Semantics and Verification 2008

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
  - τ is the **internal** or **silent** action

  (e.g. τ, *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)
	$P_1 \mid P_2$	$\mid$	parallel composition
	$P \smallsetminus L$	$\mid$	restriction ($L \subseteq \mathcal{A}$)
	$P[f]$	$\mid$	relabelling ($f : Act \to Act$) such that • $f(\tau) = \tau$ • $f(\overline{a}) = \overline{f(a)}$

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

Notation

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

Precedence

Precedence

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

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

Definition of CCS (defining equations)

CCS program

A collection of **defining equations** of the form

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

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. $A \stackrel{\mathrm{def}}{=} \overline{a}.A \mid A$.

Semantics of CCS

Syntax		Semantics
CCS (collection of defining equations)	$\longrightarrow$	**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 $(Proc, Act, \{\stackrel{a}{\longrightarrow} \mid a \in Act\})$:

- $Proc = \mathcal{P}$ (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:

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

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

$$\text{ACT} \quad \frac{}{\alpha.P \stackrel{\alpha}{\longrightarrow} P} \qquad \text{SUM}_j \quad \frac{P_j \stackrel{\alpha}{\longrightarrow} P_j'}{\sum_{i \in I} P_i \stackrel{\alpha}{\longrightarrow} P_j'} \quad j \in I$$

$$\text{COM1} \quad \frac{P \stackrel{\alpha}{\longrightarrow} P'}{P|Q \stackrel{\alpha}{\longrightarrow} P'|Q} \qquad \text{COM2} \quad \frac{Q \stackrel{\alpha}{\longrightarrow} Q'}{P|Q \stackrel{\alpha}{\longrightarrow} P|Q'}$$

$$\text{COM3} \quad \frac{P \stackrel{a}{\longrightarrow} P' \quad Q \stackrel{\overline{a}}{\longrightarrow} Q'}{P|Q \stackrel{\tau}{\longrightarrow} P'|Q'}$$

$$\text{RES} \quad \frac{P \stackrel{\alpha}{\longrightarrow} P'}{P \smallsetminus L \stackrel{\alpha}{\longrightarrow} P' \smallsetminus L} \quad \alpha, \overline{\alpha} \notin L \qquad \text{REL} \quad \frac{P \stackrel{\alpha}{\longrightarrow} P'}{P[f] \stackrel{f(\alpha)}{\longrightarrow} P'[f]}$$

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

Deriving Transitions in CCS

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

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

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

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