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

Roberto M. Amadio

Université Paris Diderot (Paris 7)

Motivations

1. Build a notion of bisimulation just from *internal reduction* and (static) *contexts*.

   Internal reduction induces a notion of *convergence*. Take this as *basic observable* (rather than *labels* or *barbs*).

   NB Previous approaches (we are aware of) take (some form of) convergence *and* labels (or barbs) as basic observable

   Ref Walker 90.

2. Have an 'intuitive' semantic framework that spans both *asynchronous/untimed* and *synchronous/timed* models. In particular understand how asynchronous/untimed behaviours can be *embedded fully abstractly* into synchronous/timed ones.

   Ref Milner 83, CCS vs. SCCS.

Time and convergence

Time passes exactly when no internal computation is possible, *i.e.*, when the system has *converged*.

Ref Berry-Cosserat 88, Yi 91, Nicolin-Sifakis 94, Hennessy-Regan 95...

Formally, suppose $\stackrel{\tau}{\rightarrow}$ describes internal computation and $\stackrel{\mathsf{tick}}{\rightarrow}$ describes the passage of one unit of time. Then

$$P \stackrel{\mathsf{tick}}{\rightarrow} \cdot \;\text{ iff }\; P \stackrel{\tau}{\not\rightarrow} \cdot$$

An elementary playground: timed CCS

CCS processes with *else_next* operator

$$P ::= 0 \parallel a.P \parallel (P \mid P) \parallel P \triangleright P \parallel \ldots$$

'Sequential' evaluation context

$$E ::= [\,] \parallel E \triangleright P$$

Internal reduction Assuming | AC:

$$E[a.P] \mid E'[\overline{a}.Q] \mid R \xrightarrow{\tau} (P \mid Q) \mid R$$

Passage of time

$$\frac{}{0 \xrightarrow{\mathsf{tick}} 0} \qquad \frac{}{a.P \xrightarrow{\mathsf{tick}} a.P}$$

$$\frac{}{E[a.P] \triangleright Q \xrightarrow{\mathsf{tick}} Q} \qquad \frac{P_i \xrightarrow{\mathsf{tick}} Q_i \quad (P_1 \mid P_2) \stackrel{\tau}{\not\rightarrow} \cdot}{(P_1 \mid P_2) \xrightarrow{\mathsf{tick}} (Q_1 \mid Q_2)}$$

Untimed vs. timed behaviours

Definition P is a *CCS process* if it does not contain the *else_next* operator.

Remark The language is designed so that CCS processes are a good candidate to represent *untimed/asynchronous behaviours*:

- CCS processes are closed under internal reduction (and 'labelled' reduction too).
- CCS processes are *time insensitive*: if P is a CCS process and $P \stackrel{\mathsf{tick}}{\rightarrow} Q$ then $P = Q$.

Questions (cf. motivations)

1. Can we define a bisimulation semantics starting from the $\xrightarrow{\tau}$ and $\xrightarrow{\mathsf{tick}}$ reductions and a notion of (static) context?

2. Is the resulting equivalence on TCCS processes *conservative* over the equivalence on CCS processes?

Convergence sensitive bisimulation

Some notation:

- $P \downarrow$ if $P \stackrel{\tau}{\not\to} \cdot$.
- $P \Downarrow$ if $P \stackrel{\tau}{\Rightarrow} Q$ and $Q \downarrow$.
- Static contexts: $C ::= [\,] \parallel C \mid P \parallel \nu a C$.

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

$$\textbf{cxt} \qquad \frac{C \text{ static context}}{C[P] \mathcal{R} C[Q]}$$

$$\textbf{red} \qquad \frac{P \stackrel{\mu}{\Rightarrow} P', \quad \mu \in \{\tau, \mathsf{tick}\}}{Q \stackrel{\mu}{\Rightarrow} Q', \quad P' \mathcal{R} Q'}$$

where $\stackrel{\mu}{\Rightarrow}$ is ‘weak’ reduction. Let $\approx$ be the largest bisimulation.

Remarks

- On CCS processes, $P \stackrel{\mathsf{tick}}{\rightarrow} \cdot$ iff $P \downarrow$.
- Hence on CCS processes we have: (1) *may convergence* as basic observable, (2) bisimulation under *internal reduction*, and (3) preservation under *static CCS contexts*.
- Because CCS contexts are less than TCCS contexts, it is not obvious that the TCCS bisimulation is *conservative* over the CCS one.
- Indeed, conservativity *fails* for:
  - *testing semantics* **Ref** Hennessy-Regan 95
    $a.(b + c.b) + a.(d + c.d) =^{test}_{CCS} a.(b + c.d) + a.(d + c.b)$.
  - the usual *convergence-insensitive bisimulation*:
    $0 \approx^{u}_{ccs} \Omega$.

Some useful concepts

Stable commitment $P \Downarrow_a$ if $P \stackrel{\tau}{\Rightarrow} Q$, $Q \downarrow$ and Q is ready to communicate on a.

Contextual convergence $P \Downarrow_C$ if $\exists\, C$ static context $C[P] \Downarrow$.

