

Value-passing 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 .

We will generalise some of the processes.

Value-passing CCS

Given value set V , the actions in **value-passing CCS** are generalised to

- input action $a(x)$ with process $a(x).P$, where action name a can be viewed as a port or channel which can receive an incoming value to replace the free variable x in P ; variable x is bound in the process $a(x).P$.
- output action $\bar{a}\langle v \rangle$ with process $\bar{a}\langle v \rangle.P$, where $v \in V$ or v is a variable; variable x is free in process $\bar{a}\langle x \rangle.P$.

We typically work with **closed** processes: for example, $a(x).b\langle x \rangle.0$ and $c(y).a(x).b\langle y \rangle.0$, rather than $a(x).b\langle y \rangle.0$.

In fact, it is possible to generalise to $a(\tilde{x}).P$, for distinct list of variables \tilde{x} , and $\bar{a}\langle \tilde{v} \rangle.P$.

Transitions for Value-passing CCS

Adapted Transition Rules

$$\text{Rule(in)} \quad a(x).P \xrightarrow{a(v)} P\{v/x\}, \quad v \in V$$

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

$$\text{Rule(out)} \quad \bar{a}\langle v \rangle.P \xrightarrow{\bar{a}\langle v \rangle} P, \quad v \in V$$

$$\text{Rules}(\tau) \quad \frac{P \xrightarrow{a(v)} P' \quad Q \xrightarrow{\bar{a}\langle v \rangle} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

$$\frac{P \xrightarrow{\bar{a}\langle v \rangle} P' \quad Q \xrightarrow{a(v)} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

Example: Copier

$$\begin{aligned} \text{Cop} &\stackrel{\text{def}}{=} \text{in}(x).\overline{\text{out}}\langle x\rangle.\text{Cop} \\ \frac{\text{in}(x).\overline{\text{out}}\langle x\rangle.\text{Cop} \xrightarrow{\text{in}(v)} \overline{\text{out}}\langle v\rangle.\text{Cop}}{\text{Cop} \xrightarrow{\text{in}(v)} \overline{\text{out}}\langle v\rangle.\text{Cop}} \end{aligned}$$

Exercise

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

(a) $\text{Cop} \stackrel{\text{def}}{=} \text{in}(x).\overline{\text{out}}\langle x\rangle.\text{Cop}$

(b) $\text{Cop}_1 \stackrel{\text{def}}{=} \text{in}(x).\text{in}(x).\overline{\text{out}}\langle x\rangle.\text{Cop}_1$

(c) $\text{Cop}_2 \stackrel{\text{def}}{=} \text{in}(x).\overline{\text{out}}\langle x\rangle.\overline{\text{out}}\langle x\rangle.\text{Cop}_2$

Example: User of a Copier

$$\begin{aligned} \text{Cop} &\stackrel{\text{def}}{=} \text{in}(x).\overline{\text{out}}\langle x \rangle.\text{Cop} \\ \text{User} &\stackrel{\text{def}}{=} \text{write}(x).\text{User}_x \\ \text{User}_v &\stackrel{\text{def}}{=} \overline{\text{in}}\langle v \rangle.\text{User} \end{aligned}$$

$$\frac{\frac{\text{in}(x).\overline{\text{out}}\langle x \rangle.\text{Cop} \xrightarrow{\text{in}(v)} \overline{\text{out}}\langle v \rangle.\text{Cop} \quad \vdots}{\text{Cop} \xrightarrow{\text{in}(v)} \overline{\text{out}}\langle v \rangle.\text{Cop}} \quad \text{User}_v \xrightarrow{\overline{\text{in}}\langle v \rangle} \text{User}}{\text{Cop} \mid \text{User}_v \xrightarrow{\tau} \overline{\text{out}}\langle v \rangle.\text{Cop} \mid \text{User}}$$

Example: More Users

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

$$\text{User} \stackrel{\text{def}}{=} \text{write}(x).\text{User}_x$$

$$\text{User}_v \stackrel{\text{def}}{=} \overline{\text{in}}\langle v\rangle.\text{User}$$

