Australian National University

# COMP3610/6361
# Principles of Programming Languages

Peter Höfner

Sep 27, 2023

# Section 20

## The Process Algebra CCS

# Towards an Abstract Mechanism for Concurrency

**The Calculus of Communicating Systems (CCS)**

- introduced by Robin Milner in 1980
- first process calculus developed with its operational semantics
- supports algebraic reasoning about equivalence
- simplifies Dijkstra's GCL by removing the store

# Actions and Communications

- processes communicate values (numbers) on channels
- communication is synchronous and between two processes
- $a$ is an arithmetic expression; evaluation is written $a \rightarrow n$
- input: $\alpha?x$
- output $\alpha!a$
- *silent* actions $\tau$ (internal to a process)
- $\lambda$ will range over all the kinds of actions, including $\tau$

# (Decorated) CCS – Syntax

**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 |
| $\mid\ p + p$ | nondeterministic choice |
| $\mid\ p \parallel p$ | parallel composition |
| $\mid\ p\backslash L$ | restriction ($L$ a set of channel identifiers) |
| $\mid\ p[f]$ | relabelling ($f$ a function on channel identifiers) |
| $\mid\ P(a_1, \ldots, a_k)$ | process identifier |

# (Decorated) CCS – Syntax

## Process Definitions:

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

(free variables of $p \subseteq \{x_1, \ldots, x_k\}$)

# Restriction and Relabelling – Examples

- $p\backslash L$: disallow *external* interaction on channels in $L$
- $p[f]$: rename *external* interface to channels by $f$

# Operational semantics of CCS
## Guarded processes

silent action

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

output

$$\frac{a \longrightarrow n}{(\alpha!a \to p) \xrightarrow{\alpha!n} p}$$

input

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

Boolean

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

# Operational semantics of CCS

## 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

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

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

# Operational semantics of CCS

## Restriction

$$\frac{p \xrightarrow{\lambda} p'}{p \backslash L \xrightarrow{\lambda} p' \backslash L} \quad \text{if } \lambda \in \{\alpha?n, \alpha!n\} \text{ then } \alpha \notin L$$

## Relabelling

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

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

## Identifiers

$$\frac{p[a_1/x_1, \ldots, a_n/x_n] \xrightarrow{\lambda} p'}{P(a_1, \ldots, a_n) \xrightarrow{\lambda} p'} \; P(x_1, \ldots, x_n) \stackrel{\text{def}}{=} p$$

## Nil process

no rules

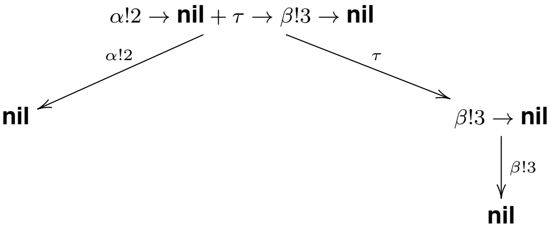
Australian National University

# A Derivation

$$(((\alpha!3 \rightarrow \textbf{nil} + P) \parallel \tau \rightarrow \textbf{nil}) \parallel \alpha?x \rightarrow \textbf{nil})\backslash\{\alpha\} \xrightarrow{\tau} ((\textbf{nil} \parallel \tau \rightarrow \textbf{nil}) \parallel \textbf{nil})\backslash\{\alpha\}$$

# More Examples

- Mixed choice



# Linking Process

(some syntactic sugar)

Let

$$P \stackrel{\text{def}}{=} in?x \rightarrow out!x \rightarrow P$$

$$Q \stackrel{\text{def}}{=} in?y \rightarrow out!y \rightarrow Q$$

Connect $P$'s output port to $Q$'s input port

$$P \frown Q = (P[c/out] \parallel Q[c/in]) \backslash \{c\}$$

where $c$ is a *fresh* channel name

# Euclid's algorithm in CSS

$$
\begin{aligned}
E(x,y) \stackrel{\text{def}}{=} \quad & x = y \rightarrow \mathrm{gcd}!x \rightarrow \textbf{nil} \\
& + x < y \rightarrow E(x, y - x) \\
& + y < x \rightarrow E(x - y, x)
\end{aligned}
$$

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