Topics in Concurrency

Lecture 2

Jonathan Hayman

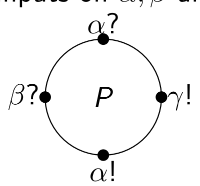
18 February 2013

The Calculus of Communicating Systems

- Introduced by Robin Milner in 1980
- First process calculus developed with its operational semantics
- Supports algebraic reasoning about equivalence
- Simplifies Dijkstra's Guarded Command Language by removing the store (store locations can be encoded as processes)
- Processes communicate by sending values (numbers) on channels.

Interface diagrams

- *Interface diagrams* describe the channels used by processes for input and output.
- The use of a channel by a process is called a *port*.
- Example: process P inputs on α, β and outputs on α, γ.



- Later examples: links between processes to represent the possibility of communication

Syntax of CCS

- Expressions: Arithmetic a and Boolean b
- Processes:

p	$::=$	**nil**	nil process
	$\mid$	$(\tau \to p)$	silent/internal action
	$\mid$	$(\alpha!a \to p)$	output
	$\mid$	$(\alpha?x \to p)$	input
	$\mid$	$(b \to p)$	Boolean guard
	$\mid$	$p_0 + p_1$	non-deterministic choice
	$\mid$	$p_0 \parallel p_1$	parallel composition
	$\mid$	$p \backslash L$	restriction (L a set of channel identifiers)
	$\mid$	$p[f]$	relabelling (f a function on channel identifiers)
	$\mid$	$P(a_1, \cdots, a_k)$	process identifier

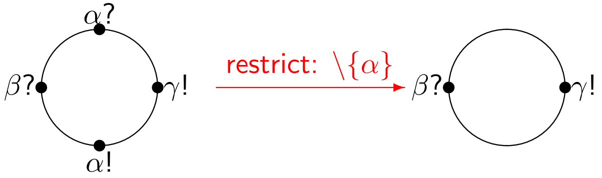
- Process definitions:

$$P(x_1, \cdots, x_k) \stackrel{\text{def}}{=} p$$

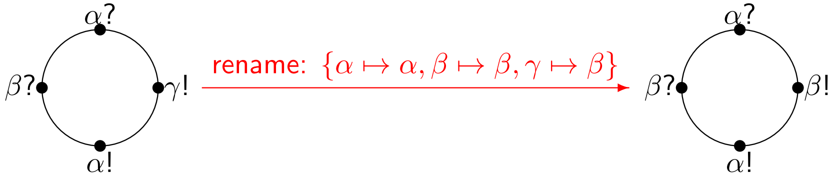
(free variables of $p \subseteq \{x_1, \cdots, x_n\}$)

Restriction and relabelling: interface diagrams

- $p \setminus L$: Disallow external interaction on channels in L



- $p[f]$: Rename external interface to channels by f



Operational semantics of CCS

- **Guarded processes**

$$(\tau \to p) \xrightarrow{\tau} p$$

$$\frac{a \to n}{(\alpha!a \to p) \xrightarrow{\alpha!n} p} \qquad (\alpha?x \to p) \xrightarrow{\alpha?n} p[n/x]$$

$$\frac{b \to \text{true} \qquad p \xrightarrow{\lambda} p'}{(b \to p) \xrightarrow{\lambda} p'}$$

- **Sum**

$$\frac{p_0 \xrightarrow{\lambda} p_0'}{p_0 + p_1 \xrightarrow{\lambda} p_0'} \qquad \frac{p_1 \xrightarrow{\lambda} p_1'}{p_0 + p_1 \xrightarrow{\lambda} p_1'}$$