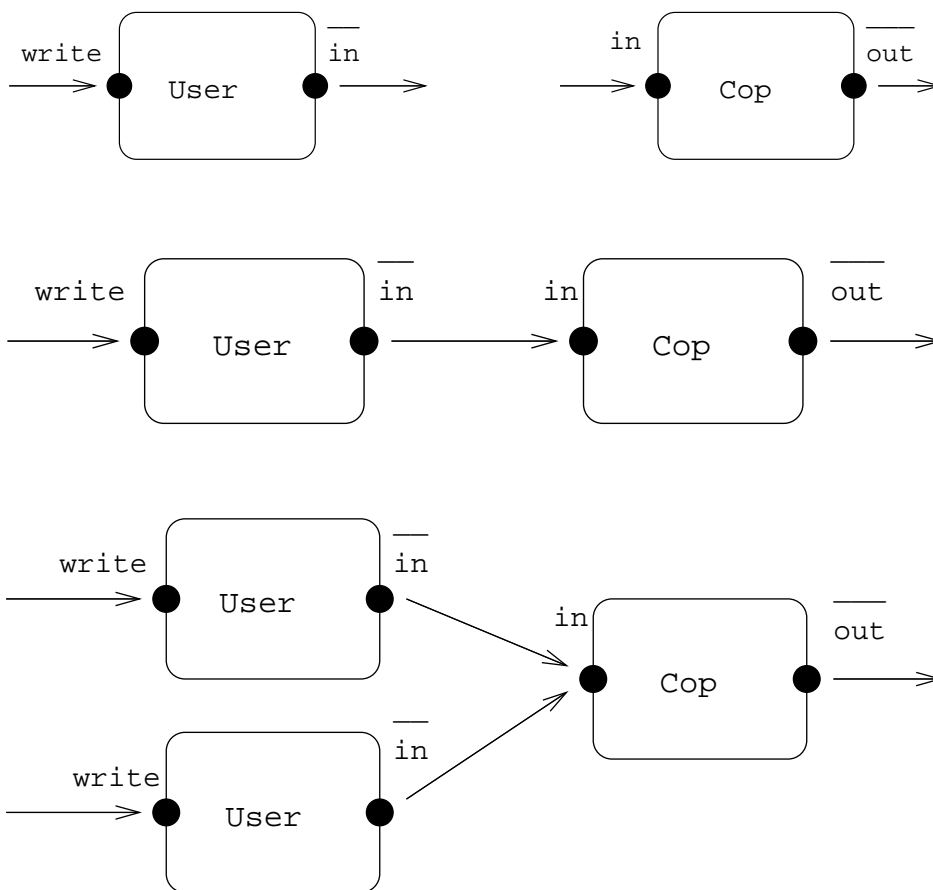
$$\frac{\begin{array}{c} \overline{\text{in}}\langle v_1\rangle.\text{User} \xrightarrow{\overline{\text{in}}\langle v_1\rangle} \text{User} \\ \hline \text{User}_{v_1} \xrightarrow{\overline{\text{in}}\langle v_1\rangle} \text{User} \end{array}}{\text{Cop} \xrightarrow{\text{in}(v_1)} \overline{\text{out}}\langle v_1\rangle.\text{Cop} \quad \text{User}_{v_1} \mid \text{User}_{v_2} \xrightarrow{\overline{\text{in}}\langle v_1\rangle} \text{User} \mid \text{User}_{v_2}}{\text{Cop} \mid (\text{User}_{v_1} \mid \text{User}_{v_2}) \xrightarrow{\tau} \overline{\text{out}}\langle v_1\rangle.\text{Cop} \mid (\text{User} \mid \text{User}_{v_2})}$$

Exercise

Derive $(\text{Cop} \mid \text{User}_{v1}) \mid \text{User}_{v2} \xrightarrow{\tau} (\overline{\text{out}}\langle v1 \rangle.\text{Cop} \mid \text{User}) \mid \text{User}_{v2}$.

