Concurrency 3

CCS : Bisimulation "up to", weak and strong bisimulation

Pierre-Louis Curien (CNRS – Université Paris 7)

MPRI concurrency course 2006/2007 with :

Francesco Zappa Nardelli (INRIA Rocquencourt)
Catuscia Palamidessi (INRIA - Futurs)
Roberto Amadio (Paris 7)

(http://mpri.master.univ-paris7.fr/C-2-3.html)

Bisimilarity is not trace equivalence

As automata $P = a \cdot (b + c)$ and $Q = a \cdot b + a \cdot c$ recognize the same language $\{ab, ac\}$ of traces.

As processes, they are not bisimilar (Q does not even simulate P). P keeps the choice after performing a, Q not.

Think of a as inserting 40 cents, b as getting tea and c as getting coffee. Imagine a vending machine with a slot for a and two buttons for b and c. The machine allows you to press b (resp. c) only if action b (resp. c) can be performed. As a customer you will prefer P.

.

Strucural equivalence

Exercice 1 Show that structural equivalence $\equiv$ is included in (strong) bisimulation $\sim$.

Variations on bisimilarity (1/3)

A bisimulation up to $\sim$ is a relation $\mathcal{R}$ such that for all P, Q :

$P \, \mathcal{R} \, Q \Rightarrow \forall \, \mu, P' \; (P \xrightarrow{\mu} P' \Rightarrow \exists \, Q' \; Q \xrightarrow{\mu} Q'$ and $P' \sim \mathcal{R} \sim Q')$ and conversely

If $\mathcal{R}$ is strong bisimulation up to $\sim$, then $\mathcal{R} \subseteq \sim$.

Exercice 2 Prove it.

Hence, to show $P \sim Q$, it is enough to find a bisimulation up to $\sim$ such that $P \, \mathcal{R} \, Q$.

Variations on bisimilarity (2/3)

As an example, take

$$
\begin{aligned}
\mathit{Sem} &= \mathsf{P} \cdot \mathit{Sem}' \qquad & \mathit{Sem}^0 &= \mathsf{P} \cdot \mathit{Sem}^1 \\
\mathit{Sem}' &= \mathsf{V} \cdot \mathit{Sem} \qquad & \mathit{Sem}^1 &= \mathsf{P} \cdot \mathit{Sem}^2 + \mathsf{V} \cdot \mathit{Sem}^0 \\
& & \mathit{Sem}^2 &= \mathsf{P} \cdot \mathit{Sem}^3 + \mathsf{V} \cdot \mathit{Sem}^1 \\
& & \mathit{Sem}^3 &= \mathsf{V} \cdot \mathit{Sem}^2
\end{aligned}
$$

Then a (strong) bisimulation up-to witnessing that $(\mathit{Sem}|\mathit{Sem}|\mathit{Sem}) \sim \mathit{Sem}^0$ is, say :

$$
\begin{aligned}
\{\ & ((\mathit{Sem}|\mathit{Sem}|\mathit{Sem})\ ,\ \mathit{Sem}^0) \\
& ((\mathit{Sem}'|\mathit{Sem}|\mathit{Sem})\ ,\ \mathit{Sem}^1) \\
& ((\mathit{Sem}'|\mathit{Sem}|\mathit{Sem}')\ ,\ \mathit{Sem}^2) \\
& ((\mathit{Sem}'|\mathit{Sem}'|\mathit{Sem}')\ ,\ \mathit{Sem}^3)\ \}
\end{aligned}
$$

Variations on bisimilarity (3/3)

For any LTS, one can change Act to $Act^\star$ (words of actions), setting

$$P \xrightarrow{s} Q \text{ if } \begin{cases} s = \mu_1 \ldots \mu_n \text{ and} \\ (\exists\, P_1, \ldots, P_n \;\; (P_n = Q \text{ and } P \xrightarrow{\mu_1} P_1 \ldots \xrightarrow{\mu_n} P_n)) \end{cases}$$

This yields a new LTS, call it LTS* (the path LTS) . Then the notions of LTS and of LTS* bisimulation coincide.

From strong to weak bisimulation (1/2)

Take the LTS of CCS, with $Act = L \cup \overline{L} \cup \{tau\}$, call it Strong. The bisimulation for this system is called strong bisimulation.

Take Strong* (its path LTS).

Consider the following LTS, call it Weak†, with the same set of actions as Strong* :

$$P \stackrel{s}{\Rightarrow} Q \text{ if and only if } (\exists t \;\; P \stackrel{t}{\rightarrow} Q \text{ and } \hat{s} = \hat{t})$$

where the function $s \mapsto \hat{s}$ is defined as follows :

$$\hat{\epsilon} = \epsilon \quad \hat{\tau} = \epsilon \quad \hat{\alpha} = \alpha \quad \hat{s\mu} = \hat{s}\hat{\mu}$$

The idea is that weak bisimulation is bisimulation with possibly τ actions interspersed.

