Concurrency Semantics
Exercises 4

Prof. Nestmann, 2005

1. Warmup

1. Show that for all processes P, Q, we have

$$a.P + \tau.Q \approx a.P + \tau.(a.P + \tau.Q).$$

   Conclude that the process equation $X \approx a.P + \tau.X$ has infinitely many solutions for X, even when taken up to weak bisimilarity. Why does this not contradict Theorem 5.11?

2. Write down a weak bisimulation relating $\mathsf{Buf}^{(2)}(\, i_0, i_1, o_0, o_1\,)$ and $\mathsf{Buf}^{(1)}\langle\, i_0, i_1, o_0, o_1\,\rangle \;{}^{o_0,o_1}\!\frown^{i_0,i_1}\; \mathsf{Buf}^{(1)}\langle\, i_0, i_1, o_0, o_1\,\rangle$, as defined in last week's exercises.

2. Scheduler

Recall the sceduler example of session 3:

- A set of n processes $P_i, 0 \le i \le n-1$ is to be scheduled as follows:
  - P_i starts a task by sync'ing on a_i with the scheduler.
  - P_i completes a task by sync'ing on b_i with the scheduler.
- Concurrency is allowed:
  - Tasks of different P_i *may* run at the same time.
- There is a mutual exclusion property to be respected:
  - Each P_i *must not* run two tasks at a time.
  - For each i, a_i and b_i must occur cyclically.
- The scheduling of start permissions shall be *round-robin*:
  - The a_i are required to occur cyclically (initially, 0 starts)
- The overall system shall provide *maximal "progress"*:
  - the scheduling must permit any action at any time provided that the other properties are not violated.

The specification can be formalized as sequential non-deterministic process.
Let $i \in \{0\ldots, n-1\}$, $X \subseteq \{0\ldots, n-1\}$, $\vec{a} \stackrel{\text{def}}{=} a_0\ldots, a_{n-1}$ and $\vec{b} \stackrel{\text{def}}{=} b_0\ldots, b_{n-1}$. Then the process constants $\mathsf{Sspec}^{\mathsf{n}}_{\mathsf{i,X}}(\,\vec{a}, \vec{b}\,)$, defined by

$$\mathsf{Sspec}^{\mathsf{n}}_{\mathsf{i,X}}(\,\vec{a}, \vec{b}\,) := \begin{cases} \sum_{j\in X} b_j.\mathsf{Sspec}^{\mathsf{n}}_{\mathsf{i,X-j}}\langle\,\vec{a}, \vec{b}\,\rangle & (i \in X) \\ \sum_{j\in X} b_j.\mathsf{Sspec}^{\mathsf{n}}_{\mathsf{i,X-j}}\langle\,\vec{a}, \vec{b}\,\rangle + a_i.\mathsf{Sspec}^{\mathsf{n}}_{\mathsf{(i\oplus_n 1),X\cup i}}\langle\,\vec{a}, \vec{b}\,\rangle & (i \notin X) \end{cases}$$

each represents a state of a scheduler, where process i is the next to get the start permission, and where every $j \in X$ is currently running. The initial state is

$$\textbf{Scheduler}^n \stackrel{\text{def}}{=} \mathsf{Sspec}^{\mathsf{n}}_{\mathsf{0,\emptyset}}\langle\,\vec{a}, \vec{b}\,\rangle$$

Today, we will attempt to implement this scheduler as a parallel composition of scheduler "cells", one for each process. These cells, of the form $\mathsf{A}(\,a,b,c,d\,)$, synchronize with the controlled process on the channel a and b as above, and pass on (resp. receive) permission to start the associated process on the channel c (resp. d). Formally, we define process constants for a single cell as

$$\begin{array}{lllllll}\mathsf{A}(\,a,b,c,d\,) & := & a.\mathsf{C}\langle\,a,b,c,d\,\rangle & \qquad & \mathsf{C}(\,a,b,c,d\,) & := & c.\mathsf{B}\langle\,a,b,c,d\,\rangle \\ \mathsf{B}(\,a,b,c,d\,) & := & b.\mathsf{D}\langle\,a,b,c,d\,\rangle & & \mathsf{D}(\,a,b,c,d\,) & := & \overline{d}.\mathsf{A}\langle\,a,b,c,d\,\rangle.\end{array}$$

For a given number n of processes to schedule, we let $\vec{a} \stackrel{\text{def}}{=} a_0\ldots,a_{n-1}$, $\vec{b} \stackrel{\text{def}}{=} b_0\ldots,b_{n-1}$ and $\vec{c} \stackrel{\text{def}}{=} c_0\ldots,c_{n-1}$. The scheduler process is then defined as follows.

$$\begin{array}{lllllll}\mathbf{A}_i^n & \stackrel{\text{def}}{=} & \mathsf{A}\langle\,a_i,b_i,c_i,c_{i\ominus_n 1}\,\rangle & \qquad & \mathbf{B}_i^n & \stackrel{\text{def}}{=} & \mathsf{B}\langle\,a_i,b_i,c_i,c_{i\ominus_n 1}\,\rangle \\ \mathbf{C}_i^n & \stackrel{\text{def}}{=} & \mathsf{C}\langle\,a_i,b_i,c_i,c_{i\ominus_n 1}\,\rangle & & \mathbf{D}_i^n & \stackrel{\text{def}}{=} & \mathsf{D}\langle\,a_i,b_i,c_i,c_{i\ominus_n 1}\,\rangle\end{array}$$

$$\mathbf{Simpl}^n \stackrel{\text{def}}{=} (\boldsymbol{\nu}\vec{c})\,(\,\mathbf{A}_0^n|\mathbf{D}_1^n|\cdots|\mathbf{D}_{n-1}^n\,)$$