Flow Graphs

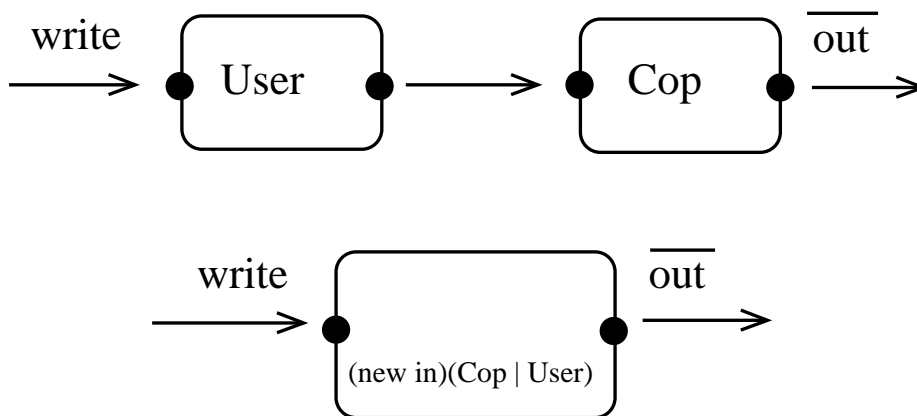
Flow graphs summarise the potential movement of information flowing into and out of ports.



Flow graphs will not be examined.

Flow graphs continued

$(\text{new in})(\text{Cop} \mid \text{User})$ is pictured as



Exercise

Give the flow graph of
 $\text{new up, down}(\text{Worker} \mid \text{Worker} \mid \text{Tool})$

Example: With Restriction

$$\begin{array}{c}
 \frac{\text{in}(x).\overline{\text{out}}\langle x\rangle.\text{Cop} \xrightarrow{\text{in}(v)} \overline{\text{out}}\langle v\rangle.\text{Cop} \quad \overline{\text{in}}\langle v\rangle.\text{User} \xrightarrow{\overline{\text{in}}\langle v\rangle} \text{User}}{\text{Cop} \xrightarrow{\text{in}(v)} \overline{\text{out}}\langle v\rangle.\text{Cop} \quad \text{User}_v \xrightarrow{\overline{\text{in}}\langle v\rangle} \text{User}} \\
 \hline
 \text{Cop} \mid \text{User}_v \xrightarrow{\tau} \overline{\text{out}}\langle v\rangle.\text{Cop} \mid \text{User} \\
 \hline
 \text{new in}(\text{Cop} \mid \text{User}_v) \xrightarrow{\tau} \text{new in}(\overline{\text{out}}\langle v\rangle.\text{Cop} \mid \text{User})
 \end{array}$$

Equations

Process descriptions can become quite large, especially when they consist of multiple components. The equation $A \stackrel{\text{def}}{=} P$ simply means that A abbreviates P .

Example

Road $\stackrel{\text{def}}{=} \text{car.up.}\overline{\text{ccross}}.\overline{\text{down}}.\text{Road}$

Rail $\stackrel{\text{def}}{=} \text{train.green.}\overline{\text{tcross}}.\overline{\text{red}}.\text{Rail}$

Signal $\stackrel{\text{def}}{=} \overline{\text{green}}.\text{red}.\text{Signal} +$
 $\overline{\text{up}}.\text{down}.\text{Signal}$

Crossing = new K (Road | Rail | Signal)

