Concurrency

Colin Stirling
School of Informatics
University of Edinburgh

April 17, 2007

School of informatics

Content

- Start with looking at a particular process calculus, Milner's CCS (Calculus of Communicating Systems: illustrates structural operational semantics)

- Consider issue of equivalence, especially bisimulation equivalence

- Examine problem of logics and issue of model-checking

- Reflections on other areas of theoretical computer science, finite model theory and formal language theory

First example

A clock that perpetually ticks

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

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

Behaviour: transitions

Behaviour of processes is captured by transitions $E \xrightarrow{a} F$

Axioms and goal directed rules for deriving transitions.

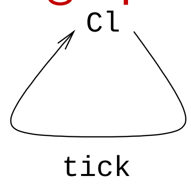
$$\mathrm{R}(.) \quad a.E \xrightarrow{a} E \qquad\qquad \mathrm{R}(\stackrel{\mathrm{def}}{=}) \quad \frac{P \xrightarrow{a} F}{E \xrightarrow{a} F} \; P \stackrel{\mathrm{def}}{=} E$$

Example

$$\mathtt{Cl} \xrightarrow{\mathtt{tick}} \mathtt{Cl}$$

Behaviour: transition graphs

Graphical representation as a labelled graph



- vertices: process expressions
- labelled edges: transitions

Each derivable transition of a vertex is depicted

Abstract from the derivations of transitions

Interlude: exercise

Draw the transition graphs for the following clocks

1. $\mathtt{Cl}_1 \stackrel{\text{def}}{=} \mathtt{tick.tock.Cl}_1$

2. $\mathtt{Cl}_2 \stackrel{\text{def}}{=} \mathtt{tick.tick.Cl}_2$

3. $\mathtt{Cl}_3 \stackrel{\text{def}}{=} \mathtt{tick.Cl}$

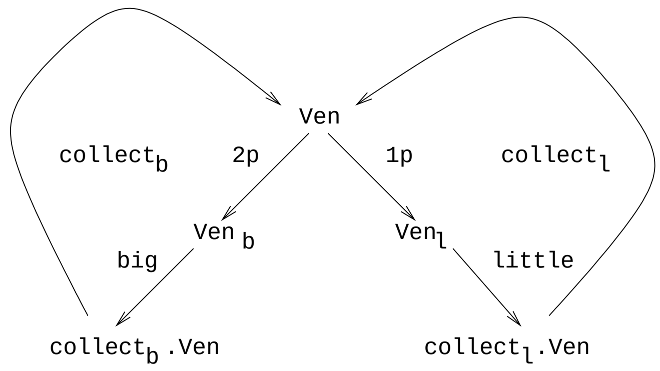
The + operator

$$\begin{aligned}\texttt{Ven} &\stackrel{\text{def}}{=} \texttt{2p}.\texttt{Ven}_\texttt{b} + \texttt{1p}.\texttt{Ven}_\texttt{l}\\ \texttt{Ven}_\texttt{b} &\stackrel{\text{def}}{=} \texttt{big}.\texttt{collect}_\texttt{b}.\texttt{Ven}\\ \texttt{Ven}_\texttt{l} &\stackrel{\text{def}}{=} \texttt{little}.\texttt{collect}_\texttt{l}.\texttt{Ven}\end{aligned}$$

Transition Rule

$$\text{R}(+) \quad \frac{E_1 \xrightarrow{a} F}{E_1 + E_2 \xrightarrow{a} F} \qquad \frac{E_2 \xrightarrow{a} F}{E_1 + E_2 \xrightarrow{a} F}$$

