並行分散計算特論 (3)

Shoji Yuen

2011/10/18

Equivalence property of $\sim$

(reflexivity) $p \sim p$ since the identity relation is a bisimulation.

(symmetry) Let $p \sim q$. There exists a bisimulation S such that $(p, q) \in S$. Since S is a bisimulation, both S and S^{-1} are simulation. Because $(S^{-1})^{-1} = S$, S^{-1} is also a bisimulation. This means $q \sim p$

(transitivity) Let $p \sim q$ and $q \sim r$. Simulations are closed under composition and seeing the fact that$(S_1 \circ S_2)^{-1} = S_2^{-1} \circ S_1^{-1}$, $p \sim r$.

Sequential process expression

$$P ::= A\langle a_1, \ldots, a_n\rangle \mid \Sigma_{i\in I}\alpha_i.P_i$$

I:finite index set
for each A: $A(\vec{a}) \overset{def}{=} P_A$

$\sum_{i\in\emptyset} \alpha_i.P_i$ is written as 0
$\sum_{\{0\}} \alpha_i.P_i$ is written as $\alpha_0.P_0$
$\sum_{i\in\{0,1\}} \alpha_i.P_i$ is written as $\alpha_0.P_0 + \alpha_1.P_1$

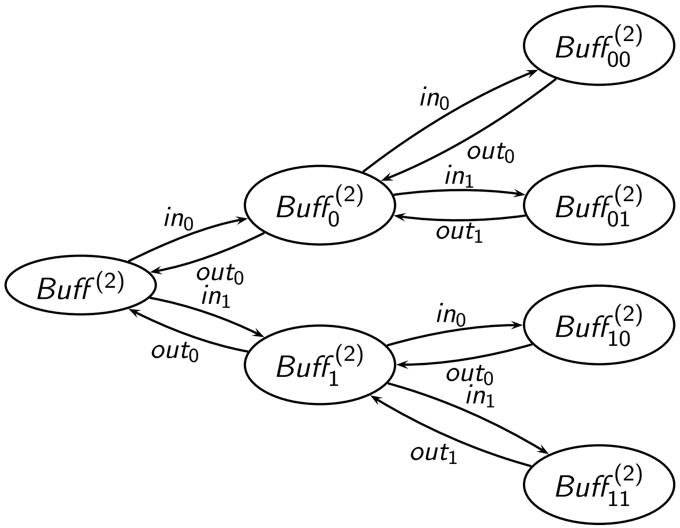
Process identifier A is for (mutual) recusion

Structural congrurence

- Unfolding of the process definition $P \equiv Q$ if $A\langle\vec{a}\rangle$ in P is replaced by $P_A(\vec{b}/\vec{a})$ where $A(x) \overset{\text{def}}{=} P_A$
  $a.A\langle a\rangle \equiv a.a.A\langle a\rangle$ where $A(x) \overset{\text{def}}{=} x.A\langle x\rangle$
- A-C property of operators (Choice)
  (Not necessary when choice is defined by $\sum$)
  $P + Q \equiv Q + P$, $(P + Q) + R \equiv P + (Q + R)$, $P + 0 \equiv P$

$\equiv$ is the trivial equivalence
We consider the quotiant of structural congruences over LTS

Boolean Buffer



Expression of Boolean Buffer

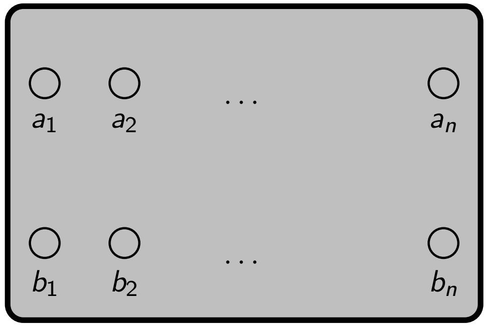
$$Buff^{(2)} \stackrel{def}{=} \sum_{i\in\{0,1\}} in_i.Buff^{(2)}_i$$

$$Buff^{(2)}_i \stackrel{def}{=} \overline{out_i}.Buff^{(2)} + \sum_{j\in\{0,1\}} in_j.Buff^{(2)}_{ji}$$