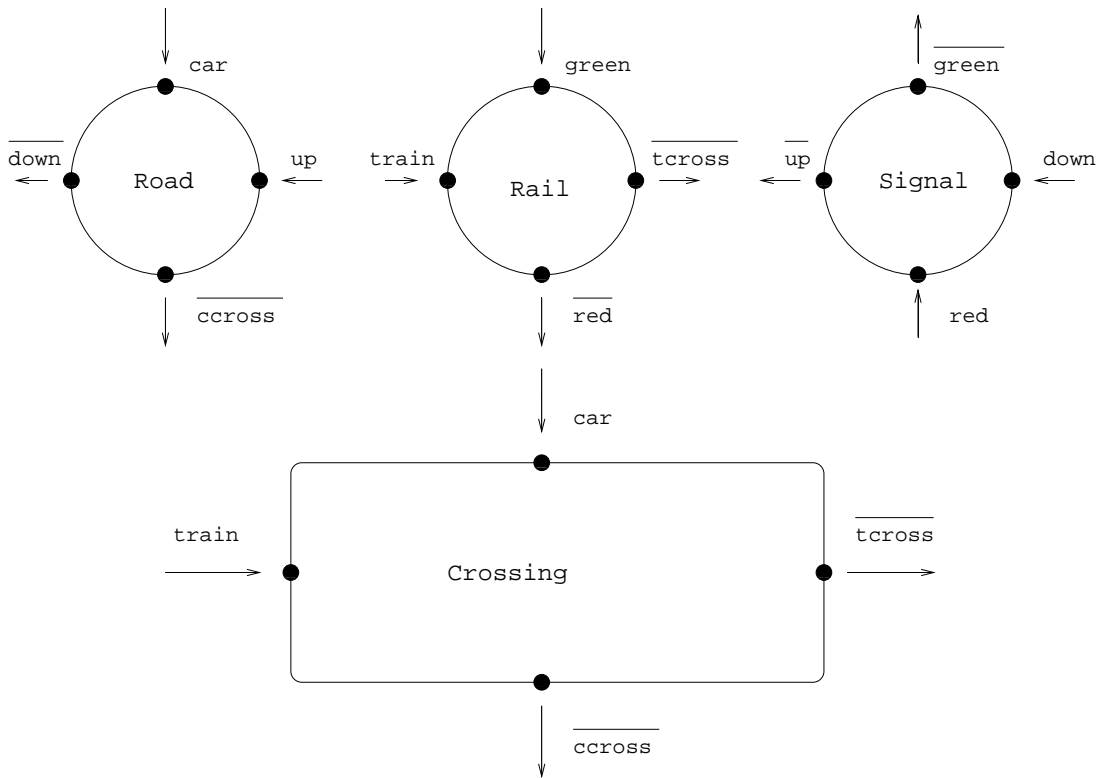
where $K = \{\text{green, red, up, down}\}$

Notation new K (P) denotes

(new green)(new red)(new up)(new down) P

We also sometimes write $P \setminus K$ instead of new K (P).

Flow Graph



Generalising Equations

In value-passing CCS, the process equation

$A \stackrel{\text{def}}{=} P$ can be generalised to

$$A(x) \stackrel{\text{def}}{=} P,$$

where the x binds any free x in P .

The process $A\langle v \rangle$ denotes a particular instance of the process equation with v (a value or variable) used in place of the bound name x .

Adapted transition rule

$$\text{Rule}(\stackrel{\text{def}}{=}) \quad \frac{P\{v/x\} \xrightarrow{\alpha} Q}{A\langle v \rangle \xrightarrow{\alpha} Q} \quad A(x) \stackrel{\text{def}}{=} P, v \in V$$

Again, it is simple to generalise to $A(\tilde{x}) \stackrel{\text{def}}{=} P$, for distinct list of variables \tilde{x} , and $A\langle \tilde{v} \rangle$.

Protocol that may lose messages

Sender $\stackrel{\text{def}}{=} \text{in}(x).\overline{\text{sm}}\langle x\rangle.\text{Send1}\langle x\rangle$

$\text{Send1}(x) \stackrel{\text{def}}{=} \text{ms}.\overline{\text{sm}}\langle x\rangle.\text{Send1}\langle x\rangle + \text{ok}.\text{Sender}$

Medium $\stackrel{\text{def}}{=} \text{sm}(y).\text{Med1}\langle y\rangle$

$\text{Med1}(y) \stackrel{\text{def}}{=} \overline{\text{mr}}\langle y\rangle.\text{Medium} + \tau.\overline{\text{ms}}.\text{Medium}$

Receiver $\stackrel{\text{def}}{=} \text{mr}(x).\overline{\text{out}}\langle x\rangle.\overline{\text{ok}}.\text{Receiver}$

Protocol = new sm, ms, mr, ok (Sender | Medium | Receiver)

Exercise Draw the transition graph of Protocol, assuming just one value v .

