並行分散計算特論 (10)

Shoji Yuen

2011/12/6

Verification of Lottery

Theorem

$L_1 \approx \text{Lotspec}$

Proof sketch: Show the following $\mathcal{S}$ is a weak bisimulation:

$\mathcal{S} = \{(L_i, \text{Lotspec}) | 1 \leq i \leq n\} \cup \{(L'_i, b_i.\text{Lotspec})\}$

Jobshop

A problem of shared resource

Three kinds of Jobs: Easy, Neutral, and Difficult

$$A \stackrel{\text{def}}{=} i_E.A' + i_N.A' + i_D.A'$$
$$A' \stackrel{\text{def}}{=} \overline{o}.A$$

$$Agency \stackrel{\text{def}}{=} A|A$$

Jobber

An easy job is done with his hands.

A neutral job is done either with a hammer or with a mallet.

A difficult job is only done with a hammer

Suppose there is only one mallet and one hammer.

$$H \stackrel{\text{def}}{=} gh.H' \qquad M \stackrel{\text{def}}{=} gm.M'$$
$$H' \stackrel{\text{def}}{=} ph.H \qquad M' \stackrel{\text{def}}{=} gm.M$$

$$J \stackrel{\text{def}}{=} \sum_{X \in \{E,N,D\}} i_X.J_X$$
$$J_E \stackrel{\text{def}}{=} \overline{o}.J$$
$$J_N \stackrel{\text{def}}{=} \overline{gh}.\overline{ph}.J_E + \overline{gmpm}.J_E$$
$$J_D \stackrel{\text{def}}{=} \overline{gh}.\overline{ph}.JE$$

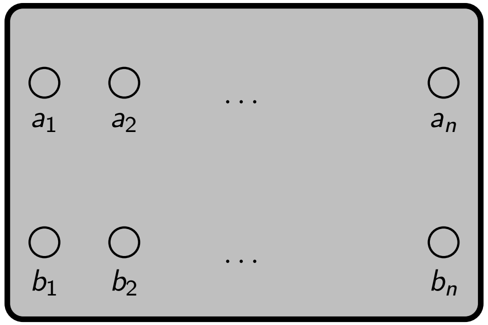
$$Jobshop \stackrel{\text{def}}{=} \text{new } \vec{t}(J|J|H|M)$$

Verification: Jobshop

Theorem

$$Agency \approx Jobshop$$

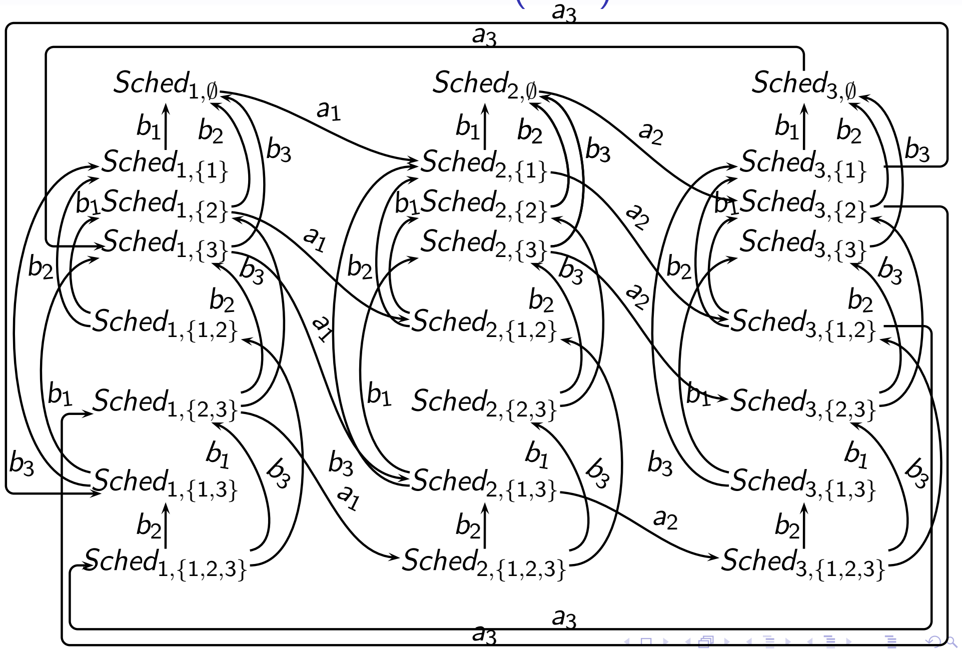
Scheduler specification



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 pusehd.

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.