1. (a) Draw the transition diagrams for $\mathbf{Scheduler}^2$ and $\mathbf{Simpl}^2$.
   (b) Argue that the two processes are not weakly bisimilar.
   (c) Explain precisely why $\mathbf{Simpl}^2$ does not satisfy the (informal) specification of the scheduler (to the assistant or your neighbor).
2. Change the definition of $\mathbf{Simpl}^n$ as follows.

$$\mathsf{C}(\,a,b,c,d\,) := c.\mathsf{E}\langle\,a,b,c,d\,\rangle \tag{1}$$

$$\mathsf{D}(\,a,b,c,d\,) := d.\mathsf{A}\langle\,a,b,c,d\,\rangle \tag{2}$$

   Give a definition of E that solves the problem of 1 above.
3. Prove that $\mathbf{Scheduler}^2 \approx \mathbf{Simpl}^2$, using your new definition of C and E.
   Hint (try first by hand for a small n (2-4)):
   (a) Provide a uniform representation of the state space of a ring of cells: Let $\mathbf{Simpl}^n_{X_A,X_B,X_C,X_D,X_E}$ represent a state where the cells with numbers in $X_A \subseteq \{0,\ldots n-1\}$ are in state A, and so on for X_B, X_C, ...

$$\mathbf{Simpl}^n_{X_A,X_B,X_C,X_D,X_E} \stackrel{\text{def}}{=} (\boldsymbol{\nu}\vec{c})\Big(\prod_{i\in X_A}\mathsf{A}_i^n\langle\vec{a},\vec{b},\vec{c}\rangle\Big|\prod_{i\in X_B}\mathsf{B}_i^n\langle\vec{a},\vec{b},\vec{c}\rangle \Big|\prod_{i\in X_C}\mathsf{C}_i^n\langle\vec{a},\vec{b},\vec{c}\rangle\Big|\prod_{i\in X_D}\mathsf{D}_i^n\langle\vec{a},\vec{b},\vec{c}\rangle\Big|\prod_{i\in X_E}\mathsf{E}_i^n\langle\vec{a},\vec{b},\vec{c}\rangle\Big)$$

   (b) Assuming that X_A,X_B,X_C,X_D,X_E are mutually disjoint, give the transitions of $\mathbf{Simpl}^n_{X_A,X_B,X_C,X_D,X_E}$ (to other $\mathbf{Simpl}^n_{X'_A,X'_B,X'_C,X'_D,X'_E}$).
   Note that $|X_A|+|X_B|+|X_C|$ is invariant. What is the intuitive meaning of states where $|X_A|+|X_B|+|X_C| = 1$ and $X_A\cup X_B\cup X_C\cup X_D\cup X_E = \{0,\ldots n-1\}$?
   (c) Apply the expansion law (Proposition 3.13) once on an $\mathbf{Simpl}^n_{X_A,X_B,X_C,X_D,X_E}$ where X_A,X_B,X_C,X_D,X_E are mutually disjoint, $|X_A|+|X_B|+|X_C| = 1$, and $X_A\cup X_B\cup X_C\cup X_D\cup X_E = \{0,\ldots n-1\}$.

(d) Show that the relation S defined below, where $\{i\}, X_D, X_E$ are mutually disjoint and $\{i\} \cup X_D \cup X_E = \{0, \ldots n-1\}$, is a weak bisimulation up to $\sim$.

$$\begin{aligned} S \stackrel{\text{def}}{=} & \{(\mathsf{Sspec}^{\mathsf{n}}_{\mathsf{i},\mathsf{X_D}}\langle \vec{a}, \vec{b}\rangle, \mathbf{Simpl}^n_{\{i\},\emptyset,\emptyset,X_D,X_E} \mid i, X_D, X_E)\} \\ & \cup \{(\mathsf{Sspec}^{\mathsf{n}}_{\mathsf{i},\mathsf{X_D}\cup\{\mathsf{i}\}}\langle \vec{a}, \vec{b}\rangle, \mathbf{Simpl}^n_{\emptyset,\{i\},\emptyset,X_D,X_E} \mid i, X_D, X_E)\} \\ & \cup \{(\mathsf{Sspec}^{\mathsf{n}}_{\mathsf{i+1},\mathsf{X_D}\cup\{\mathsf{i}\}}\langle \vec{a}, \vec{b}\rangle, \mathbf{Simpl}^n_{\emptyset,\emptyset,\{i\},X_D,X_E} \mid i, X_D, X_E)\} \end{aligned}$$

(e) Conclude that $\mathbf{Scheduler}^n \approx \mathbf{Simpl}^n$.

3. An Alternative Definition of Weak Simulation

We extend the definition of $\stackrel{\cdot}{\Rightarrow}$ to tuples of visible actions in the following way:

$$\stackrel{\lambda_0\vec{\lambda}}{\Longrightarrow} \stackrel{\text{def}}{=} \stackrel{\lambda_0}{\Longrightarrow}\stackrel{\vec{\lambda}}{\Longrightarrow} .$$

We can then give an alternative definition of weak simulation:

Definition 3.1 (Weak Tuple Simulation)
Given any LTS $(\mathcal{Q}, \mathcal{T})$.
Let S be a binary relation over $\mathcal{Q}$.
Then S is said to be a weak tuple simulation
if, whenever $P\ S\ Q$ and $\vec{\lambda}$ is a tuple of visible actions, we have

- *if $P \stackrel{\vec{\lambda}}{\Longrightarrow} P'$ then there is $Q' \in \mathcal{P}$ such that $Q \stackrel{\vec{\lambda}}{\Longrightarrow} Q'$ and $P'\ S\ Q'$.*

Prove that an arbitrary relation S is a weak simulation if and only if it is a weak tuple simulation.