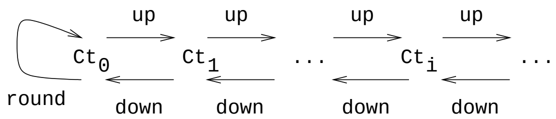
Transition Graph



Generalising I

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

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



Generalising II

$\sum\{E_i : i \in I\}$ I indexing set $E_1 + E_2$ abbreviates $\sum\{E_i : i \in \{1,2\}\}$

$$\mathtt{Reg}'_{\mathtt{i}} \stackrel{\text{def}}{=} \mathtt{read}_{\mathtt{i}}.\mathtt{Reg}'_{\mathtt{i}} + \sum\{\mathtt{write}_j.\mathtt{Reg}'_j : j \in \mathbb{N}\}$$

Transition Rule for Σ

$$\mathrm{R}(\sum) \quad \frac{\sum\{E_i : i \in I\} \xrightarrow{a} F}{E_j \xrightarrow{a} F} \; j \in I$$

Special Case: $\sum\{E_i : i \in \emptyset\}$ abbreviated to 0 “nil”

Generalising III

input of data at port a, $a(x).E$

where $a(x)$ binds free occurrences of x in E. The label a represents the set $\{a(v) : v \in D\}$ where D is appropriate family of data values.

output of data at port a, $\overline{a}(e).E$, e a data expression.

Transition Rules: depends on Val(e) the data value in D, if there is one, to which e evaluates

$$\text{R(in)} \quad a(x).E \xrightarrow{a(v)} E\{v/x\} \quad \text{if} \;\; v \in D$$

where $\{v/x\}$ is substitution

$$\text{R(out)} \quad \overline{a}(e).E \xrightarrow{\overline{a}(v)} E \quad \text{if Val}(e) = v$$

Examples

A Copier: $\mathtt{Cop} \stackrel{\text{def}}{=} \mathtt{in}(x).\overline{\mathtt{out}}(x).\mathtt{Cop}$

$$\frac{\mathtt{Cop} \xrightarrow{\mathtt{in}(v)} \overline{\mathtt{out}}(v).\mathtt{Cop}}{\mathtt{in}(x).\overline{\mathtt{out}}(x).\mathtt{Cop} \xrightarrow{\mathtt{in}(v)} \overline{\mathtt{out}}(v).\mathtt{Cop}}$$

A Register: $\mathtt{Reg}_{\mathtt{i}} \stackrel{\text{def}}{=} \overline{\mathtt{read}}(i).\mathtt{Reg}_{\mathtt{i}} + \mathtt{write}(x).\mathtt{Reg}_x$

$$\frac{\frac{\mathtt{Reg}_5 \xrightarrow{\mathtt{write}(3)} \mathtt{Reg}_3}{\overline{\mathtt{read}}(5).\mathtt{Reg}_5 + \mathtt{write}(x).\mathtt{Reg}_x \xrightarrow{\mathtt{write}(3)} \mathtt{Reg}_3}}{\mathtt{write}(x).\mathtt{Reg}_x \xrightarrow{\mathtt{write}(3)} \mathtt{Reg}_3}$$

Exercise

Assume that the space of values consists of two elements, 0 and 1. Draw transition graphs for the following three copiers

1. $\mathtt{Cop} \stackrel{\text{def}}{=} \mathtt{in}(x).\overline{\mathtt{out}}(x).\mathtt{Cop}$

2. $\mathtt{Cop}_1 \stackrel{\text{def}}{=} \mathtt{in}(x).\mathtt{in}(x).\overline{\mathtt{out}}(x).\mathtt{Cop}_1$

3. $\mathtt{Cop}_2 \stackrel{\text{def}}{=} \mathtt{in}(x).\overline{\mathtt{out}}(x).\overline{\mathtt{out}}(x).\mathtt{Cop}_2$

Concurrent Composition $E \mid F$

$$\mathrm{R}(\mid \mathrm{com}) \quad \frac{E \mid F \xrightarrow{\tau} E' \mid F'}{E \xrightarrow{a} E' \qquad F \xrightarrow{\overline{a}} F'}$$

$$\mathrm{R}(\mid) \quad \frac{E \mid F \xrightarrow{a} E' \mid F}{E \xrightarrow{a} E'} \qquad \frac{E \mid F \xrightarrow{a} E \mid F'}{F \xrightarrow{a} F'}$$

