \approx Q$.*

PROOF. We show that labelled bisimulation is preserved by static contexts. In view of remark 13(3), we shall work with an asymmetric definition of bisimulation. With respect to this definition, we show that the following relations are labelled bisimulations:

$$\{(va P, va Q) \mid P \approx^\ell Q\} \cup \approx^\ell, \quad \{(P \mid R, Q \mid R) \mid P \approx^\ell Q\} \cup \approx^\ell.$$

The case for restriction is a routine verification so we focus on parallel composition. Suppose $(P \mid R) \xrightarrow{\mu} \cdot$. We proceed by case analysis.

- $(P \mid R) \xrightarrow{\alpha} (P \mid R')$ because $R \xrightarrow{\alpha} R'$. Then $(Q \mid R) \xrightarrow{\alpha} (Q \mid R')$.
- $(P \mid R) \xrightarrow{\text{tick}} (P' \mid R')$ because $P \xrightarrow{\text{tick}} P'$ and $R \xrightarrow{\text{tick}} R'$. Then $Q \xrightarrow{\text{tick}} Q'$ and $P' \approx^\ell Q'$. Hence $(Q \mid R) \xrightarrow{\text{tick}} (Q' \mid R')$.
- Suppose $(P \mid R) \Downarrow_C$ and $(P \mid R) \xrightarrow{a} (P' \mid R)$ because $P \xrightarrow{a} P'$. Then $P \Downarrow_C$ and therefore $Q \xrightarrow{\alpha} Q'$, $\alpha \in \{a, \tau\}$, and $P' \approx^\ell Q'$. If $P' \Downarrow_C$ then $\alpha = a$ and if $P' \not\Downarrow_C$ then $Q' \not\Downarrow_C$ hence $(P' \mid R) \approx^\ell (Q' \mid R)$ by lemma 15.
- Suppose $(P \mid R) \xrightarrow{\tau} (P' \mid R)$ because $P \xrightarrow{\tau} P'$. Then $Q \xrightarrow{\tau} Q'$ and $P' \approx^\ell Q'$.
- Suppose $(P \mid R) \xrightarrow{\tau} (P' \mid R')$ because $P \xrightarrow{a} P'$ and $R \xrightarrow{\bar{a}} R'$. If $P, P' \Downarrow_C$ then $Q \xrightarrow{a} Q'$ and $P' \approx^\ell Q'$. If $P \Downarrow_C$ and $P' \not\Downarrow_C$ then $Q \xrightarrow{\alpha} Q'$, $\alpha \in \{a, \tau\}$, and $P' \approx^\ell Q'$. But then $(P' \mid R), (Q' \mid R) \not\Downarrow_C$, and we apply lemma 15. If $P \not\Downarrow_C$ then $Q \not\Downarrow_C$ and therefore $(Q \mid R) \not\Downarrow_C$, and we apply again lemma 15. \square

4 Embedding CCS in TCCS

In view of the previous characterisation, we can now show that two CCS processes are bisimilar when observed in an untimed/asynchronous environment if and only if they are bisimilar in a timed/synchronous environment.

Theorem 17 *Suppose P, Q are CCS processes. Then $P \approx Q$ if and only if $P \approx_{ccs} Q$.*

PROOF. By propositions 14 and 16 we know that $\approx = \approx^\ell$. By remark 13(1), the labelled bisimulation on untimed processes coincides with the restriction to CCS processes of the timed labelled bisimulation. \square

It is also worth noting that the notion of convergence-sensitive bisimulation we have introduced collapses with the usual one on reactive processes.

Proposition 18 *Suppose P, Q are reactive processes. Then $P \approx Q$ if and only if $P \approx^u Q$.*

PROOF. We know that $\approx = \approx^\ell$. Reactive processes are closed under labelled transitions and on reactive processes the conditions that define labelled bisimulation coincide with the ones for the usual bisimulation. \square

To summarise, we contrast the usual labelled bisimulation with the contextual (or equivalently labelled) convergence sensitive bisimulation we have introduced. The two equivalences coincide on reactive processes while they are incomparable on non-reactive ones. The important property of the convergence sensitive bisimulation is that \approx restricted to CCS processes coincides with \approx_{ccs} . This property holds for the usual bisimulation on reactive processes but fails on non-reactive processes where the untimed bisimulation is strictly weaker than the timed one.

5 Conclusion

We have presented a natural notion of contextual and convergence sensitive bisimulation and we have shown that it can be characterised by a variant of the usual notion of labelled bisimulation relying on the concept of contextual convergence. Thanks to this characterisation we can show that (untimed) CCS processes are embedded fully abstractly into timed ones.

The notion of bisimulation we have introduced just requires the notions of reduction and static context as opposed to previous approaches that build on the notion of ‘labelled’ transition or on the notion of ‘barb’. It would be interesting to apply the proposed approach to other contexts where the notion of equivalence is unclear. Another related question is to see what happens if one additionally observes must-convergence (strong normalisation). Note that such a ‘must-convergence’ bisimulation is finer than the one considered here as it distinguishes $A = \tau.A + \tau.0$ from 0.

References

- [1] R. Amadio. The SL synchronous language, revisited. *Journal of Logic and Algebraic Programming*, 70:121-150, 2007.
- [2] R. Amadio. A synchronous π -calculus. *Information and Computation*, 205(9):1470–1490, 2007.
- [3] G. Berry, L. Cosserat. The Esterel synchronous programming language and its mathematical semantics. INRIA technical report 842, Sophia-Antipolis, 1988.

- [4] G. Berry and G. Gonthier. The Esterel synchronous programming language. *Science of computer programming*, 19(2):87–152, 1992.
- [5] F. Boussinot and R. De Simone. The SL synchronous language. *IEEE Trans. on Software Engineering*, 22(4):256–266, 1996.
- [6] M. Hennessy, T. Reagan. A process algebra of timed systems. *Information and Computation*, 117(2):221-239, 1995.
- [7] K. Honda and N. Yoshida. On reduction-based process semantics. *Theoretical Computer Science*, 151(2):437-486, 1995.
- [8] M. Lohrey, P. D’Argenio, and H. Hermanns: Axiomatising Divergence. In Proc. ICALP, SLNCS 2380:585-596, 2002.
- [9] R. Milner. Calculi for synchrony and asynchrony. *Theoretical Computer Science*, 25(3):267–310, 1983.
- [10] R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.
- [11] R. Milner and D. Sangiorgi. Barbed bisimulation. In Proc. *ICALP*, SLNCS 623:685–695, 1992.
- [12] X. Nicolin, J. Sifakis. The algebra of timed processes (ATP): theory and application. *Information and Computation*, 114(1):131-178, 1994.
- [13] J. Rathke, V. Sassone and P. Sobocinski. Semantic barbs and biorthogonality. In Proc. FoSSaCS 2007, SLNCS 4423:302-316, 2007.
- [14] W. Yi. A calculus of real time systems. PhD thesis. Chalmers University, 1991.
- [15] D. Walker. Bisimulation and divergence. *Information and Computation*, 85:202-241, 1990.