Let Weak be the LTS on *Act* whose transitions are $P \stackrel{\mu}{\Rightarrow} Q$, that is :

$$P \stackrel{\tau}{\Rightarrow} Q \text{ if and only if } P \stackrel{\tau}{\rightarrow}^\star Q \qquad P \stackrel{\alpha}{\Rightarrow} Q \text{ if and only if } P \stackrel{\tau}{\rightarrow}^\star \stackrel{\alpha}{\rightarrow} \stackrel{\tau}{\rightarrow}^\star Q$$

Then one has Weak† = Weak*.

.

From strong to weak bisimulation (2/2)

None of the three equivalent definitions of weak bisimulation obtained from the LTS's (Weak, Weak†, Weak*) is practical. The following is a fourth, equivalent, and more tractable version :

A weak bisimulation is a relation $\mathcal{R}$ such that

$$P\,\mathcal{R}\,Q \Rightarrow \forall\,\mu, P' \ \ (P \xrightarrow{\mu} P' \Rightarrow \exists\, Q' \ \ Q \overset{\mu}{\Rightarrow} Q' \text{ and } P'\,\mathcal{R}\,Q') \text{ and conversely}$$

(Note the dissymetry between the use of $\xrightarrow{\mu}$ on the left and of $\overset{\mu}{\Rightarrow}$ on the right.)

Two processes are weakly bisimilar if (notation $P \approx Q$) if there exists a weak bisimulation $\mathcal{R}$ such that $P\,\mathcal{R}\,Q$.

Bisimulation is a congruence (1/6)

We define $\sim^*$ inductively by the following rules :

$$\frac{P \sim Q}{P \sim^* Q} \qquad \frac{P \sim^* Q}{Q \sim^* P} \qquad \frac{P \sim^* Q \quad Q \sim^* R}{P \sim^* R}$$

$$\frac{\forall\, i \in I \ \ P_i \sim^* Q_i}{\Sigma_{i\in I}\mu_i \cdot P_i \sim^* \Sigma_{i\in I}\mu_i \cdot Q_i} \qquad \frac{P_1 \sim^* Q_1 \ \ P_2 \sim^* Q_2}{P_1 \mid P_2 \sim^* Q_1 \mid Q_2} \qquad \frac{P \sim^* Q}{(\nu a)P \sim^* (\nu a)Q}$$

Clearly $\sim \subseteq \sim^*$ and $\sim^*$ is a congruence, by construction. It is enough to show that $\sim^*$ is a bisimulation (since then $\sim = \sim^*$ is a congruence).

Bisimulation is a congruence (2/6)

Proof by rule induction. We look at case $P_1 \mid P_2 \sim^* Q_1 \mid Q_2$:

1. (backward) decomposition phase : if $P_1|P_2 \xrightarrow{\mu} P'$, then $P' = P'_1|P'_2$ and three cases may occur, corresponding to the three rules for parallel composition in the labelled operational semantics. We only consider the synchronisation case. If $P_1 \xrightarrow{a} P'_1$ and $P_2 \xrightarrow{\overline{a}} P'_2$, then

2. by induction there exists Q'_1 such that $Q_1 \xrightarrow{a} Q'_1$ and $P'_1 \sim^* Q'_1$, and there exists Q'_2 such that $Q_2 \xrightarrow{\overline{a}} Q'_2$ and $P'_2 \sim^* Q'_2$.

3. Hence (forward phase) we have $Q_1 \mid Q_2 \xrightarrow{\tau} Q'_1 \mid Q'_2$ and $P'_1 \mid P'_2 \sim^* Q'_1 \mid Q'_2$.

Bisimulation is a congruence (3/6)

$\approx$ is also a congruence (for our choice of language with guarded sums).

Same proof technique : define $\approx^*$. For the forward phase, we use the following properties, which are true :

$$
\begin{array}{lcl}
(P \stackrel{\mu}{\Rightarrow} P') & \Rightarrow & ((\nu a)P \stackrel{\mu}{\Rightarrow} (\nu a)Q') \\
(Q_1 \stackrel{\mu}{\Rightarrow} Q_1') & \Rightarrow & (Q_1 \mid Q_2 \stackrel{\mu}{\Rightarrow} Q_1' \mid Q_2) \\
(Q_1 \stackrel{a}{\Rightarrow} Q_1' \text{ and } Q_2 \stackrel{\overline{a}}{\Rightarrow} Q_2') & \Rightarrow & (Q_1 \mid Q_2 \stackrel{\tau}{\Rightarrow} Q_1' \mid Q_2')
\end{array}
$$