Renaming

Canonical buffer

$$B \stackrel{\text{def}}{=} i(x).\bar{o}\langle x\rangle.B$$

Renaming

We write $P[b/a]$ for the process obtained by replacing action name a by b in P .

Transition Rule for Renaming

$$\text{Rule}([\cdot]) \frac{P \xrightarrow{\beta} Q}{P[b/a] \xrightarrow{\alpha} Q[b/a]} \alpha = \beta[b/a]$$

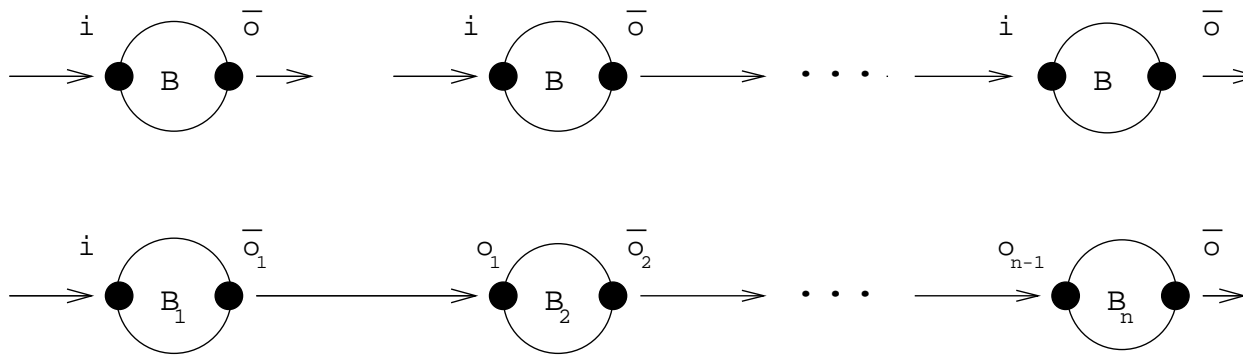
Example

Cop has the same behaviour as $B[\text{in}/i][\text{out}/o]$.

We can generalise to $P[\tilde{b}_i/\tilde{a}_i]$, for list of action names \tilde{a}_i, \tilde{b}_i with the a_i distinct.

Building an n -place buffer

$$B \stackrel{\text{def}}{=} i(x).\bar{o}\langle x \rangle.B$$



$$B_1 = B[o_1/o]$$

$$B_{j+1} = B[o_j/i][o_{j+1}/o] \quad 1 \leq j < n - 1$$

$$B_n = B[o_{n-1}/i]$$

n-place buffer new $o_1, \dots, o_n (B_1 \mid \dots \mid B_n)$

A Scheduler

Problem: assume n tasks when $n > 1$.

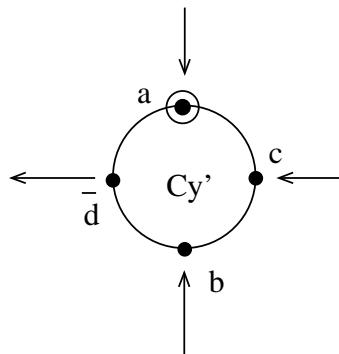
a_i initiates the i th task

b_i signals its completion

The scheduler plans the order of task initiation, ensuring

1. actions $a_1 \dots a_n$ carried out cyclically
2. tasks may terminate in any order
3. but a task can not be restarted until its previous operation has finished. (a_i and b_i happen alternately for each i .)