The action $\overline{a}$ is co-action of a: $\overline{\overline{a}} = a$

Co-action of $\mathtt{in}(v)$ is $\overline{\mathtt{in}}(v)$

τ is silent action

Example: user of a copier

$$\begin{array}{lcl} \texttt{Cop} & \stackrel{\text{def}}{=} & \texttt{in}(x).\overline{\texttt{out}}(x).\texttt{Cop} \\ \texttt{User} & \stackrel{\text{def}}{=} & \texttt{write}(x).\texttt{User}_x \\ \texttt{User}_\texttt{v} & \stackrel{\text{def}}{=} & \overline{\texttt{in}}(v).\texttt{User} \end{array}$$

$$\dfrac{\dfrac{\texttt{in}(x).\overline{\texttt{out}}(x).\texttt{Cop} \xrightarrow{\texttt{in}(v)} \overline{\texttt{out}}(v).\texttt{Cop}}{\texttt{Cop} \xrightarrow{\texttt{in}(v)} \overline{\texttt{out}}(v).\texttt{Cop}} \qquad \dfrac{\overline{\texttt{in}}(v).\texttt{User} \xrightarrow{\overline{\texttt{in}}(v)} \texttt{User}}{\texttt{User}_\texttt{v} \xrightarrow{\overline{\texttt{in}}(v)} \texttt{User}}}{\texttt{Cop} \mid \texttt{User}_\texttt{v} \xrightarrow{\tau} \overline{\texttt{out}}(v).\texttt{Cop} \mid \texttt{User}}$$

More Users

$$\mathtt{Cop} \stackrel{\text{def}}{=} \mathtt{in}(x).\overline{\mathtt{out}}(x).\mathtt{Cop}$$
$$\mathtt{User} \stackrel{\text{def}}{=} \mathtt{write}(x).\mathtt{User}_x$$
$$\mathtt{User}_{\mathtt{v}} \stackrel{\text{def}}{=} \overline{\mathtt{in}}(v).\mathtt{User}$$

$$\dfrac{\dfrac{\mathtt{in}(x).\overline{\mathtt{out}}(x).\mathtt{Cop} \xrightarrow{\mathtt{in}(v1)} \overline{\mathtt{out}}(v1).\mathtt{Cop}}{\mathtt{Cop} \xrightarrow{\mathtt{in}(v1)} \overline{\mathtt{out}}(v1).\mathtt{Cop}} \quad \dfrac{\dfrac{\dfrac{\overline{\mathtt{in}}(v1).\mathtt{User} \xrightarrow{\overline{\mathtt{in}}(v1)} \mathtt{User}}{\mathtt{User}_{\mathtt{v1}} \xrightarrow{\overline{\mathtt{in}}(v1)} \mathtt{User}}}{\mathtt{User}_{\mathtt{v1}} \mid \mathtt{User}_{\mathtt{v2}} \xrightarrow{\overline{\mathtt{in}}(v1)} \mathtt{User} \mid \mathtt{User}_{\mathtt{v2}}}}{}}{\mathtt{Cop} \mid (\mathtt{User}_{\mathtt{v1}} \mid \mathtt{User}_{\mathtt{v2}}) \xrightarrow{\tau} \overline{\mathtt{out}}(v1).\mathtt{Cop} \mid (\mathtt{User} \mid \mathtt{User}_{\mathtt{v2}})}$$

