Concurrency: Languages, Programming and Theory

– Equivalences for CCS –

Session 11 – January 14, 2004

Uwe Nestmann

EPFL-LAMP

Bisimulation on CCS

- check out Session 4, again
- add 1+1 . . .

“Algebraic” Properties (I)

- $\beta.P + \beta.P + M \sim \beta.P + M$
- $(\boldsymbol{\nu} a)\, a.P \sim \mathbf{0}$
- $(\boldsymbol{\nu} a)\, \overline{a}.P \sim \mathbf{0}$
- $(\boldsymbol{\nu} c)\,(\ a.c.P \mid b.\overline{c}.Q\ ) \sim (\boldsymbol{\nu} c)\,(\ a.c.Q \mid b.\overline{c}.P\ )$
- ...
- Why *algebraic* ?

“Algebraic” Properties (II)

- $a \mid b \sim a.b + b.a$

- For all $P \in \mathcal{P}$, $P \sim \sum\{\ \beta.Q \mid P \xrightarrow{\beta} Q\ \}$.

- For all $n \geq 0$ and $P_1, \ldots, P_n \in \mathcal{P}$:

$$P_1|\cdots|P_n \sim \begin{cases} \sum\{\ \beta.(\ P_1|\cdots|P_i'|\cdots|P_n\ ) \\ \qquad \mid 1{\leq}i{\leq}n, P_i \xrightarrow{\beta} P_i'\ \} \\ + \\ \sum\{\ {\color{red}\tau}.(\ P_1|\cdots|P_i'|\cdots|P_j'|\cdots|P_n\ ) \\ \qquad \mid 1{\leq}i{<}j{\leq}n, P_i \xrightarrow{\lambda} P_i', P_j \xrightarrow{\overline{\lambda}} P_j'\ \} \end{cases}$$

“Algebraic” Properties (III)

For all $n \geq 0$, $P_1, \ldots, P_n \in \mathcal{P}$, and $\vec{a}$:

$$(\boldsymbol{\nu}\vec{a})\,(\;P_1|\cdots|P_n\;) \sim \left\{ \begin{array}{l} \sum\{\;\beta.(\boldsymbol{\nu}\vec{a})\,(\;P_1|\cdots|P_i'|\cdots|P_n\;) \\ \qquad\quad |\;1{\leq}i{\leq}n, P_i \xrightarrow{\beta} P_i',\ \text{and}\ \beta,\overline{\beta} \notin \vec{a}\;\} \\ + \\ \sum\{\;{\color{red}\tau}.(\boldsymbol{\nu}\vec{a})\,(\;P_1|\cdots|P_i'|\cdots|P_j'|\cdots|P_n\;) \\ \qquad\quad |\;1{\leq}i{<}j{\leq}n, P_i \xrightarrow{\lambda} P_i', P_j \xrightarrow{\overline{\lambda}} P_j'\;\} \end{array} \right.$$

Expansion Law ! (also called: *Interleaving*)
Compare to the notions of *standard forms* in Milner’s book:
every process term can be transformed into a form
that matches the left-hand side of the above equation.

Process Contexts

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

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

The **elementary contexts** are
$\alpha.[\cdot]+M$, $M+\alpha.[\cdot]$, $(\boldsymbol{\nu} a)\,[\cdot]$, $[\cdot]|P$, $P|[\cdot]$.

$C[Q]$ denotes the result of filling the hole $[\cdot]$ of $C[\cdot]$ with process Q.

Process congruence

Definition:(Process congruence)
Let $\cong$ be an *equivalence relation* over $\mathcal{P}$.

Then $\cong$ is said to be a ***process congruence***, if
for *all* contexts $C[\cdot]$,
$P \cong Q$ implies $C[P] \cong C[Q]$.

Process congruence (II)

Proposition:

An arbitrary equivalence relation $\cong$ is a process congruence if, and only if, it is preserved by all *elementary contexts*; i.e., if $P \cong Q$, then

$$\begin{array}{rclcrcl} \alpha.P + M & \cong & \alpha.Q + M & \qquad & P|R & \cong & Q|R \\ M + \alpha.P & \cong & M + \alpha.Q & & R|P & \cong & R|Q \end{array}$$

$$(\boldsymbol{\nu} a)\,P \;\cong\; (\boldsymbol{\nu} a)\,Q \;.$$

Note:

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

Congruence Properties

Proposition:

Bisimilarity is a process congruence, i.e., . . .