$$Cy' \stackrel{\text{def}}{=} a.c.b.\bar{d}.Cy'$$

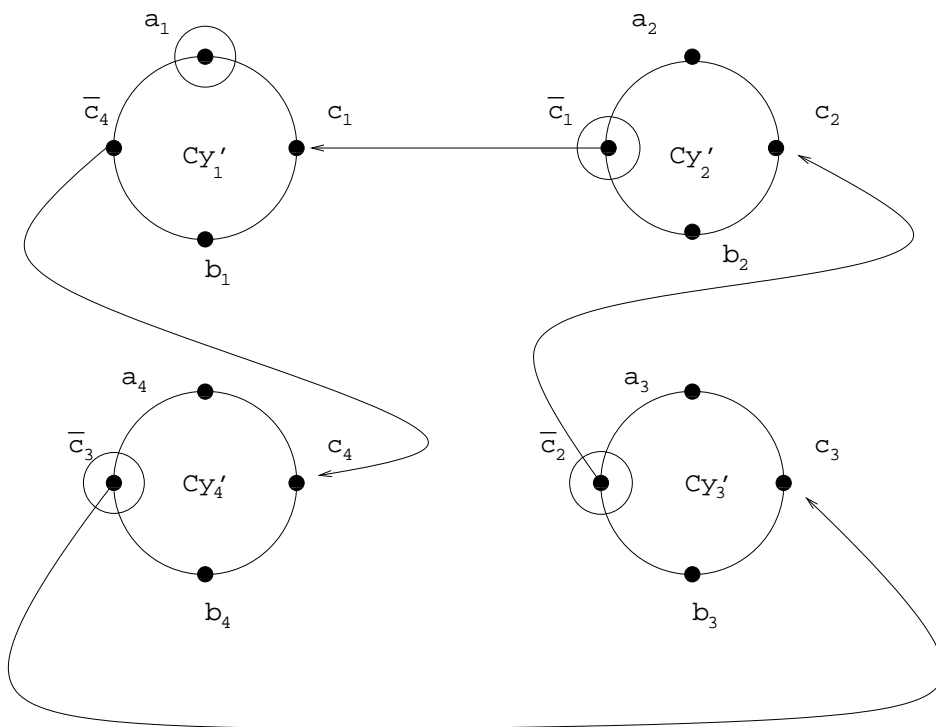


What is wrong?

$$Cy'_1 = Cy'[a_1/a][c_1/c][b_1/b][c_n/d]$$

$$Cy'_i = (\bar{d}.Cy')[a_i/a][c_i/c][b_i/b][c_{i-1}/d],$$

$$1 < i \leq n$$



$$\text{Sched}'_4 = \text{new } c_1, \dots, c_4 (Cy'_1 | Cy'_2 | Cy'_3 | Cy'_4)$$

