Introducing CCS

CCS processes

Given a set of **action names**, the set of **CCS processes** $\mathcal{P}$ is defined by the grammar

$P ::=$	0	empty process
	$\alpha.P$	α is an **action**,
		defined using action names
	A	equation $A \stackrel{\text{def}}{=} P$,
		A process identifier
	$P + Q$	choice, either P or Q
	$P \mid Q$	parallel composition
	$(\text{new } a)\ P$	restriction, a action name,
		a bound in P
	$P[b/a]$	renaming, the action name b
		replaces action name a in P.

Eventually we will generalise some of the processes.

First Example

A clock that perpetually ticks.

$$\mathtt{Cl} \stackrel{\mathrm{def}}{=} \mathtt{tick}.\mathtt{Cl}$$

- `tick` action
- `Cl` process identifier
- $\stackrel{\mathrm{def}}{=}$ ties a process identifier to a process
- `tick.Cl` process
- . prefix operator

Behaviour: transitions

The behaviour of a process is determined by its **actions**. It is captured by **transitions** of the form

transition $P \xrightarrow{\alpha} Q$, α action.

Some transition rules for deriving transitions:

- axiom for prefix

$$\text{Rule}(.) \qquad \alpha.P \xrightarrow{\alpha} P$$

- rule for equations

$$\text{Rule}(\stackrel{\text{def}}{=}) \qquad \frac{P \xrightarrow{\alpha} Q}{A \xrightarrow{\alpha} Q} \qquad A \stackrel{\text{def}}{=} P$$

The empty process 0 has no transitions.

Warning! Milner writes the transition rules this way up. However, Stirling writes the rules the other way around, with $A \xrightarrow{\alpha} Q$ at the top. **Very confusing!**

Example

$$\frac{\texttt{tick.Cl} \xrightarrow{\texttt{tick}} \texttt{Cl}}{\texttt{Cl} \xrightarrow{\texttt{tick}} \texttt{Cl}} \qquad \texttt{Cl} \stackrel{\text{def}}{=} \texttt{tick.Cl}$$

Behaviour: transition graphs

Graphical representation

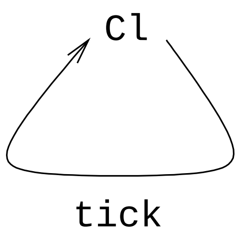


Figure 1: The transition graph for `Cl`

The vertices are processes.

The labelled edges are transitions.

Exercise

1. Draw the transition graphs for the following:

   (a) $\alpha.0$

   (b) $\mathtt{Cl}_1 \stackrel{\text{def}}{=} \mathtt{tick.tock.Cl}_1$

   (c) $\mathtt{Cl}_2 \stackrel{\text{def}}{=} \mathtt{tick.tick.Cl}_2$

   (d) $\mathtt{Cl}_3 \stackrel{\text{def}}{=} \mathtt{tick.Cl}$ where $\mathtt{Cl} \stackrel{\text{def}}{=} \mathtt{tick.Cl}$

The $+$ operator

$$\texttt{Ven} \stackrel{\text{def}}{=} \texttt{2p}.\texttt{Ven}_\texttt{b} + \texttt{1p}.\texttt{Ven}_\texttt{l}$$

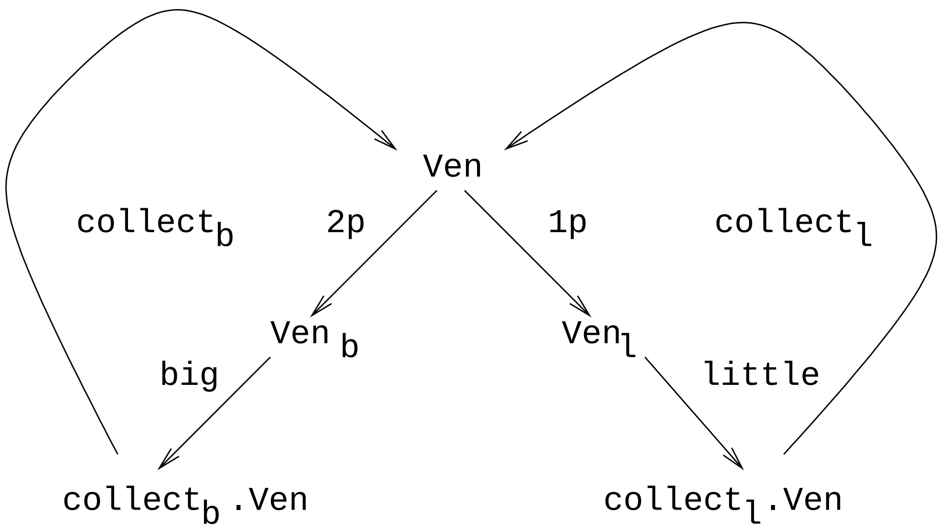
$$\texttt{Ven}_\texttt{b} \stackrel{\text{def}}{=} \texttt{big}.\texttt{collect}_\texttt{b}.\texttt{Ven}$$

$$\texttt{Ven}_\texttt{l} \stackrel{\text{def}}{=} \texttt{little}.\texttt{collect}_\texttt{l}.\texttt{Ven}$$

Transition rules for choice

$$\text{Rules}(+) \quad \frac{P_1 \xrightarrow{\alpha} Q}{P_1 + P_2 \xrightarrow{\alpha} Q} \qquad \frac{P_2 \xrightarrow{\alpha} Q}{P_1 + P_2 \xrightarrow{\alpha} Q}$$

Transition Graph



Process Composition

Transition Rules

$$\text{Rules}(|)\quad \frac{P \xrightarrow{\alpha} P'}{P \,|\, Q \xrightarrow{\alpha} P' \,|\, Q} \qquad \frac{Q \xrightarrow{\alpha} Q'}{P \,|\, Q \xrightarrow{\alpha} P \,|\, Q'}$$

One more to come...

