TMC Session 11 @ 16/01/2002

U. Nestmann

EPFL-LAMP

Solution: Overtaking Cars

many implementations might be valid ...
... here's just one proposal

$$\text{Car}\langle\, x, b, f \,\rangle \quad \stackrel{\text{def}}{=}$$
$$\text{Fast}\langle\, x, b, f \,\rangle \quad \stackrel{\text{def}}{=}$$
$$\text{Slow}\langle\, x, b, f, b' \,\rangle \quad \stackrel{\text{def}}{=}$$

Buffers in New Clothes . . .

$$
\begin{aligned}
B(\,i,o\,) &\stackrel{\text{def}}{=} i(x).C\langle\,x,i,o\,\rangle\\
C(\,x,i,o\,) &\stackrel{\text{def}}{=} \quad \overline{o}\langle x\rangle.B\langle\,i,o\,\rangle\\
&\qquad + \; i(y).(\,C\langle\,y,i,o\,\rangle \frown C\langle\,x,i,o\,\rangle\,)
\end{aligned}
$$

where

$$
\begin{aligned}
X\langle\,i,o\,\rangle \frown Y\langle\,i,o\,\rangle &\stackrel{\text{def}}{=}\\
&(\boldsymbol{\nu} m)\,(\,X\langle\,i,o\,\rangle\{^m/_o\} \mid Y\langle\,i,o\,\rangle\{^m/_i\}\,)
\end{aligned}
$$

- Observe how much nicer value-passing is :-)
- Follow the sequence $\xrightarrow{i1}\xrightarrow{i2}\xrightarrow{\bar{o}2}\xrightarrow{\cdots}$

Elastic Buffers

Make the buffer elastic,
i.e., make empty cells disappear!

Several design decisions need to be taken concerning the question *when* an empty cell should cut itself out of a chain and die.

- if empty cell is next to a full/empty cell?
- if empty cell is left/right to a cell?
- should it be *allowed* (suicide)
  or *forced* (murder) to die?

Elastic Buffers (II)

$$
\begin{aligned}
B(\, i, l, o, r\,) &\stackrel{\text{def}}{=} i(x).C\langle\, x, i, l, o, r\,\rangle \\
&+ \ \ldots \\
C(\, x, i, l, o, r\,) &\stackrel{\text{def}}{=} \overline{o}\langle x\rangle.B\langle\, i, l, o, r\,\rangle \\
&+ \ i(y).(\, C\langle\, y, i, l, o, r\,\rangle \frown C\langle\, x, i, l, o, r\,\rangle\,) \\
&+ \ \ldots
\end{aligned}
$$

where

$$
X\langle\, i, l, o, r\,\rangle \frown Y\langle\, i, l, o, r\,\rangle \stackrel{\text{def}}{=} \ldots
$$

Syntax Conventions

$$\begin{array}{lll} \mathcal{N} & \text{names} & a, b, c \ldots, x, y, z \\ \mathcal{A} & \text{actions} & \pi \quad ::= \quad x(y) \quad \Big| \quad \overline{x}\langle y \rangle \quad \Big| \quad \tau \end{array}$$

- finite sequences $\vec{a} \ldots$
- parametric processes with defining equations are modeled through a more primitive notion of replication and name-passing

Syntax / Grammar

Definition: The set $\mathcal{P}$ of π-calculus proc. exp. is defined (precisely) by the following syntax:

$$\begin{array}{rcl} P & ::= & M \quad | \quad P|P \quad | \quad (\boldsymbol{\nu} a)\, P \quad | \quad \boxed{\,!\, P\,} \\ M & ::= & \mathbf{0} \quad | \quad \boxed{\pi.P} \quad | \quad M + M \end{array}$$

We use P, Q, P_i to stand for process expressions.

