Concurrency Semantics
Week 7

Course Notes 2005
EPFL – I&C

Uwe Nestmann
Johannes Borgström

April 29, 2005

6 Value-Passing CCS

6.1 Notation We use the following sets of entities with corresponding meta-variables:

$\mathcal{I}$	*process identifiers*	$\mathsf{A}, \mathsf{B} \ldots$
$\mathcal{N}$	*(channel) names*	$a, b, c \ldots$
$\mathcal{V}$	*values*	v, w
$\mathcal{X}$	*variables*	x, y, z
$\mathcal{A}$	*actions*	$\mu \ ::= \ \overline{a}\langle v\rangle \ \big\vert \ a(x) \ \big\vert \ \tau$

"negative" actions $\overline{a}\langle v\rangle$: *send name v over channel a.*

"positive" actions $a(x)$: *receive any value, say v, over channel a and "bind the result" to variable x.*

Binding results in *substitution* $\{^v/_x\}$ of the formal parameter x by the actual parameter v.

6.2 Definition (Value-Passing Processes) The set $\mathcal{P}^{\mathrm{VP}}$ is defined by the same grammar as the set $\mathcal{P}$ except that actions μ are now interpreted as in Notation 6.1.

6.3 Definition (Free and Bound Names)
The sets $\mathrm{fn}(P)$ and $\mathrm{bn}(P)$ are defined inductively precisely as for concurrent process expressions, except for the base cases of actions.

$$\mathrm{fn}(\mu) \quad \stackrel{\mathrm{def}}{=} \quad \begin{cases} \{a, \vec{v}\} & \text{if } \mu = \overline{a}\langle\vec{v}\rangle \\ \{a\} & \text{if } \mu = a(\vec{x}) \\ \emptyset & \text{if } \mu = \tau \end{cases}$$

$$\mathrm{bn}(\mu) \quad \stackrel{\mathrm{def}}{=} \quad \begin{cases} \emptyset & \text{if } \mu = \overline{a}\langle\vec{v}\rangle \\ \{\vec{x}\} & \text{if } \mu = a(\vec{x}) \\ \emptyset & \text{if } \mu = \tau \end{cases}$$

α-conversion now includes also the consistent renaming of input variables.
Substitution is defined accordingly, avoiding name-clashes via silent α-conversion wherever necessary.

6.4 Definition (Operational Semantics) The LTS $(\mathcal{P}^{\mathrm{VP}}, \mathcal{T})$ of sequential process expressions over $\mathcal{A}$ has $\mathcal{P}^{\mathrm{VP}}$ as states, and its transitions $\mathcal{T}$ are precisely generated by the operational semantics of $\mathcal{P}$, with the rules PRE and COM replaced by the following four rules:

$$\text{TAU: } \tau.P \xrightarrow{\tau} P$$

$$\text{OUT: } \overline{a}\langle v\rangle.P \xrightarrow{\overline{a}\langle v\rangle} P \qquad \text{INP: } \frac{v \in \mathcal{V}}{a(x).P \xrightarrow{av} \{^{v}\!/_{x}\}P}$$

$$\text{COM: } \frac{P \xrightarrow{\overline{a}\langle v\rangle} P' \qquad Q \xrightarrow{av} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

6.5 Definition (Translational Semantics) Let $\mathcal{V} = \{v_0 \cdots v_n\}$ be finite.

$[\![\ ]\!] : \mathcal{P}^{\text{VP}}$	$\rightarrow$	$\mathcal{P}$
$[\![\overline{a}\langle v\rangle.P]\!]$	$\stackrel{\text{def}}{=}$	$\overline{a_v}.[\![P]\!]$
$[\![a(x).P]\!]$	$\stackrel{\text{def}}{=}$	$\sum_{v\in\mathcal{V}} a_v.[\![\{^{v}\!/_{x}\}P]\!]$
$[\![\mathbf{0}]\!]$	$\stackrel{\text{def}}{=}$	$\mathbf{0}$
$[\![\mu.P]\!]$	$\stackrel{\text{def}}{=}$	$[\![\mu]\!].[\![P]\!]$
$[\![M_1 + M_2]\!]$	$\stackrel{\text{def}}{=}$	$[\![M_1]\!] + [\![M_2]\!]$
$[\![\mathsf{A}\langle\vec{a}\rangle]\!]$	$\stackrel{\text{def}}{=}$	$\mathsf{A}\langle[\![\vec{a}]\!]\rangle$
$[\![P_1 \mid P_2]\!]$	$\stackrel{\text{def}}{=}$	$[\![P_1]\!] \mid [\![P_2]\!]$
$[\![(\boldsymbol{\nu}a)\,P]\!]$	$\stackrel{\text{def}}{=}$	$(\boldsymbol{\nu}[\![a]\!])\,[\![P]\!]$
$[\![a]\!]$	$\stackrel{\text{def}}{=}$	$a_{v_0} \cdots a_{v_n}$
$[\![a\vec{b}]\!]$	$\stackrel{\text{def}}{=}$	$[\![a]\!][\![\vec{b}]\!]$

