

CCS Basics (Sequential Fragment)

CCS Basics (Parallelism and Renaming)

Semantics and Verification 2006

Lecture 2

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

- 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.

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

Definition of CCS (channels, actions, process names)

Definition of CCS (expressions)

Precedence

Let

- \mathcal{A} be a set of **channel names** (e.g. *tea*, *coffee* are channel names)
- $\mathcal{L} = \mathcal{A} \cup \bar{\mathcal{A}}$ be a set of **labels** where
 - $\bar{\mathcal{A}} = \{\bar{a} \mid a \in \mathcal{A}\}$
 - (\mathcal{A} are called names and $\bar{\mathcal{A}}$ are called co-names)
 - by convention $\bar{\bar{a}} = a$
- $Act = \mathcal{L} \cup \{\tau\}$ is the set of **actions** where
 - τ is the **internal** or **silent** action
 - (e.g. τ , *tea*, *coffee* are actions)
- \mathcal{K} is a set of **process names (constants)** (e.g. CM).

| | | |
|--------|----------------------|--|
| $P :=$ | K | process constants ($K \in \mathcal{K}$) |
| | $\alpha.P$ | prefixing ($\alpha \in Act$) |
| | $\sum_{i \in I} P_i$ | summation (I is an arbitrary index set) |
| | $P_1 P_2$ | parallel composition |
| | $P \setminus L$ | restriction ($L \subseteq \mathcal{A}$) |
| | $P[f]$ | relabelling ($f : Act \rightarrow Act$) such that <ul style="list-style-type: none"> • $f(\tau) = \tau$ • $f(\bar{a}) = \overline{f(a)}$ |

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

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

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

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. $A \stackrel{\text{def}}{=} \bar{a}.A \mid A$.

Semantics of CCS

Syntax

CCS
(collection of defining equations)



Semantics

LTS
(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 τ)
- transition relation is given by **SOS rules** of the form:

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

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

$$\text{ACT } \frac{}{\alpha.P \xrightarrow{\alpha} P}$$

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

$$\text{COM1 } \frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q}$$

$$\text{COM2 } \frac{Q \xrightarrow{\alpha} Q'}{P|Q \xrightarrow{\alpha} P|Q'}$$

$$\text{COM3 } \frac{P \xrightarrow{a} P' \quad Q \xrightarrow{\bar{a}} Q'}{P|Q \xrightarrow{\tau} P'|Q'}$$

$$\text{RES } \frac{P \xrightarrow{\alpha} P'}{P \setminus L \xrightarrow{\alpha} P' \setminus L} \quad \alpha, \bar{\alpha} \notin L$$

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

$$\text{CON } \frac{P \xrightarrow{\alpha} P'}{K \xrightarrow{\alpha} P'} \quad K \stackrel{\text{def}}{=} P$$

Deriving Transitions in CCS

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

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

$$\begin{array}{c} \text{ACT } \frac{}{a.A \xrightarrow{a} A} \\ \text{CON } \frac{a.A \xrightarrow{a} A}{A \xrightarrow{a} A} \quad A \stackrel{\text{def}}{=} a.A \\ \text{COM1 } \frac{A \mid \bar{a}.Nil \xrightarrow{a} A \mid \bar{a}.Nil}{(A \mid \bar{a}.Nil) \mid b.Nil \xrightarrow{a} (A \mid \bar{a}.Nil) \mid b.Nil} \\ \text{REL } \frac{\text{COM1}}{((A \mid \bar{a}.Nil) \mid b.Nil)[c/a] \xrightarrow{c} ((A \mid \bar{a}.Nil) \mid b.Nil)[c/a]} \end{array}$$

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