- $(\boldsymbol{\nu} ab)\, P$ abbreviates $(\boldsymbol{\nu} a)\, (\boldsymbol{\nu} b)\, P$
- $\sum_{i \in \{1..n\}} \pi_i.P_i$ abbreviates $\pi_1.P_1 + \ldots + \pi_n.P_n$

Bound and Free Names

- $(\boldsymbol{\nu} x)\, P$ $\boxed{\text{and } y(x).P}$ **bind** x in P
- x occurs **bound** in P, if it occurs
  in a subterm $(\boldsymbol{\nu} x)\, Q$ $\boxed{\text{or } y(x).P}$ of P
- x occurs **free** in P, if it occurs
  without enclosing $(\boldsymbol{\nu} x)\, Q$ $\boxed{\text{or } y(x).P}$ in P
- Note the use of parentheses (round brackets).
- Define $\mathrm{fn}(P)$ and $\mathrm{bn}(P)$ inductively on $\mathcal{P}$
  (sets of free/bound names of P) $\ldots$

Reaction (Example 9.2, [Mil99])

$$P \stackrel{\text{def}}{=} (\boldsymbol{\nu} z)\Big(\ (\overline{x}\langle y\rangle + z(w).\overline{w}\langle y\rangle) \,|\, x(u).\overline{u}\langle v\rangle \,|\, \overline{x}\langle z\rangle\ \Big)$$

$$P_1 \stackrel{\text{def}}{=}$$

$$P_2 \stackrel{\text{def}}{=}$$

$$P_3 \stackrel{\text{def}}{=}$$

Exercise 9.3:
Write down a process Q such that $Q|P_1$ has a redex, but $Q|P_2$ has no redex except that in P_2.

Process Contexts

Definition: A **process context** $C[\cdot]$ is (precisely) defined by the following syntax:

$$C[\cdot] \quad ::= \quad [\cdot] \quad | \quad \pi.C[\cdot] + M \quad | \quad M + \pi.C[\cdot]$$
$$| \quad (\boldsymbol{\nu} a)\, C[\cdot] \quad | \quad C[\cdot]|P \quad | \quad P|C[\cdot]$$
$$| \quad \boxed{!\, C[\cdot]}$$

The **elementary contexts** are

$\pi.[\cdot] + M$, $M + \pi.[\cdot]$, $(\boldsymbol{\nu} a)\, [\cdot]$, $[\cdot]|P$, $P|[\cdot]$, $\boxed{!\,[\cdot]}$.

Process Congruence

Definition:
Let $\cong$ be an equivalence relation over $\mathcal{P}$.
Then $\cong$ is said to be a **process congruence**, if it is preserved by all elementary contexts; i.e., if $P \cong Q$, then

$$\begin{array}{rclcrcl} \pi.P + M & \cong & \pi.Q + M & \qquad & P|R & \cong & Q|R \\ M + \pi.P & \cong & M + \pi.Q & & R|P & \cong & R|Q \\ (\boldsymbol{\nu} a)\, P & \cong & (\boldsymbol{\nu} a)\, Q & & \mathbf{!}\, P & \cong & \mathbf{!}\, Q \quad . \end{array}$$

Process congruence (II)

Proposition:

An arbitrary equivalence relation $\cong$ is a process congruence if and only if, for *all* contexts $C[\cdot]$, $P \cong Q$ implies $C[P] \cong C[Q]$.

Note:

For proving that an equivalence relation is a congruence, the elementary contexts suffice.

Structural Congruence

Definition: Structural congruence, written $\equiv$, is the (smallest) process congruence over $\mathcal{P}$ determined by the following equations.

1. $=_\alpha$ Now for two binders!
2. commutative monoids $(\mathcal{P}, +, \mathbf{0})$ and $(\mathcal{P}, |, \mathbf{0})$
3. $(\boldsymbol{\nu} a)\,(P|Q) \equiv P|(\boldsymbol{\nu} a)\,Q$, if $a \notin \mathrm{fn}(P)$
   $(\boldsymbol{\nu} a)\,\mathbf{0} \equiv \mathbf{0}$, $(\boldsymbol{\nu} ab)\,P \equiv (\boldsymbol{\nu} ba)\,P$
4. $\mathbf{!}\,P \equiv P \mid \mathbf{!}\,P$