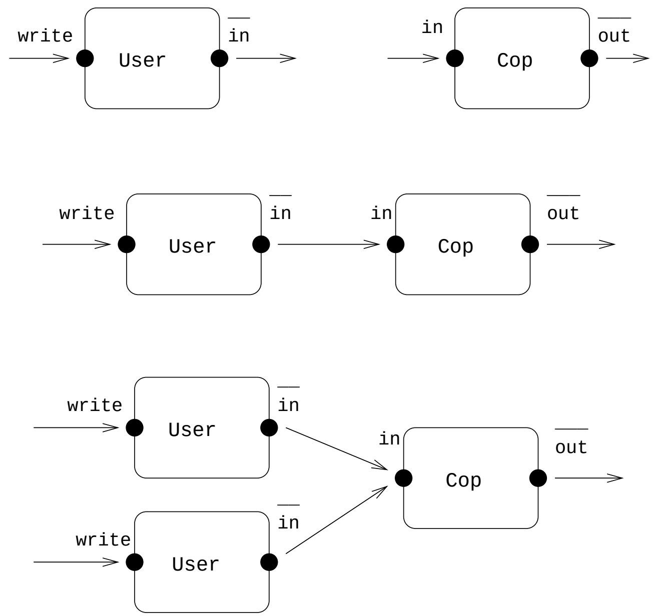
Exercise

Draw the transition graph of Cnt

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

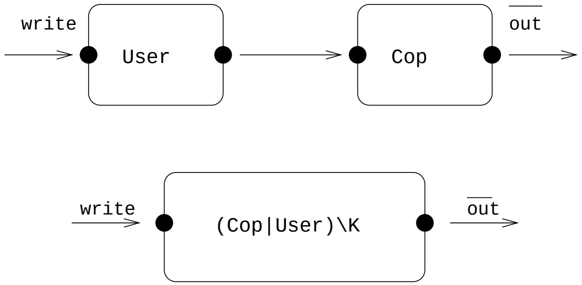
Flow Graphs

Potential movement of information flowing into and out of ports



A private copier?

Operation $\backslash K$ where $K = \{\mathtt{in}(v)\ :\ v \in D\}$ abbreviate to `in`



School of informatics

Transition Rules for $\backslash K$

$\tau \notin K$

$\overline{J}$ is $\{\overline{a} : a \in J\}$

$$\frac{E\backslash J \xrightarrow{a} F\backslash J}{E \xrightarrow{a} F}\; a \notin J \cup \overline{J}$$

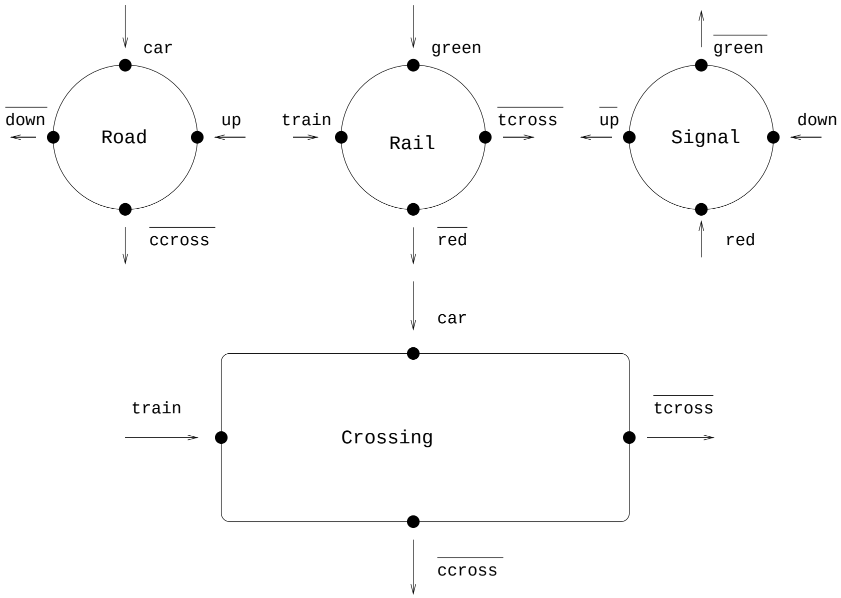
$$\dfrac{\dfrac{\dfrac{\texttt{Cop} \xrightarrow{\texttt{in}(v)} \overline{\texttt{out}}(v).\texttt{Cop}}{\texttt{in}(x).\overline{\texttt{out}}(x).\texttt{Cop} \xrightarrow{\texttt{in}(v)} \overline{\texttt{out}}(v).\texttt{Cop}} \qquad \dfrac{\texttt{User}_{\texttt{v}} \xrightarrow{\overline{\texttt{in}}(v)} \texttt{User}}{\overline{\texttt{in}}(v).\texttt{User} \xrightarrow{\overline{\texttt{in}}(v)} \texttt{User}}}{\texttt{Cop} \mid \texttt{User}_{\texttt{v}} \xrightarrow{\tau} \overline{\texttt{out}}(v).\texttt{Cop} \mid \texttt{User}}}{(\texttt{Cop} \mid \texttt{User}_{\texttt{v}})\backslash K \xrightarrow{\tau} (\overline{\texttt{out}}(v).\texttt{Cop} \mid \texttt{User})\backslash K}$$

