Semantics and Verification 2005

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 ($\stackrel{\mathrm{def}}{=}$)
- 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 $(P[f])$

Introduction to CCS
Syntax of CCS
Semantics of CCS

Notation
CCS Process Expressions
CCS Defining Equations

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
  - $\overline{\mathcal{A}} = \{\bar{a} \mid a \in \mathcal{A}\}$
    ($\mathcal{A}$ are called names and $\overline{\mathcal{A}}$ 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*, $\overline{coffee}$ are actions)

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

Introduction to CCS
Syntax of CCS
Semantics of CCS

Notation
CCS Process Expressions
CCS Defining Equations

Definition of CCS (expressions)

$$P := \quad K \qquad\qquad | \qquad\qquad \text{process constants } (K \in \mathcal{K})$$
$$\alpha.P \qquad | \qquad \text{prefixing } (\alpha \in Act)$$
$$\textstyle\sum_{i \in I} P_i \quad | \qquad \text{summation } (I \text{ is an arbitrary index set})$$
$$P_1 | P_2 \qquad | \qquad \text{parallel composition}$$
$$P \smallsetminus L \qquad | \qquad \text{restriction } (L \subseteq \mathcal{A})$$
$$P[f] \qquad | \qquad \text{relabelling } (f : Act \to Act) \text{ such that}$$

- $f(\tau) = \tau$
- $f(\bar{a}) = \overline{f(a)}$

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

Notation

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

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

Introduction to CCS
Syntax of CCS
Semantics of CCS

Notation
CCS Process Expressions
CCS Defining Equations

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

Introduction to CCS
Syntax of CCS
Semantics of CCS

Motivation
SOS Rules for CCS
Examples

Semantics of CCS

Syntax
CCS
(collection of defining equations)

$\longrightarrow$

Semantics
LTS
(labelled transition systems)

HOW?

Introduction to CCS
Syntax of CCS
Semantics of CCS

Motivation
SOS Rules for CCS
Examples

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:

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

Introduction to CCS
Syntax of CCS
Semantics of CCS

Motivation
SOS Rules for CCS
Examples

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

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

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

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

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

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

Introduction to CCS
Syntax of CCS
Semantics of CCS

Motivation
SOS Rules for CCS
Examples

Deriving Transitions in CCS

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

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

$$
\text{REL} \; \cfrac{\text{COM1} \; \cfrac{\text{COM1} \; \cfrac{\text{CON} \cfrac{\text{ACT} \; \cfrac{}{a.A \stackrel{a}{\longrightarrow} A}}{A \stackrel{a}{\longrightarrow} A} A \stackrel{\text{def}}{=} a.A}{A \,|\, \overline{a}.Nil \stackrel{a}{\longrightarrow} A \,|\, \overline{a}.Nil}}{(A \,|\, \overline{a}.Nil) \,|\, b.Nil \stackrel{a}{\longrightarrow} (A \,|\, \overline{a}.Nil) \,|\, b.Nil}}{((A \,|\, \overline{a}.Nil) \,|\, b.Nil)[c/a] \stackrel{c}{\longrightarrow} ((A \,|\, \overline{a}.Nil) \,|\, b.Nil)[c/a]}
$$

Introduction to CCS
Syntax of CCS
Semantics of CCS

Motivation
SOS Rules for CCS
Examples

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