Structural Congruence (II)

reflexive-symmetric-transitive context closure
(of a set of equations)

$$\frac{}{P = P} \qquad \frac{P = Q}{Q = P} \qquad \frac{P = Q \qquad Q = R}{P = R}$$

$$\frac{P = Q}{C[P] = C[Q]} \text{ for arbitrary ``process context'' } C[\cdot]$$

allows **equational reasoning**, i.e. any number of applications, in either direction, to any subterm

Standard Forms

Definition:

A process expression

$$(\boldsymbol{\nu}\vec{a})\,(\;M_1|\cdots|M_m|\,\mathbf{!}\,Q_1|\cdots|\,\mathbf{!}\,Q_n\;)$$

is in **standard form** if each M_i is a non-empty sum, and each Q_j is itself in standard form.

If $m = n = 0$ then the form is $\mathbf{0}$.

If $\vec{a}$ is empty then there is no restriction.

Theorem: Every process is structurally congruent to a standard form.

Reaction

Definition: $\rightarrow$ over $\mathcal{P}$ is generated by:

$$\textsc{tau}: \ \tau.P+M \ \rightarrow \ P$$

$$\textsc{react}: \ \overline{y}\langle z\rangle.P+M \mid y(x).Q+N \ \rightarrow \ \{^z/_x\}P \mid Q$$

$$\textsc{par}: \ \frac{P \ \rightarrow \ P'}{P|Q \ \rightarrow \ P'|Q} \qquad \textsc{res}: \ \frac{P \ \rightarrow \ P'}{(\boldsymbol{\nu} a)\,P \ \rightarrow \ (\boldsymbol{\nu} a)\,P'}$$

$$\textsc{struct}: \ \frac{P \ \rightarrow \ P'}{Q \ \rightarrow \ Q'} \ \textsc{if} \ P \equiv Q \ \textsc{and} \ P' \equiv Q'$$

Reaction (Exercise 9.18)

Exhibit the redex in

$$x(z).\overline{y}\langle z\rangle \mid \boldsymbol{!}\,(\boldsymbol{\nu} y)\,\overline{x}\langle y\rangle.Q$$

and give the result of the reaction.

Mobility ? “Flowgraphs” !

$$P = \overline{x}\langle z\rangle.P'$$
$$Q = x(y).Q'$$
$$R = \ldots z \ldots$$

Assume that $z \notin \mathrm{fn}(P')$.

Depict the transition

$$(\boldsymbol{\nu} z)\,(\ P|R\ )\mid Q \rightarrow P' \mid (\boldsymbol{\nu} z)\,(\ R|Q'\ )$$

as a flow graph (with scopes) and verify it using the reaction and congruence rules.

Polyadism

$$[\![\, \overline{y}\langle \vec{z} \rangle .P \,]\!] \stackrel{\text{def}}{=}$$

$$[\![\, y(\vec{x}).P \,]\!] \stackrel{\text{def}}{=}$$

$$[\![\, \overline{y}\langle \vec{z} \rangle .P \,]\!] \stackrel{\text{def}}{=}$$

$$[\![\, y(\vec{x}).P \,]\!] \stackrel{\text{def}}{=}$$

Recursion

$A(\,\vec{x}\,) \stackrel{\text{def}}{=} Q_A$, where $Q_A \stackrel{\text{def}}{=} \cdots A\langle\,\vec{u}\,\rangle \cdots A\langle\,\vec{v}\,\rangle \cdots$

can be used in: $P \stackrel{\text{def}}{=} \cdots A\langle\,\vec{y}\,\rangle \cdots A\langle\,\vec{z}\,\rangle \cdots$

can be modeled through:

1. invent a to stand for A
2. for any R, let $\widehat{R}$ denote the result of replacing any call $A\langle\,\vec{w}\,\rangle$ by $\overline{a}\langle\vec{w}\rangle$
3. replace P by

$$(\boldsymbol{\nu} a)\,(\;\widehat{P} \mid \mathbf{!}\,a(\vec{x}).\widehat{Q_A}\;)$$