Example

Process descriptions can become quite large, especially when they consist of multiple components. $P \equiv F$ means that P abbreviates F.

$$
\begin{array}{lcl}
\texttt{Road} & \stackrel{\text{def}}{=} & \texttt{car}.\texttt{up}.\overline{\texttt{ccross}}.\overline{\texttt{down}}.\texttt{Road} \\
\texttt{Rail} & \stackrel{\text{def}}{=} & \texttt{train}.\texttt{green}.\overline{\texttt{tcross}}.\overline{\texttt{red}}.\texttt{Rail} \\
\texttt{Signal} & \stackrel{\text{def}}{=} & \overline{\texttt{green}}.\texttt{red}.\texttt{Signal} + \overline{\texttt{up}}.\texttt{down}.\texttt{Signal} \\
 & & \\
\texttt{Crossing} & \equiv & (\texttt{Road} \mid \texttt{Rail} \mid \texttt{Signal}) \backslash K \\
 & & \\
K & = & \{\texttt{green}, \texttt{red}, \texttt{up}, \texttt{down}\}
\end{array}
$$

Flow Graphs


car
down
Road
up
ccross
green
train
Rail
tcross
red
green
up
Signal
down
red
car
train
Crossing
tcross
ccross

Protocol that may lose messages

$$\begin{array}{lcl}
\texttt{Sender} & \stackrel{\text{def}}{=} & \texttt{in}(x).\overline{\texttt{sm}}(x).\texttt{Send1}(x) \\
\texttt{Send1}(x) & \stackrel{\text{def}}{=} & \texttt{ms}.\overline{\texttt{sm}}(x).\texttt{Send1}(x) + \texttt{ok}.\texttt{Sender} \\
\texttt{Medium} & \stackrel{\text{def}}{=} & \texttt{sm}(y).\texttt{Med1}(y) \\
\texttt{Med1}(y) & \stackrel{\text{def}}{=} & \overline{\texttt{mr}}(y).\texttt{Medium} + \tau.\overline{\texttt{ms}}.\texttt{Medium} \\
\texttt{Receiver} & \stackrel{\text{def}}{=} & \texttt{mr}(x).\overline{\texttt{out}}(x).\overline{\texttt{ok}}.\texttt{Receiver}
\end{array}$$

$$\texttt{Protocol} \equiv (\texttt{Sender} \mid \texttt{Medium} \mid \texttt{Receiver}) \backslash \{\texttt{sm}, \texttt{ms}, \texttt{mr}, \texttt{ok}\}$$

Abstracting from silent activity

Difference between τ and "observable" actions Assume E may at some time perform `ok`

$$(E \mid \overline{\texttt{ok}}.\texttt{Resource})\backslash\{\texttt{ok}\}$$

Access to `Resource` is triggered by `ok` by E

Observation of `ok` $=$ release of `Resource`

τ cannot be observed in this way

Observable transitions

$E \stackrel{\varepsilon}{\Longrightarrow} F$ or $E \stackrel{a}{\Longrightarrow} F$ where $a \neq \tau$