Towards Observation Equivalence

Let us assume that our LTSs may dispose of a single distinguished *internal action* symbol, say: τ, as is the case for our language of concurrent process expressions. Then:

“Different internal behavior” should “not count” !

Definition:(observations / weak actions)

1. $\Rightarrow \stackrel{\text{def}}{=} \xrightarrow{\tau} {}^*$
2. $\stackrel{\lambda}{\Longrightarrow} \stackrel{\text{def}}{=} \Rightarrow \xrightarrow{\lambda} \Rightarrow$

Weak Simulation

Definition:
$\mathcal{S}$ is a weak simulation **iff**, whenever $P \; \mathcal{S} \; Q$,

- if $P \xrightarrow{\tau} P'$ then there is $Q' \in \mathcal{P}$
  such that $Q \Rightarrow Q'$ and $P' \; \mathcal{S} \; Q'$.
- if $P \xrightarrow{\lambda} P'$ then there is $Q' \in \mathcal{P}$
  such that $Q \stackrel{\lambda}{\Longrightarrow} Q'$ and $P' \; \mathcal{S} \; Q'$.

q **weakly simulates** p,
if there is a weak simulation $\mathcal{S}$ such that $p \; \mathcal{S} \; q$.

Example:
Prove that $Q = \tau.a.\tau.b.Q$ weakly simulates $P = a.b.P$.
Prove that $P = a.b.P$ weakly simulates $Q = \tau.a.\tau.b.Q$.

Weak Bisimulation

Definition:(* straightforward / should be no surprise *)
A binary relation $\mathcal{B}$ is *a* **weak bisimulation**
if both $\mathcal{B}$ and its converse $\mathcal{B}^{-1}$ are weak simulations.

P and Q are **weakly bisimilar**, **weakly equivalent**, or **observation equivalent**, written $P \approx Q$,
if there exists a weak bisimulation $\mathcal{B}$ with $P \; \mathcal{B} \; Q$.

Alternatively:

$$\approx \stackrel{\text{def}}{=} \bigcup \{ \; \mathcal{B} \mid \mathcal{B} \text{ is weak bisimulation } \}$$

Proposition:

1. $\approx$ is itself a weak bisimulation.
2. $\approx$ is an equivalence relation.

Strong vs Weak

1. every strong simulation is also a weak one
2. $P \sim Q$ implies $P \approx Q$

   Examples ?

   Proof ?

Example

$$
\begin{aligned}
A &\stackrel{\mathrm{def}}{=} a.A' \quad (= a.\overline{b}.A) \\
A' &\stackrel{\mathrm{def}}{=} \overline{b}.A \\
B &\stackrel{\mathrm{def}}{=} b.B' \quad (= b.\overline{c}.B) \\
B' &\stackrel{\mathrm{def}}{=} \overline{c}.B
\end{aligned}
$$

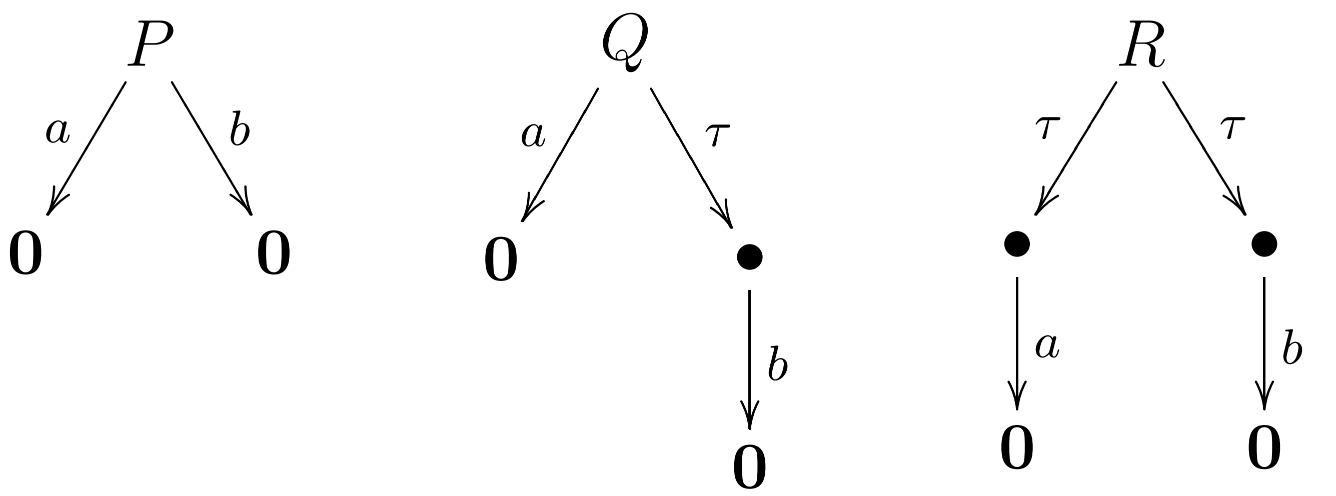
$$
\begin{aligned}
E &\stackrel{\mathrm{def}}{=} a.E' \\
E' &\stackrel{\mathrm{def}}{=} a.E'' + \overline{c}.E \\
E'' &\stackrel{\mathrm{def}}{=} \overline{c}.E'
\end{aligned}
$$