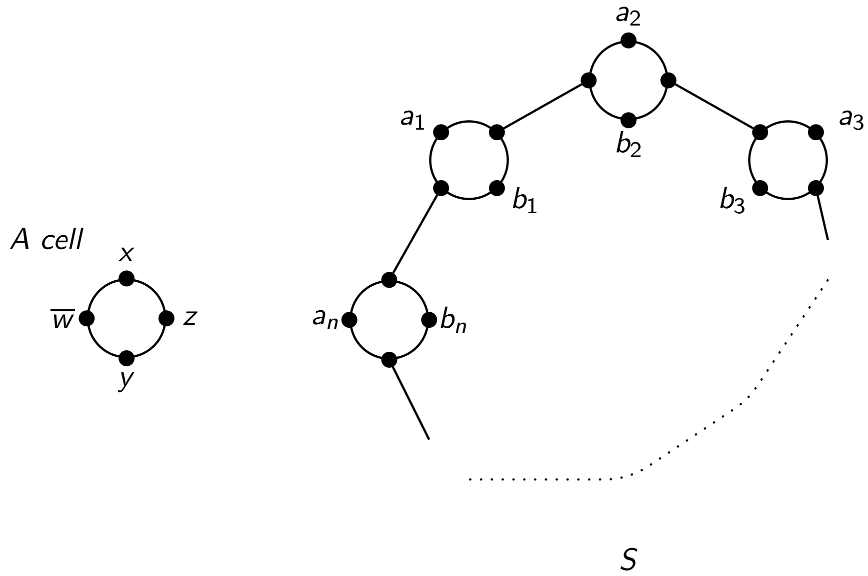
Schedular implementation by composition

$$
\begin{aligned}
A(x,y,z,w) &\stackrel{\text{def}}{=} x.B(x,y,z.w)\\
B(x,y,z,w) &\stackrel{\text{def}}{=} z.C(x,y,z.w)\\
C(x,y,z,w) &\stackrel{\text{def}}{=} y.D(x,y,z.w)\\
D(x,y,z,w) &\stackrel{\text{def}}{=} \overline{w}.A(x,y,z.w)
\end{aligned}
$$

$$
\begin{aligned}
A_i &\stackrel{\text{def}}{=} A(a_i, b_i, c_i, c_{i-1})\\
D_i &\stackrel{\text{def}}{=} D(a_i, b_i, c_i, c_{i-1})
\end{aligned}
$$

$$S \stackrel{\text{def}}{=} \text{new } \vec{c}(A_1|D_2|\cdots|D_n)$$

Schedular implementation by composition



Verification:Scheduler

Theorem

$$S \approx Scheduler$$

Buffer

$$Buff^{(n)} \stackrel{\text{def}}{=} \sum_u in_u.Buff_u^{(n)}$$

$$Buff_{\vec{v},w}^{(n)} \stackrel{\text{def}}{=} \begin{cases} \sum_u in_u.Buff_{u,\vec{v},w}^{(n)} + \overline{out_w}.Buff_{\vec{v}}^{(n)} & (|\vec{v}| < n-1)\\ \overline{out_w}.Buff_{\vec{v}}^{(n)} & (|\vec{v}| = n-1)\end{cases}$$

$$
\begin{aligned}
Cell &\stackrel{\text{def}}{=} \sum_v .in_v.Cell_v\\
Cell_v &\stackrel{\text{def}}{=} \overline{out_v}.Cell
\end{aligned}
$$

$$Cell \frown Cell \stackrel{\text{def}}{=} \text{new } \vec{m}(Cell\{m_v/out_v\}_{v\in V}|Cell\{m_v/in_v\}_{v\in V})$$

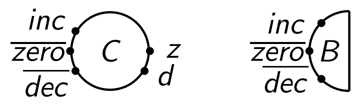
Verification:Buffer

Theorem

$$Buff^{(n)} \approx \overbrace{Cell \frown Cell \frown \cdots \frown Cell}^{n\ times}$$

Stack and counter

$$\begin{aligned} Count_0 &\stackrel{\text{def}}{=} inc.Count_1 + \overline{zero}.Count_0 \\ Count_{n+1} &\stackrel{\text{def}}{=} inc.Count_{n+2} + \overline{dec}.Count_n \end{aligned}$$



$$P \frown Q \stackrel{\text{def}}{=} \text{new } i'z'd'(P\{i'z'd'/izd\} | Q\{i'z'd'/inc\ zero\ dec\})$$

$$\begin{aligned} B &\stackrel{\text{def}}{=} inc.(C \frown B) + \overline{zero}.B \\ C &\stackrel{\text{def}}{=} inc.(C \frown C) + \overline{dec}.D \\ D &\stackrel{\text{def}}{=} d.C + z.B \end{aligned}$$

Verification:Counter

$$C_n \stackrel{\text{def}}{=} \overbrace{C \frown \cdots \frown C}^{n\ times} \frown B$$

Lemma

$$\begin{aligned} C_0 &\approx inc.C_1 + \overline{zero}.C_0 \\ C_{n+1} &\approx inc.C_{n+2} + \overline{dec}.C_n \end{aligned}$$

Theorem

$C_n \approx Count_n$

Verification:Stack

$$\begin{aligned} B &\stackrel{\text{def}}{=} \sum_v .inc_v.(C_v \frown B) + \overline{zero}.B \\ C_v &\stackrel{\text{def}}{=} \sum_u .inc_u.(C_u \frown C_v) + \overline{dec}.D_v \\ D_v &\stackrel{\text{def}}{=} d.C_v + z.B \end{aligned}$$

$C_{\vec{v}} = C_{v_1} \frown \cdots \frown C_{v_k} \frown B$, where $\vec{v} = v_1, \cdots, v_k$

$C_{\vec{v}} \approx Stack_{\vec{v}}$

$$\begin{aligned} Stack &\stackrel{\text{def}}{=} \sum_u push_u.Stack_u + \overline{empty}.Stack \\ Stack_{v,\vec{w}} &\stackrel{\text{def}}{=} \sum_u push_u.Stack_{u,v,\vec{w}} + \overline{pop_v}.Stack_{\vec{w}} \end{aligned}$$