- **Parallel composition**

$$\frac{p_0 \xrightarrow{\lambda} p_0'}{p_0 \parallel p_1 \xrightarrow{\lambda} p_0' \parallel p_1} \qquad \frac{p_0 \xrightarrow{\alpha?n} p_0' \qquad p_1 \xrightarrow{\alpha!n} p_1'}{p_0 \parallel p_1 \xrightarrow{\tau} p_0' \parallel p_1'}$$

$$\frac{p_1 \xrightarrow{\lambda} p_1'}{p_0 \parallel p_1 \xrightarrow{\lambda} p_0 \parallel p_1'} \qquad \frac{p_0 \xrightarrow{\alpha!n} p_0' \qquad p_1 \xrightarrow{\alpha?n} p_1'}{p_0 \parallel p_1 \xrightarrow{\tau} p_0' \parallel p_1'}$$

- **Restriction**

$$\frac{p \xrightarrow{\lambda} p'}{p \setminus L \xrightarrow{\lambda} p' \setminus L} \qquad \text{where if } \lambda \equiv \alpha?n \text{ or } \lambda \equiv \alpha!n \text{ then } \alpha \notin L$$

- **Relabelling**

$$\frac{p \xrightarrow{\lambda} p'}{p[f] \xrightarrow{f(\lambda)} p'[f]}$$

where f is extended to labels as $f(\tau)t = \tau$ and $f(a?n) = f(a)?n$ and $f(a!n) = f(a)!n$

- **Identifiers**

$$\frac{p[a_1/x_1, \cdots, a_n/x_n] \xrightarrow{\lambda} p'}{P(a_1, \cdots, a_n) \xrightarrow{\lambda} p'}$$

- **Nil process** no rules

A simple derivation from the operational semantics

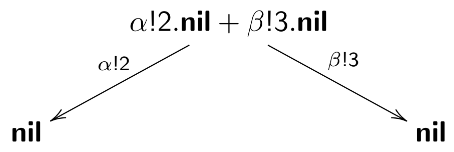
$$\cfrac{\cfrac{\cfrac{\cfrac{(\alpha!3 \to \textbf{nil}) \xrightarrow{\alpha!3} \textbf{nil}}{(\alpha!3 \to \textbf{nil}) + P \xrightarrow{\alpha!3} \textbf{nil}}}{((\alpha!3 \to \textbf{nil}) + P) \parallel (\tau \to \textbf{nil}) \xrightarrow{\alpha!3} \textbf{nil} \parallel (\tau \to \textbf{nil})} \qquad (\alpha?x \to \textbf{nil}) \xrightarrow{\alpha?3} \textbf{nil}}{(((\alpha!3 \to \textbf{nil}) + P) \parallel (\tau \to \textbf{nil})) \parallel (\alpha?x \to \textbf{nil}) \xrightarrow{\tau} (\textbf{nil} \parallel (\tau \to \textbf{nil})) \parallel \textbf{nil}}}{(((\alpha!3 \to \textbf{nil} + P) \parallel \tau \to \textbf{nil}) \parallel \alpha?x \to \textbf{nil}) \setminus \{\alpha\} \xrightarrow{\tau} ((\textbf{nil} \parallel \tau \to \textbf{nil}) \parallel \textbf{nil}) \setminus \{\alpha\}}$$

Final line: parallel composition is left-associative

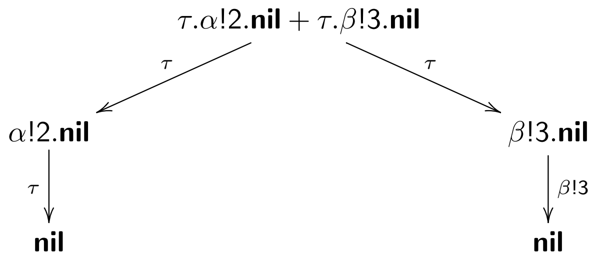
Further examples

(Write . for $\to$)
Each step justified by a derivation:

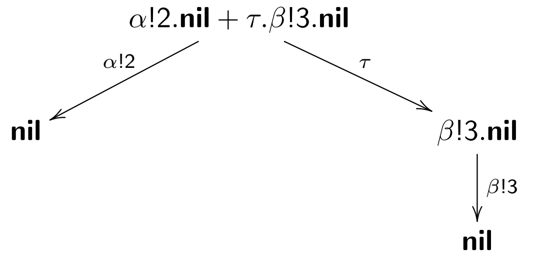
- External choice



- Internal choice



- Mixed choice



- Exercise:

$$\alpha!3.\mathbf{nil} \parallel \alpha?x.\beta!x.\mathbf{nil}$$

- Exercise:

$$(\alpha!3.\mathbf{nil} \parallel \alpha?x.\beta!x.\mathbf{nil}) \setminus \{\alpha\}$$

- Exercise:

$$(\alpha?x.\mathbf{nil} \parallel \beta!4)[\alpha \mapsto \beta, \beta \mapsto \beta]$$

- Exercise:

$$P(0) \text{ where } P(x) \stackrel{\text{def}}{=} x < 2 \to \alpha!x \to P$$

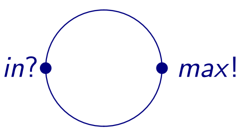
Conditionals

- Encoding of conditionals:

$$\texttt{if } b \texttt{ then } p_0 \texttt{ else } p_1 \equiv (b \to p_0) + (\neg b \to p_1)$$

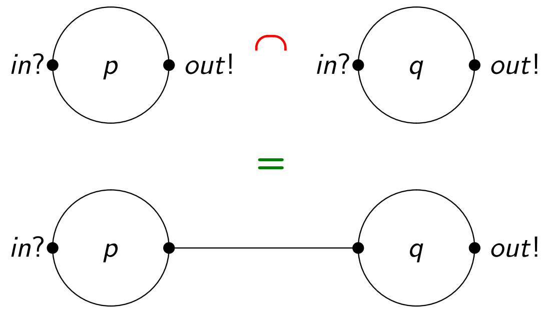
- Example: Maximum of two inputs

$$\begin{aligned} & in?x \to (in?y \to \\ & \quad (\ \ x \le y \to max!y \\ & \quad + \\ & \quad\ \ y \le x \to max!x \quad)) \end{aligned}$$



Linking processes

Connect p's output port to q's input port:



Definition:

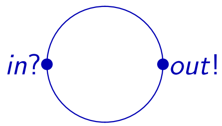
$$p \frown q = (p[c/out] \parallel q[c/in]) \setminus c$$

where c is a fresh channel name

Buffers

- **Definition:**

$B \stackrel{\text{def}}{=} in?x \to (out!x \to B)$



- n-ary buffer

$$\underbrace{B \frown B \frown \cdots \frown B}_{n \text{ times}}$$

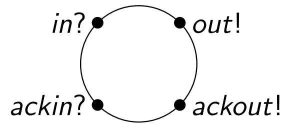
- Exercise: Draw the transition system for $B \frown B$

Remember: $p \frown q = (p[c/out] \parallel q[c/in]) \setminus c$

Buffer with acknowledgements

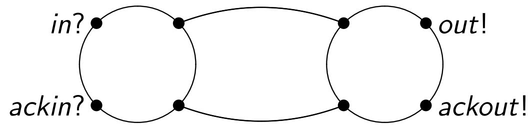
- **Definition:**

$$D \stackrel{\text{def}}{=} in?x \to out!x \to ackout? \to ackin! \to D$$



- Chaining now establishes two links:

$D \Cap D$

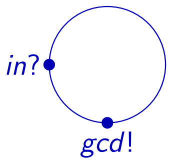


- How would this differ from the following process?

$$D' \stackrel{\text{def}}{=} in?x \to ackin! \to out!x \to ackout? \to D'$$

Euclid's algorithm in CCS

Interface:



