TMC Session 11 @ 16/01/2002

U. Nestmann

EPFL-LAMP

Solution: Overtaking Cars

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

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

Buffers in New Clothes . . .

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

where

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

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

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)

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

where

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

Syntax Conventions

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

- finite sequences $\vec{a}$ ...
- 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:

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

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) ...

Reaction (Example 9.2, [Mil99])

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

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

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

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

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:

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

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

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

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. $=_\alpha$ $\boxed{\text{Now for two binders!}}$
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. $\boxed{!\,P \equiv P \mid !\,P}$

Structural Congruence (II)

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

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

$$\frac{P = Q}{C[P] = C[Q]} \quad \text{FOR ARBITRARY "PROCESS CONTEXT" } C[\cdot]$$

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

Standard Forms

Definition:
A process expression

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

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: $\longrightarrow$ over $\mathcal{P}$ is generated by:

$$\text{TAU: } \tau.P{+}M \ \longrightarrow\ P$$

$$\text{REACT: } \overline{y}\langle z\rangle.P{+}M \mid y(x).Q{+}N \ \longrightarrow\ \{^z\!/\!_x\}P \mid Q$$

$$\text{PAR: } \frac{P \longrightarrow P'}{P|Q \longrightarrow P'|Q} \qquad \text{RES: } \frac{P \longrightarrow P'}{(\boldsymbol{\nu}a)\,P \longrightarrow (\boldsymbol{\nu}a)\,P'}$$

$$\text{STRUCT: } \frac{P \longrightarrow P'}{Q \longrightarrow Q'} \ \text{IF } P \equiv Q \text{ AND } P' \equiv Q'$$

Reaction (Exercise 9.18)

Exhibit the redex in

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

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

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