Example

$$\begin{aligned} \mathtt{Ct}_0 &\stackrel{\text{def}}{=} \mathtt{up}.\mathtt{Ct}_1 \\ \mathtt{Ct}_{\mathtt{i}+1} &\stackrel{\text{def}}{=} \mathtt{up}.\mathtt{Ct}_{\mathtt{i}+2} + \mathtt{down}.\mathtt{Ct}_{\mathtt{i}} \end{aligned}$$

$$\frac{\dfrac{\mathtt{up}.\mathtt{Ct}_4 \xrightarrow{\mathtt{up}} \mathtt{Ct}_4}{\mathtt{up}.\mathtt{Ct}_4 + \mathtt{down}.\mathtt{Ct}_2 \xrightarrow{\mathtt{up}} \mathtt{Ct}_4}}{\mathtt{Ct}_3 \xrightarrow{\mathtt{up}} \mathtt{Ct}_4}$$

Draw the transition graph.

Now consider $\mathtt{Cnt} \stackrel{\text{def}}{=} \mathtt{up}.(\mathtt{Cnt} \mid \mathtt{down}.0)$

Draw the transition graph. It's huge!

Behaviour

For arbitrary P, Q, R, we will prove that the following have the same behaviour:

- P and $P \mid 0$
- $P \mid Q$ and $Q \mid P$
- $(P \mid Q) \mid R$ and $P \mid (Q \mid R)$
- $\mathtt{Cnt}$ and $\mathtt{Ct}_0$.

Notation

Write $P \mid Q \mid R$ for either $(P \mid Q) \mid R$ or $P \mid (Q \mid R)$

CCS Actions

The other transition rule for composition depends on the particular actions of CCS.

Given a set of action names A, the CCS actions are defined by

CCS actions $\qquad \alpha ::= a \;\big|\; \overline{a} \;\big|\; \tau, \quad a \in A$

a is called an **input** action

$\overline{a}$ is called an **output** action

τ is called the **silent** (internal) action.

Given process $\alpha.P$, an action name in α is free in $\alpha.P$.

Process Composition Continued

Transition Rules for τ

$$\frac{P \xrightarrow{a} P' \qquad Q \xrightarrow{\overline{a}} Q'}{P \,|\, Q \xrightarrow{\tau} P' \,|\, Q'}$$

$$\frac{P \xrightarrow{\overline{a}} P' \qquad Q \xrightarrow{a} Q'}{P \,|\, Q \xrightarrow{\tau} P' \,|\, Q'}$$

These rules intuitively describes a handshake between processes P and Q.

Example

Informal Description

Raw articles arrive on a conveyor belt. They are then processed using a tool, and the finished products placed on another conveyor belt. There are two workers, but only one tool, which means that only one worker can operate at once.

Define a process which describes this system, using action $\texttt{in}$ to model articles arriving and action $\overline{\texttt{out}}$ to model the finished products leaving.

It should **mimic** the specification process

$$\texttt{Spec} \stackrel{\text{def}}{=} \texttt{in}.\overline{\texttt{out}}.\texttt{Spec}$$

Solution: first try

Model the interaction between a worker and the tool by

$$\texttt{Worker} \mid \texttt{Tool}$$

where

$$\texttt{Tool} \stackrel{\text{def}}{=} \texttt{up.down.Tool}$$
$$\texttt{Worker} \stackrel{\text{def}}{=} \overline{\texttt{up}}.\texttt{in}.\overline{\texttt{out}}.\overline{\texttt{down}}.\texttt{Worker}$$

Draw the transition graph of $\texttt{Worker} \mid \texttt{Tool}$.

(It's a little complex.)

Now add the other worker:

$$\texttt{Worker} \mid \texttt{Worker} \mid \texttt{Tool}$$

Think about its transition graph.

(Don't draw in full as it's too complex.)

This does not have the same behaviour as $\texttt{Spec}$.

Restriction

Transition Rule

$$\mathrm{R}(\texttt{new}) \quad \frac{P \xrightarrow{\alpha} Q}{(\texttt{new}\ a)\, P \xrightarrow{\alpha} (\texttt{new}\ a)\, Q}\ \alpha \neq a, \overline{a}$$

Restriction simplifies the behaviour of a process, in that it restricts the possible transitions preventing further interaction with the environment.

Example continued

Specification

$$\mathtt{Spec} \stackrel{\text{def}}{=} \mathtt{in}.\overline{\mathtt{out}}.\mathtt{Spec}$$

Notation We sometimes write $\mathtt{new}\ a\ (P)$ for $(\mathtt{new}\ a)P$ and $\mathtt{new}\ a, b\ (P)$ for $(\mathtt{new}\ a)(\mathtt{new}\ b)P$ when the meaning is clear.

Solution

$$\mathtt{new}\ \mathtt{up}, \mathtt{down}(\mathtt{Worker} \mid \mathtt{Worker} \mid \mathtt{Tool})$$

where

$$\begin{aligned} \mathtt{Tool} &\stackrel{\text{def}}{=} \mathtt{up}.\mathtt{down}.\mathtt{Tool} \\ \mathtt{Worker} &\stackrel{\text{def}}{=} \overline{\mathtt{up}}.\mathtt{in}.\overline{\mathtt{out}}.\overline{\mathtt{down}}.\mathtt{Worker} \end{aligned}$$

Draw the transition graph. It is simpler!

$\mathtt{new}\ \mathtt{up}, \mathtt{down}\ (\mathtt{Worker} \mid \mathtt{Worker} \mid \mathtt{Tool})$ has the same behaviour as $\mathtt{Spec}$.