# Semantics and Verification

Lecture 2

- informal introduction to CCS
- syntax of CCS
- semantics of CCS

# CCS Basics (Sequential Fragment)

- *Nil* (or 0) process (the only atomic process)
- action prefixing ($a.P$)
- names and recursive definitions ($\stackrel{\text{def}}{=}$)
- nondeterministic choice ($+$)

**This is Enough to Describe Sequential Processes**

Any finite LTS can be (up to isomorphism) described by using the operations above.

# CCS Basics (Parallelism and Renaming)

- parallel composition ($|$)
  (synchronous communication between two components = handshake synchronization)
- restriction ($P \smallsetminus L$)
- relabelling ($P[f]$)

# Definition of CCS (channels, actions, process names)

Let

- $\mathcal{A}$ be a set of channel names (e.g. *tea*, *coffee* are channel names)

- $\mathcal{L} = \mathcal{A} \cup \overline{\mathcal{A}}$ be a set of labels where
  - $\overline{\mathcal{A}} = \{\overline{a} \mid a \in \mathcal{A}\}$
    ($\mathcal{A}$ are called names and $\overline{\mathcal{A}}$ are called co-names)
  - by convention $\overline{\overline{a}} = a$

- $Act = \mathcal{L} \cup \{\tau\}$ is the set of actions where
  - $\tau$ is the internal or silent action

  (e.g. $\tau$, *tea*, $\overline{coffee}$ are actions)

- $\mathcal{K}$ is a set of process names (constants) (e.g. CM).

# Definition of CCS (expressions)

| | | | |
|---|---|---|---|
| $P :=$ | $K$ | $\mid$ | process constants ($K \in \mathcal{K}$) |
| | $\alpha.P$ | $\mid$ | prefixing ($\alpha \in Act$) |
| | $\sum_{i \in I} P_i$ | $\mid$ | summation ($I$ is an arbitrary index set) |
| | $P_1 \vert P_2$ | $\mid$ | parallel composition |
| | $P \smallsetminus L$ | $\mid$ | restriction ($L \subseteq \mathcal{A}$) |
| | $P[f]$ | $\mid$ | relabelling ($f : Act \to Act$) such that <br>• $f(\tau) = \tau$ <br>• $f(\overline{a}) = \overline{f(a)}$ |

The set of all terms generated by the abstract syntax is called CCS process expressions (and denoted by $\mathcal{P}$).

## Notation

$$P_1 + P_2 = \sum_{i \in \{1,2\}} P_i \qquad Nil = 0 = \sum_{i \in \emptyset} P_i$$

# Precedence

## Precedence

1. restriction and relabelling (tightest binding)
2. action prefixing
3. parallel composition
4. summation

Example: $R + a.P|b.Q \setminus L$ means $R + \big((a.P)|(b.(Q \setminus L))\big)$.

# Definition of CCS (defining equations)

## CCS program

A collection of defining equations of the form

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

where $K \in \mathcal{K}$ is a process constant and $P \in \mathcal{P}$ is a CCS process expression.

- Only one defining equation per process constant.
- Recursion is allowed: e.g. $A \stackrel{\text{def}}{=} \overline{a}.A \mid A$.

# Semantics of CCS

| Syntax | | Semantics |
|---|---|---|
| CCS<br>(collection of defining equations) | $\longrightarrow$ | LTS<br>(labelled transition systems) |

HOW?

# Structural Operational Semantics for CCS

### Structural Operational Semantics (SOS) – G. Plotkin 1981

Small-step operational semantics where the behaviour of a system is inferred using syntax driven rules.

Given a collection of CCS defining equations, we define the following LTS $(Proc, Act, \{\xrightarrow{a} \mid a \in Act\})$:

- $Proc = \mathcal{P}$ (the set of all CCS process expressions)
- $Act = \mathcal{L} \cup \{\tau\}$ (the set of all CCS actions including $\tau$)
- transition relation is given by SOS rules of the form:

$$\text{RULE} \quad \frac{\textit{premises}}{\textit{conclusion}} \quad \textit{conditions}$$

# SOS rules for CCS ($\alpha \in Act$, $a \in \mathcal{L}$)

ACT $\dfrac{}{\alpha.P \xrightarrow{\alpha} P}$

SUM$_j$ $\dfrac{P_j \xrightarrow{\alpha} P'_j}{\sum_{i \in I} P_i \xrightarrow{\alpha} P'_j}$ $j \in I$

COM1 $\dfrac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q}$

COM2 $\dfrac{Q \xrightarrow{\alpha} Q'}{P|Q \xrightarrow{\alpha} P|Q'}$

COM3 $\dfrac{P \xrightarrow{a} P' \quad Q \xrightarrow{\overline{a}} Q'}{P|Q \xrightarrow{\tau} P'|Q'}$

RES $\dfrac{P \xrightarrow{\alpha} P'}{P \setminus L \xrightarrow{\alpha} P' \setminus L}$ $\alpha, \overline{\alpha} \notin L$

REL $\dfrac{P \xrightarrow{\alpha} P'}{P[f] \xrightarrow{f(\alpha)} P'[f]}$

CON $\dfrac{P \xrightarrow{\alpha} P'}{K \xrightarrow{\alpha} P'}$ $K \stackrel{\mathrm{def}}{=} P$

# Deriving Transitions in CCS

Let $A \stackrel{\mathrm{def}}{=} a.A$. Then

$$\big((A \mid \overline{a}.Nil) \mid b.Nil\big)[c/a] \xrightarrow{c} \big((A \mid \overline{a}.Nil) \mid b.Nil\big)[c/a].$$

$$\text{REL}\ \dfrac{\text{COM1}\ \dfrac{\text{COM1}\ \dfrac{\text{CON}\ \dfrac{\text{ACT}\ \dfrac{}{a.A \xrightarrow{a} A}}{A \xrightarrow{a} A}\ A \stackrel{\mathrm{def}}{=} a.A}{A \mid \overline{a}.Nil \xrightarrow{a} A \mid \overline{a}.Nil}}{(A \mid \overline{a}.Nil) \mid b.Nil \xrightarrow{a} (A \mid \overline{a}.Nil) \mid b.Nil}}{\big((A \mid \overline{a}.Nil) \mid b.Nil\big)[c/a] \xrightarrow{c} \big((A \mid \overline{a}.Nil) \mid b.Nil\big)[c/a]}$$

# LTS of the Process $a.Nil \mid \overline{a}.Nil$

$$
\begin{tikzcd}
& a.Nil \mid \overline{a}.Nil \arrow[dl, "a"'] \arrow[dd, "\tau"] \arrow[dr, "\overline{a}"] & \\
Nil \mid \overline{a}.Nil \arrow[dr, "\overline{a}"] & & a.Nil \mid Nil \arrow[dl, "a"'] \\
& Nil \mid Nil &
\end{tikzcd}
$$