$\boxed{\text{Prove that } (\boldsymbol{\nu} b)(A|B) \approx E.}$

$$
\begin{tikzcd}
E \arrow[r, shift left, "a"] & E' \arrow[l, shift left, "\overline{c}"] \arrow[r, shift right, "a"'] & E'' \arrow[l, shift right, "\overline{c}"']
\end{tikzcd}
$$

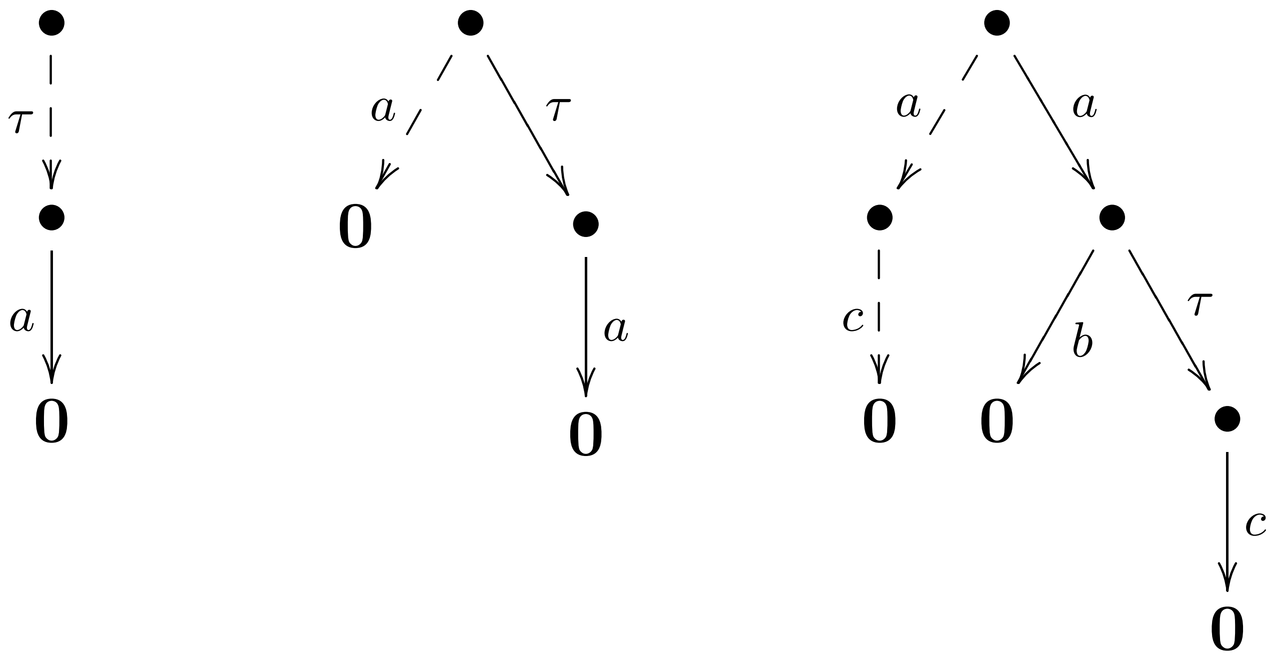
$$
\begin{tikzcd}
 & (\boldsymbol{\nu} b)(A'|B) \arrow[dd, "\tau"'] & \\
(\boldsymbol{\nu} b)(A|B) \arrow[ur, "a"] & & (\boldsymbol{\nu} b)(A'|B') \arrow[ul, "\overline{c}"'] \\
 & (\boldsymbol{\nu} b)(A|B') \arrow[ul, "\overline{c}"] \arrow[ur, "a"'] &
\end{tikzcd}
$$