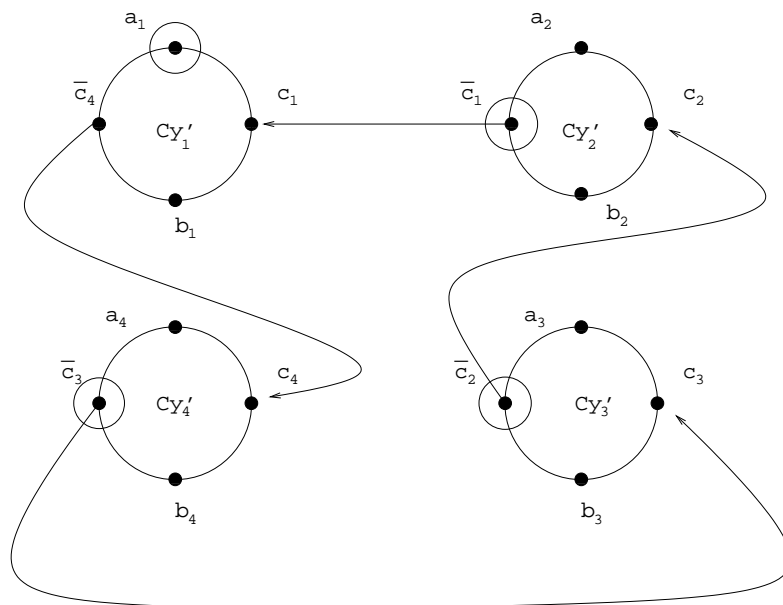
A solution

$$Cy \stackrel{\text{def}}{=} a.c.(b.\bar{d}.Cy + \bar{d}.b.Cy)$$

$$Cy_1 = Cy[a_1/a][c_1/c][b_1/b][c_n/d]$$

$$Cy_i = (\bar{d}.Cy)[a_i/a][c_i/c][b_i/b][c_{i-1}/d],$$

$$1 < i \leq n$$



How do we know it is right?

Summary: processes

The set of **CCS processes** is given by

$$P ::= 0 \mid \alpha.P \mid A \mid P + Q \mid \\ P \mid Q \mid (\text{new } a) P \mid P[b/a]$$

where

- α is an action of the form a , \bar{a} or τ ;
- a, b are action names;
- A is a process identifier with equation $A \stackrel{\text{def}}{=} P$.

In **value-passing CCS**,

- actions are generalised to $a(x)$, $\bar{a}\langle v \rangle$ or τ , where v is a value or a variable, and variable x is bound in the process $a(x).P$;
- equations are generalised to $A(x) \stackrel{\text{def}}{=} P$, the x binds free x in P , and called by the process $A\langle v \rangle$ where v is either a value or a variable.

Choice is sometimes generalised to $\Sigma\{P_i : i \in I\}$.

Summary: CCS transitions

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

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

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

$$\text{Rule}(|) \quad \frac{P \xrightarrow{\alpha} P'}{P | Q \xrightarrow{\alpha} P' | Q} \quad \frac{Q \xrightarrow{\alpha} Q'}{P | Q \xrightarrow{\alpha} P | Q'}$$

$$\text{Rules}(\tau) \quad \frac{P \xrightarrow{a} P' \quad Q \xrightarrow{\bar{a}} Q'}{P | Q \xrightarrow{\tau} P' | Q'} \quad \frac{P \xrightarrow{\bar{a}} P' \quad Q \xrightarrow{a} Q'}{P | Q \xrightarrow{\tau} P' | Q'}$$

$$\text{Rule}(\text{new}) \quad \frac{P \xrightarrow{\alpha} Q}{(\text{new } a) P \xrightarrow{\alpha} (\text{new } a) Q} \quad \alpha \neq a, \bar{a}$$

$$\text{Rule}([\cdot]) \quad \frac{P \xrightarrow{\beta} Q}{P[b/a] \xrightarrow{\alpha} Q[b/a]} \quad \alpha = \beta[b/a]$$

Summary: value-passing CCS transitions

Replace Rule(\cdot), Rules(τ) and Rule($\stackrel{\text{def}}{=}$) by

$$\text{Rule(in)} \quad a(x).P \xrightarrow{a(v)} P\{v/x\}, \quad v \in V$$

$$\text{Rule(out)} \quad \bar{a}\langle v \rangle.P \xrightarrow{\bar{a}\langle v \rangle} P, \quad v \in V$$

$$\text{Rules}(\tau) \quad \frac{P \xrightarrow{a(v)} P' \quad Q \xrightarrow{\bar{a}\langle v \rangle} Q'}{P | Q \xrightarrow{\tau} P' | Q'}$$

$$\frac{P \xrightarrow{\bar{a}\langle v \rangle} P' \quad Q \xrightarrow{a(v)} Q'}{P | Q \xrightarrow{\tau} P' | Q'}$$

$$\text{Rule}(\stackrel{\text{def}}{=}) \quad \frac{P\{v/x\} \xrightarrow{\alpha} Q}{A\langle v \rangle \xrightarrow{\alpha} Q} \quad A(x) \stackrel{\text{def}}{=} P, v \in V$$