Implementation:

$$\begin{array}{lll} E(x,y) & \stackrel{\text{def}}{=} & \; x = y \to gcd!x \to \textbf{nil} \\ & & + \; x < y \to E(x, y-x) \\ & & + \; y < x \to E(x-y, x) \end{array}$$

$$Euclid \stackrel{\text{def}}{=} in?x \to in?y \to E(x,y)$$

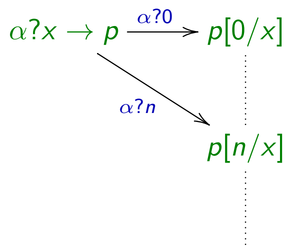
Euclid's algorithm in CCS (without parameterized processes)

$$\begin{array}{lll} Step & \stackrel{\text{def}}{=} & in?x \to \\ & & in?y \to \\ & & \quad (x = y \to gcd!x \to \textbf{nil}) \\ & & \; + \\ & & \quad (x < y \to out!x \to out!(y-x) \to \textbf{nil}) \\ & & \; + \\ & & \quad (y < x \to out!(x-y) \to out!y \to \textbf{nil}) \end{array}$$

$$Euclid \stackrel{\text{def}}{=} Step \frown Euclid$$

Towards a more basic language

- Transitions for value passing carry labels τ, $a?n$, $a!n$



- This suggests introducing prefix $\alpha?n.p$ (as well as $\alpha!n.p$) and view $\alpha?x \to p$ as a sum $\sum_n \alpha?n.p[n/x]$ ← infinite sum
- View $\alpha?n$ and $\alpha!n$ as complementary actions
- Synchronization can only occur on complementary actions

Pure CCS

- Actions: a, b, c, ...
- Complementary actions: $\overline{a}$, $\overline{b}$, $\overline{c}$, ...
- Internal action: τ
- Notational convention: $\overline{\overline{a}} = a$
- Processes:

p	$::=$	$\lambda.p$	prefix	λ ranges over τ, a, $\overline{a}$ for any action label a
	$\mid$	$\sum_{i \in I} p_i$	sum	I is an indexing set
	$\mid$	$p_0 \parallel p_1$	parallel	
	$\mid$	$p \backslash L$	restriction	L a set of channel identifiers
	$\mid$	$p[f]$	relabelling	f a function on channel identifiers
	$\mid$	P		process identifier

- Process definitions:

$$P \stackrel{\text{def}}{=} p$$

Transition rules for Pure CCS

- **Nil process** no rules
- **Guarded processes**

$$\lambda.p \xrightarrow{\lambda} p$$

- **Sum**

$$\frac{p_j \xrightarrow{\lambda} p' \qquad j \in I}{\sum_{i \in I} p_i \to \lambda p'_0}$$

- **Parallel composition**

$$\frac{p_0 \xrightarrow{\lambda} p'_0}{p_0 \parallel p_1 \xrightarrow{\lambda} p'_0 \parallel p_1} \qquad \frac{p_1 \xrightarrow{\lambda} p'_1}{p_0 \parallel p_1 \xrightarrow{\lambda} p_0 \parallel p'_1}$$

$$\frac{p_0 \xrightarrow{a} p'_0 \qquad p_1 \xrightarrow{\overline{a}} p'_1}{p_0 \parallel p_1 \xrightarrow{\tau} p'_0 \parallel p'_1}$$

- **Restriction**

$$\frac{p \xrightarrow{\lambda} p' \qquad \lambda \notin L \cup \overline{L}}{p \setminus L \xrightarrow{\lambda} p' \setminus L} \qquad \text{where } \overline{L} = \{\overline{a} \mid a \in L\}$$

- **Relabelling**

$$\frac{p \xrightarrow{\lambda} p'}{p[f] \xrightarrow{f(\lambda)} p'[f]}$$

where f is a function such that $f(\tau) = \tau$ and $f(\overline{a}) = \overline{f(a)}$

- **Identifiers**

$$\frac{p \xrightarrow{\lambda} p' \qquad P \stackrel{\text{def}}{=} p}{P \xrightarrow{\lambda} p'}$$