NB $P \Downarrow_C$ iff $\exists\, Q$ CCS process $(P \mid Q) \Downarrow$.

$\Omega = \tau.\tau \ldots$ is the prototypical process such that $\Omega \not\Downarrow_C$.

Some properties of bisimulation

1. Bisimilar processes have the same stable commitments.

   Reduce to the situation: $P \downarrow$, $Q \downarrow$, $P \xrightarrow{a} \cdot$.

   Take $C = ([\;] \mid \overline{a}.\Omega)$.

   Then $C[P] \not\Downarrow$.

   Note $C[Q] \Downarrow$ if and only if $Q \stackrel{a}{\not\rightarrow} \cdot$.

2. Bisimilar processes cannot be separated by the contextual convergence.

3. All processes which are not contextual convergent are identified. E.g. $\Omega \approx (\Omega \mid a)$.

   Intuition: divergence makes all observations impossible.

A labelled bisimulation

- To characterise the bisimulation we rely on the *usual labelled transition system* for (T)CCS.
- We replace the *stability under context* condition with a suitable condition on labelled transitions.
- A symmetric relation $\mathcal{R}$ on processes is a *labelled bisimulation* if $P\mathcal{R}Q$ implies (a is a communication action):

$$\textbf{lab} \quad \frac{P \Downarrow_C, \qquad P \stackrel{a}{\Rightarrow} P'}{Q \stackrel{\alpha}{\Rightarrow} Q', \quad \alpha \in \{a, \tau\}, \quad P'\mathcal{R}Q', \quad (P' \Downarrow_C \supset \ \alpha = a)}$$

$$\textbf{red} \quad \frac{P \stackrel{\mu}{\Rightarrow} P', \mu \in \{\tau, \mathsf{tick}\}}{Q \stackrel{\mu}{\Rightarrow} Q', \quad P'\mathcal{R}Q'}$$

Denote with $\approx^{\ell}$ the largest labelled bisimulation.

Characterisation

The labelled bisimulation characterizes the (contextual) bisimulation

$$P \approx Q \text{ iff } P \approx^{\ell} Q$$

($\Leftarrow$) $\approx^{\ell}$ is preserved by static contexts.

NB This fails if we replace $\Downarrow_C$ by $\Downarrow$.

($\Rightarrow$) Show that $\approx$ is a labelled bisimulation.

Key step: if $P \stackrel{a}{\Rightarrow} P'$ then, for b, c fresh, consider:

$$C = [\;] \mid T, \quad T = \overline{a}.((b \oplus 0) \oplus c)$$

Corollaries of characterisation

Def A process P is *reactive* if whenever $P \stackrel{\mu_1}{\Rightarrow} \cdots \stackrel{\mu_n}{\Rightarrow} Q$ we have that all internal (τ) reduction sequences from Q terminate.

1. Bisimulation on timed CCS is *conservative* over the bisimulation on (untimed) CCS.

2. On *reactive* (T)CCS processes, the bisimulation *coincides* with the usual ones (denoted with $\approx^u$ and $\approx^u_{ccs}$).

3. On *non-reactive* TCCS processes, $P \approx^u Q$ implies $P \approx Q$.

4. The converse fails recalling $\Omega \approx (\Omega \mid a)$.

5. On *non-reactive* CCS processes, $P \approx^u Q$ implies both $P \approx^u_{ccs} Q$ and $P \approx Q$ while $\approx^u_{ccs}$ and $\approx$ are *incomparable*.

Summary

- Internal reduction provides automatically an observable: *may convergence*.
- Observing may convergence is quite natural in a *timed* context.
- The characterisation of the resulting equivalence relies on the concept of *contextual convergence*.

More work

- One can play a similar game when additionally observing *divergence* or equivalently *must convergence* (= strong normalisation). One distinguishes $A = \tau.A + \tau.0$ from 0.

- The approach seems to work in other contexts. E.g.
  - CCS with *asynchronous communication*.
  - TCCS with *signal based communication*:

$$\mathsf{emit}(s) = \overline{s}.\mathsf{emit}(s) \triangleright 0$$

Remark on previous work

- The usual bisimulation $\approx^u$ restricted to CCS processes corresponds to a known bisimulation where one observes both labels and may convergence.
- This bisimulation is called *stable* in Lohrey et al. 02 and it provides another way to embed fully abstractly CCS into TCCS.

Why testing equivalence is not conservative

Let

$$P = a.(b + c.b) + a.(d + c.d) \qquad Q = a.(b + c.d) + a.(d + c.b)$$

Then

$$P =_{CCS}^{test} Q \qquad P \neq_{TCCS}^{test} Q$$

Indeed, consider the test $T = \overline{a}.(\overline{b}.\top \triangleright \overline{c}.\overline{b}.\top)$.

$Q \mid T$ must produce the 'observable' action $\top$ in the first or second instant while $P \mid T$ may fail to do that.

The *else_next* operator allows to test that a process in a stable state cannot perform a certain action

Why contextual convergence is needed

- Consider

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

- Then $(P_1 \mid Q) \approx^{\ell\Downarrow} (P_2 \mid Q)$ because both processes fail to converge.

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