$$Buff^{(2)}_{ij} \stackrel{def}{=} \overline{out_j}.Buff^{(2)}_i$$

LIFO buffer

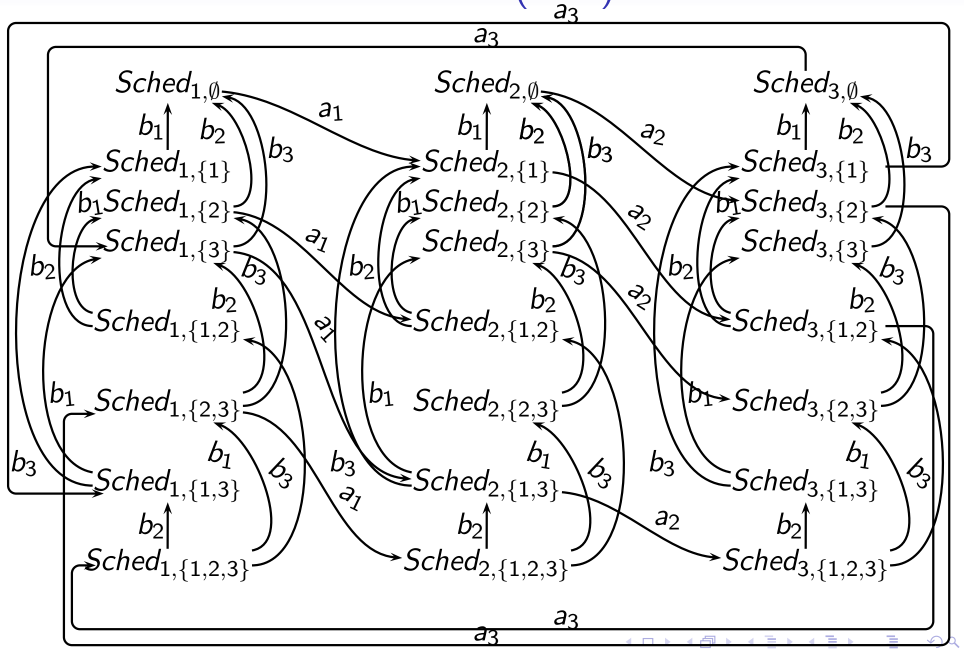
Scheduler specification



a_i: Job_i starts
b_j: job_j ends

Each job can be invoked only once at a time.
a_i is unavailable without pushing b_i.
A job can be overtaken by other jobs.
You can push b_j before b_i $(i < j)$ is pushed.

Scheduler (n=3)



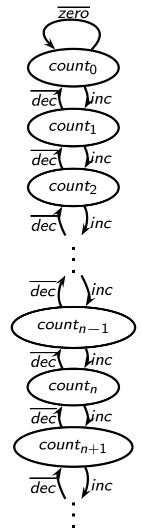
Expression of scheduler

$$Scheduler \stackrel{def}{=} Sched_{1,\emptyset}$$

$$Sched_{i,X} \stackrel{def}{=} \begin{cases} \sum_{j\in X} b_j.Sched_{i,X\setminus j} & i \in X \\ \sum_{j\in X} b_j.Sched_{i,X\setminus j} + a_i.Sched_{i+1,X\cup i} & i \notin X \end{cases}$$

$Sched_{i,X}$:a_i is in turn. X is the set of jobs whose b action is not yet done.

Counter



$$Count_0 \stackrel{def}{=} inc.Count_1 + \overline{zero}.Count_0$$

$$Count_{n+1} \stackrel{def}{=} inc.Count_{n+2} + \overline{dec}.Count_n$$

Counter is not in the regular class.
(More expressive than the sequential processes)
Infinite number of process identifiers.

Another charactrization

Definition

$P \sim' Q$ iff for all $a \in Act$,

(i) Whenever $p \xrightarrow{a} p'$, for some q', $q \xrightarrow{a} q'$ such that $p' \sim q'$

(ii) Whenever $q \xrightarrow{a} q'$, for some p', $p \xrightarrow{a} p'$ such that $p' \sim q'$

$\sim \subseteq \sim'$ by definition.

Proposition

$\sim'$ is a bisimulation

Corollary

$\sim' = \sim$

Bisimulation as a fixpoint

Definition

$\mathcal{F}(R) \stackrel{def}{=} \{(p,q) | \mathrm{T}_R(p,q)\}$

where $\mathrm{T}_R(p,q)$ holds if the followins conditions hold;

(i) Whenever $p \xrightarrow{a} q$, for some $q'.q \xrightarrow{a} q'$ and $(p',q') \in R$;

(ii) Whenever $q \xrightarrow{a} q'$, for some $p'.p \xrightarrow{a} p'$ and $(p',q') \in R$;

Fact

$\mathcal{F} : \mathcal{Q} \times \mathcal{Q} \to \mathcal{Q} \times \mathcal{Q}$ is monotonic.

Proposition

$\mathcal{F}(\sim) = \sim$ and $\sim$ is the largest fixpoint of $\mathcal{F}$.

Seqential Processes and $\sim$

(1) $p \sim q$ implies $a.p \sim a.q$
Show $R_1 = \{(a.p, a.q) | p \sim q\} \cup \sim$ is a bisimulation

(2) $p \sim q$ implies $a.p + r \sim a.q + r$
Show $R_2 = \{(a.p + r, a.q + r) | p \sim q\} \cup \sim$ is a bisimulation.