Bisimulation is a congruence (4/6)

Consider CCS with prefix and sums instead of guarded sums, i.e., replace $\Sigma_{i\in I}\mu_i \cdot P_i$ by two constructs $\Sigma_{i\in I}P_i$ and $a \cdot P$, with rules

$$\frac{P_i \xrightarrow{\mu} P_i'}{\Sigma_{i\in I}P_i \xrightarrow{\mu} P_i'} \qquad \frac{}{\mu \cdot P \xrightarrow{\mu} P}$$

Then strong bisimulation is a congruence, and weak bisimulation is not a congruence.

The problem arises because more processes (like $P + (Q|R)$) are allowed.

Bisimulation is a congruence (5/6)

What goes wrong is the sum rule ? For the forward phase, we would need the property :

$$(Q_1 \stackrel{\mu}{\Rightarrow} Q_1') \quad \Rightarrow \quad (Q_1 + Q_2 \stackrel{\mu}{\Rightarrow} Q_1')$$

which does not hold (take $\mu = \tau$ and $Q_1' = Q_1$).

Counter-example : $\tau \cdot a \cdot 0 + b \cdot 0 \not\approx a \cdot 0 + b \cdot 0$

Bisimulation is a congruence (6/6)

We have left out recursion, but even so we have :

Proposition : For any process S (possibly with recursive definitions) with free variables in $\vec{K}$:

$$\forall \vec{Q}, \vec{Q}' \ (\vec{Q} \approx \vec{Q}' \Rightarrow S[\vec{K} \leftarrow \vec{Q}] \approx S[\vec{K} \leftarrow \vec{Q}'])$$

The proof is by induction on the size of S. The non-recursion cases follow by congruence. For the recursive definition case $S = let\ \vec{L} = \vec{P}\ in\ L_j$, the trick is to unfold :

$$\begin{array}{lll} S[\vec{K} \leftarrow \vec{Q}] & =_{\mathsf{def}} & let\ \vec{L} = \vec{P}[\vec{K} \leftarrow \vec{Q}]\ in\ L_j \\ & \approx & P_j[\vec{K} \leftarrow \vec{Q}][\vec{L} \leftarrow (let\ \vec{L} = \vec{P}\ in\ \vec{L})] \\ & \approx_{\mathsf{ind}} & P_j[\vec{K} \leftarrow \vec{Q}'][\vec{L} \leftarrow (let\ \vec{L} = \vec{P}\ in\ \vec{L})] \\ & \approx & S[\vec{K} \leftarrow \vec{Q}'] \end{array}$$

Specification and weak bisimulation

HAMMER

$H = g \cdot H' \quad H' = p \cdot H$

JOBBER

$J = in \cdot S \quad S = \overline{g} \cdot U$

$U = \overline{p} \cdot F \quad F = \overline{out} \cdot J$

STRONG JOBBER

$K = in \cdot D \quad D = \overline{out} \cdot K$

We have : $(\nu g, h)(J \mid J \mid H) \approx K \mid K$. Their first actions are the same :

$$(\nu g, h)(J \mid J \mid H) \mathrel{\mathcal{R}} K \mid K \qquad (\nu g, h)(S \mid J \mid H) \mathrel{\mathcal{R}} D \mid K$$
$$(\nu g, h)(J \mid S \mid H) \mathrel{\mathcal{R}} K \mid D \qquad (\nu g, h)(S \mid S \mid H) \mathrel{\mathcal{R}} D \mid D$$

The only possible sequence of actions out of, say, $(\nu g, h)(S \mid S \mid H)$ is :

$$(\nu g, h)(S \mid S \mid H) \xrightarrow{\tau} (\nu g, h)(S \mid U \mid H') \xrightarrow{\tau} (\nu g, h)(S \mid U \mid H') \xrightarrow{\overline{out}} (S \mid J \mid H)$$

Hence we complete $\mathcal{R}$ with :

$$(\nu g, h)(S \mid U \mid H') \mathrel{\mathcal{R}} D \mid D \qquad (\nu g, h)(S \mid F \mid H) \mathrel{\mathcal{R}} D \mid D$$
$$(\nu g, h)(J \mid U \mid H') \mathrel{\mathcal{R}} K \mid D \qquad (\nu g, h)(J \mid F \mid H) \mathrel{\mathcal{R}} K \mid D$$
$$(\nu g, h)(U \mid J \mid H') \mathrel{\mathcal{R}} D \mid K \qquad (\nu g, h)(F \mid J \mid H) \mathrel{\mathcal{R}} D \mid K$$