Defining equations for process constants must also be translated.

$$[\![\mathsf{A}(\vec{x}) := M]\!] \stackrel{\text{def}}{=} \mathsf{A}([\![\vec{x}]\!]) := [\![M]\!]$$

6.6 Proposition Let $P, P' \in \mathcal{P}^{\text{VP}}$. Let:

$$[\![\overline{a}\langle v\rangle]\!] \stackrel{\text{def}}{=} a_v$$
$$[\![a_v]\!] \stackrel{\text{def}}{=} a_v$$
$$[\![\tau]\!] \stackrel{\text{def}}{=} \tau$$

Then $P \xrightarrow{\mu} P'$ iff $[\![P]\!] \xrightarrow{[\![\mu]\!]} [\![P']\!]$.

6.7 Notation (Polyadic Communication)
The *polyadic actions* $\overline{a}\langle\vec{v}\rangle$ and $a(\vec{x})$ (with $\vec{x}$ pairwise different) *transmit many values at a time*.
All definitions are straightforwardly generalized.

6.8 Proposition (Value-Passing) CCS is Turing-powerful.

6.9 Proposition The halting problem for Turing machines can be reduced to the existence of inifinite sequences of internal transitions.

7 Pi Calculus

7.1 Notation We use the following sets of entities with corresponding meta-variables:

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

7.2 Definition (Mobile Processes)
The set $\mathcal{P}^\pi$ of *π-calculus process expressions* is defined (precisely) by the following syntax:

$$P \ ::= \ \mathsf{A}\langle\, \vec{a}\, \rangle \;\big|\; M \;\big|\; P|P \;\big|\; (\boldsymbol{\nu} a)\, P \;\big|\; \mathbf{!}\, P$$

$$M \ ::= \ \mathbf{0} \;\big|\; \pi.P \;\big|\; M + M$$

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

$$\begin{array}{lll} C[\cdot] & ::= & [\cdot] \;\big|\; \pi.C[\cdot] + M \;\big|\; M + \pi.C[\cdot] \\ & \big| & P|C[\cdot] \;\big|\; C[\cdot]|P \;\big|\; (\boldsymbol{\nu} a)\, C[\cdot] \;\big|\; \mathbf{!}\, C[\cdot] \end{array}$$

The *elementary process contexts* are

$$\begin{array}{ccc} (\boldsymbol{\nu} a)\, [\cdot] & \pi.[\cdot] + M & M + \pi.[\cdot] \\ \mathbf{!}\, [\cdot] & [\cdot] \mid P & P \mid [\cdot] \end{array}$$

7.4 Definition (Process Congruence)
Let $\cong$ be an equivalence relation over $\mathcal{P}^\pi$.
Then $\cong$ is said to be a process congruence,
if it is preserved by all elementary contexts;
i.e., $P \cong Q$ implies all of the following:

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

7.5 Proposition An arbitrary equivalence relation $\cong$ over processes $\mathcal{P}^\pi$ is a process congruence if and only if, for *all* contexts $C[\cdot]$, $P \cong Q$ implies $C[P] \cong C[Q]$.

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

1. $=_\alpha$ (now for two binding operators!)

2. commutative monoid laws for $(\mathcal{M}^\pi, +, \mathbf{0})$

3. commutative monoid laws for $(\mathcal{P}^\pi, \mid, \mathbf{0})$

4. $$\begin{array}{rcll} (\boldsymbol{\nu} a)\,(P \mid Q) & \equiv & P \mid (\boldsymbol{\nu} a)\,Q, & \text{if } a \notin \mathrm{fn}(P) \\ (\boldsymbol{\nu} ab)\,P & \equiv & (\boldsymbol{\nu} ba)\,P & \\ (\boldsymbol{\nu} a)\,\mathbf{0} & \equiv & \mathbf{0} & \end{array}$$

5. $\mathsf{A}\langle \vec{b} \rangle \equiv \{^{\vec{b}}/_{\vec{a}}\}M$, if $\mathsf{A}(\vec{a}) \stackrel{\text{def}}{=} M$.

6. $!\,P \equiv P|!\,P$

7.7 Definition (Standard Form)
A π-calculus process expression

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

where each M_i is a non-empty sum, is said to be in *standard form*, if each Q_j is itself in standard form.
If $m = 0$ then $M_1|\cdots|M_m$ means $\mathbf{0}$.
If $n = 0$ then $!\,Q_1|\cdots|!\,Q_n$ means $\mathbf{0}$.
If $\vec{a}$ is empty then there is no restriction.

7.8 Theorem Every π-calculus process expression is structurally congruent to some standard form.

7.9 Definition The reaction relation $\rightarrow$ over $\mathcal{P}^\pi$ is generated precisely by the following rules:

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

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

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

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