Concurrency Semantics
Exercises 5

Prof. Nestmann, 2005

1. Warmup

Handover A handover protocol for mobile phones can be defined as follows.

Let $\vec{n} \stackrel{\text{def}}{=} \mathrm{talk}_0, \mathrm{talk}_1, \mathrm{swch}_0, \mathrm{swch}_1, \mathrm{alrt}_0, \mathrm{alrt}_1, \mathrm{give}_0, \mathrm{give}_1$ and
$\vec{v} \stackrel{\text{def}}{=} \mathrm{talk}, \mathrm{swch}, \mathrm{give}, \mathrm{alrt}$.

$$
\begin{aligned}
\mathsf{Car}(\,\mathrm{talk}, \mathrm{swch}\,) &:= \overline{\mathrm{talk}}.\mathsf{Car}\langle\,\mathrm{talk}, \mathrm{swch}\,\rangle + \mathrm{swch}(t', s').\mathsf{Car}\langle\, t', s'\,\rangle \\
\mathsf{Base}(\,\vec{v}\,) &:= \mathrm{talk}.\mathsf{Base}\langle\,\vec{v}\,\rangle + \mathrm{give}(t', s').\overline{\mathrm{swch}}\langle t', s'\rangle.\mathsf{Idle}\langle\,\vec{v}\,\rangle \\
\mathsf{Idle}(\,\vec{v}\,) &:= \mathrm{alrt}.\mathsf{Base}\langle\,\vec{v}\,\rangle \\
\mathsf{Ctre}_i(\,\vec{n}\,) &:= \overline{\mathrm{give}_i}\langle \mathrm{talk}_{1-i}, \mathrm{swch}_{1-i}\rangle.\overline{\mathrm{alrt}_{1-i}}.\mathsf{Ctre}_{1-i}\langle\,\vec{n}\,\rangle \\
Syst_i &\stackrel{\text{def}}{=} (\boldsymbol{\nu}\vec{n})\Big(\, \mathsf{Car}\langle\,\mathrm{talk}_i, \mathrm{swch}_i\,\rangle \mid \mathsf{Base}\langle\,\mathrm{talk}_i, \mathrm{swch}_i, \mathrm{give}_i, \mathrm{alrt}_i\,\rangle \\
&\qquad \mid \mathsf{Idle}\langle\,\mathrm{talk}_{1-i}, \mathrm{swch}_{1-i}, \mathrm{give}_{1-i}, \mathrm{alrt}_{1-i}\,\rangle \mid \mathsf{Ctre}_i\langle\,\vec{n}\,\rangle\,\Big)
\end{aligned}
$$

Show that $Syst_0 \ (\rightarrow)^3 \ Syst_1$, giving formal derivations for the reactions (but not necessarily for all structural congruences).

2. Overtaking cars

A car $\mathsf{C}\langle\, n, b, f\,\rangle$ on a road is connected to its back and front neighbor through b and f, respectively, where n represents its identifier. The road is assumed to be infinite, so we ignore any boundary problem, and it is static in the sense that no cars may enter or leave the road.

1. Define $\mathsf{C}(\, x, b, f\,)$ such that a car may overtake another car. Beware of deadlocks and nested overtake attempts. You are not allowed to change the parameter x of instances of C.

2. Define fast cars $\mathsf{F}(\, x, b, f\,)$ that cannot be overtaken and slow cars $\mathsf{S}(\, x, b, f\,)$ that cannot overtake.

3. Value-passing CCS

We add matching to the syntax of $\mathcal{P}^{\mathrm{VP}}$: A sum M may now also be of the form $[\,u = v\,]P$, with the semantics

$$\textsc{equals:}\ \frac{}{[\,u = v\,]P \xrightarrow{\ \tau\ } P'} \text{ if } u = v$$

1. Define one- and two-place buffers of values $\mathcal{V}$ (with $\mathcal{V}$ finite) in V-P CCS.
   How many process constants do you need for the two-place buffer?
   How many nodes are there in the transition diagram of the two-place buffer?

For infinite sets of values $\mathcal{V}$, we can also define buffers without using matching by changing how process constants are handled.

2. We introduce process constants $\mathsf{A}\langle\,\vec{a}\ ;\ \vec{v}\,\rangle$ that also can take value parameters. Define defining equations, operational semantics, free and bound names, and substitution for such process constants.

3. Define one- and two-place buffers for this new kind of process constants.
   How many process constants do you need for the two-place buffer?
   How many nodes are there in the transition diagram of the two-place buffer?