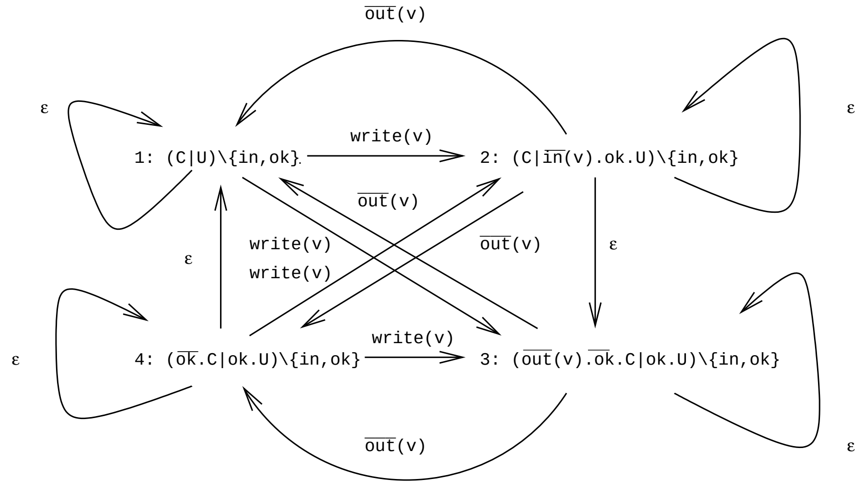
$$\mathrm{R}(\stackrel{\varepsilon}{\Longrightarrow}) \quad E \stackrel{\varepsilon}{\Longrightarrow} E \qquad \frac{E \xrightarrow{\tau} E' \qquad E' \stackrel{\varepsilon}{\Longrightarrow} F}{E \stackrel{\varepsilon}{\Longrightarrow} F}$$

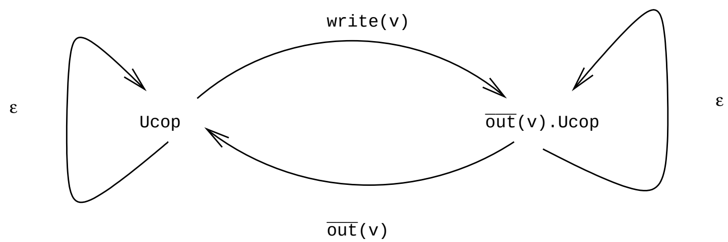
$$\mathrm{R}(\stackrel{a}{\Longrightarrow}) \quad \frac{E \stackrel{\varepsilon}{\Longrightarrow} E' \quad E' \xrightarrow{a} F' \quad F' \stackrel{\varepsilon}{\Longrightarrow} F}{E \stackrel{a}{\Longrightarrow} F}$$

Observable Transition Graphs

$$\begin{aligned}
\mathtt{C} &\stackrel{\mathrm{def}}{=} \mathtt{in}(x).\overline{\mathtt{out}}(x).\overline{\mathtt{ok}}.\mathtt{C} \\
\mathtt{U} &\stackrel{\mathrm{def}}{=} \mathtt{write}(x).\overline{\mathtt{in}}(x).\mathtt{ok}.\mathtt{U} \\
\mathtt{Ucop} &\stackrel{\mathrm{def}}{=} \mathtt{write}(x).\overline{\mathtt{out}}(x).\mathtt{Ucop}
\end{aligned}$$

Observable Transition Graphs





Summary

1. Syntax of CCS: prefix, sum, parallel composition, restriction (but not renaming)
2. Two types of transition, $\xrightarrow{a}$ $\stackrel{a}{\Longrightarrow}$
3. Two types of transition graph that abstracts from derivation of transitions
4. Flow Graphs