# Semantics and Verification

Lecture 2

9 February 2010

# Overview

Last lecture:

- Binary relations
- Labeled transition systems
- CCS informally

This lecture:

- CCS: syntax
- semantics of CCS (as transition systems)

Next lecture:

- Behavioral equivalences

1. Introduction
2. Syntax
3. Semantics
4. Example

# CCS Basics (Sequential Fragment)

- *Nil* (or 0) process (the only atomic process)
- action prefixing $a.P$
- names and recursive definitions via $\stackrel{\mathrm{def}}{=}$
- 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 $P \setminus L$
- 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) |
| | $P_1 \mid P_2$ | $\mid$ | parallel composition |
| | $P \setminus L$ | $\mid$ | 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: $R + a.P \mid b.Q \smallsetminus L$ means $R + \big((a.P) \mid (b.(Q \smallsetminus L))\big)$.

# 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: Motivation

**Syntax**

CCS
(collection of defining equations)

$\longrightarrow$

**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 $\tau$)
- transition relation is given by SOS rules of the form:

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

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

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

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

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

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

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

# Deriving Transitions in CCS

Let $A \stackrel{\text{def}}{=} a.A$. Let's check whether

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

# Deriving Transitions in CCS

Let $A \stackrel{\text{def}}{=} a.A$. Let's check whether

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

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

# Deriving Transitions in CCS

Let $A \stackrel{\text{def}}{=} a.A$. Let's check whether

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

$$\text{REL}\ \dfrac{\text{COM1}\ \dfrac{}{(A \mid \overline{a}.Nil) \mid b.Nil \xrightarrow{a}}}{\big((A \mid \overline{a}.Nil) \mid b.Nil\big)[c/a] \xrightarrow{c}}$$

Let $A \stackrel{\mathrm{def}}{=} a.A$. Let's check whether

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

$$\text{REL}\ \dfrac{\text{COM1}\ \dfrac{\text{COM1}\ \dfrac{}{A \mid \overline{a}.Nil \xrightarrow{a}}}{(A \mid \overline{a}.Nil) \mid b.Nil \xrightarrow{a}}}{\big((A \mid \overline{a}.Nil) \mid b.Nil\big)[c/a] \xrightarrow{c}}$$

# Deriving Transitions in CCS

Let $A \stackrel{\mathrm{def}}{=} a.A$. Let's check whether

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

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

# Deriving Transitions in CCS

Let $A \stackrel{\text{def}}{=} a.A$. Let's check whether

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

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

# Deriving Transitions in CCS

Let $A \stackrel{\text{def}}{=} a.A$. Let's check whether

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

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

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