Some Inequivalences



$P = a + b$ $\quad$ $Q = a + \tau.b$ $\quad$ $R = \tau.a + \tau.b$

Some Equivalences



$$\tau.a \approx a \qquad a + \tau.a \approx \tau.a \qquad a.c + a.(b + \tau.c) \approx a.(b + \tau.c)$$

Some Equations

Theorem:
Let P be any process.
Let N, M any summations. Then:

1. $P \approx \tau.P$
2. $M + N + \tau.N \approx M + \tau.N$
3. $M + \alpha.P + \alpha(\tau.P + N) \approx M + \alpha(\tau.P + N)$

Congruence Properties

Proposition:

Weak bisimilarity is a process congruence, i.e., . . .

Example:

- Observe $b \approx \tau.b$!
- Let $C[\cdot] \; = a + [\cdot]$.

  Compare $C[b] = \boxed{a + b \overset{?}{\approx} a + \tau.b} = C[\tau.b]$!

Two-Place Buffers

$\mathsf{Buff}_s^{(1)}(\,\vec{a}\,)$	:	1-place buffer containing s, where $\vec{a} = \{\mathsf{in}, \mathsf{out}\}$
$\mathsf{Buff}_\epsilon^{(1)}(\,\vec{a}\,)$	$\stackrel{\mathrm{def}}{=}$	$\mathsf{in}(x).\mathsf{Buff}_x^{(1)}\langle\,\vec{a}\,\rangle$
$\mathsf{Buff}_v^{(1)}(\,\vec{a}\,)$	$\stackrel{\mathrm{def}}{=}$	$\overline{\mathsf{out}}\langle v\rangle.\mathsf{Buff}_\epsilon^{(1)}\langle\,\vec{a}\,\rangle$
$\mathsf{Buff}_s^{(2)}(\,\vec{a}\,)$	:	2-place buffer containing s — SPECIFICATION
$\mathsf{Buff}_\epsilon^{(2)}(\,\vec{a}\,)$	$\stackrel{\mathrm{def}}{=}$	$\mathsf{in}(x).\mathsf{Buff}_x^{(2)}\langle\,\vec{a}\,\rangle$
$\mathsf{Buff}_v^{(2)}(\,\vec{a}\,)$	$\stackrel{\mathrm{def}}{=}$	$\overline{\mathsf{out}}\langle v\rangle.\mathsf{Buff}_\epsilon^{(2)}\langle\,\vec{a}\,\rangle + \mathsf{in}(w).\mathsf{Buff}_{wv}^{(2)}\langle\,\vec{a}\,\rangle$
$\mathsf{Buff}_{wv}^{(2)}(\,\vec{a}\,)$	$\stackrel{\mathrm{def}}{=}$	$\overline{\mathsf{out}}\langle w\rangle.\mathsf{Buff}_v^{(2)}\langle\,\vec{a}\,\rangle$
$\mathsf{Bluff}_s^{(2)}(\,\vec{a}\,)$	:	2-place buffer containing s — IMPLEMENTATION
$\mathsf{Bluff}_\epsilon^{(2)}(\,\vec{a}\,)$	$\stackrel{\mathrm{def}}{=}$	$(\boldsymbol{\nu}\mathsf{x})\left(\,\mathsf{Buff}^{(1)}\langle\,\mathsf{in},\mathsf{x}\,\rangle \mid \mathsf{Buff}^{(1)}\langle\,\mathsf{x},\mathsf{out}\,\rangle\,\right)$

- prove that $\mathsf{Buff}_\epsilon^{(2)}\langle\,\vec{a}\,\rangle \approx \mathsf{Bluff}_\epsilon^{(2)}\langle\,\vec{a}\,\rangle$

Unique Solution of Equations

Theorem:

Let $\vec{X} = X_1, X_2, \ldots$ be a (possibly infinite) sequence of process variables. In the equations

$$\begin{array}{rcl} X_1 & \approx & \alpha_{11}.X_{k(11)} + \cdots + \alpha_{1n_1}.X_{k(1n_1)} \\ X_2 & \approx & \alpha_{21}.X_{k(11)} + \cdots + \alpha_{2n_1}.X_{k(2n_1)} \\ \cdots & \approx & \cdots \end{array}$$

assume that $\alpha_{ij} \neq \tau$. Then, up to $\approx$, there is a unique sequence $P_1, P_2, \ldots$ of processes which satisfies the equations.