An example

Let's fix the alphabet $V = \{e, c, t, cl\}$

$$
\dfrac{\dfrac{\dfrac{\dfrac{e \in V}{e \xrightarrow{e} 1}\ (act)}{e.c \xrightarrow{e} c}\ (seq_2) \qquad c \neq 1}{e.c + e.t \xrightarrow{e} c}\ (cho_1) \qquad c \neq 1}{(e.c + e.t).cl \xrightarrow{e} c.cl}\ (seq_1)
$$

$$
\begin{tikzcd}
(e.c + e.t).cl \arrow[r, "e"] \arrow[dr, "e"'] & c.cl \arrow[r, "c"] & cl \arrow[r, "cl"] & 1 \\
 & t.cl \arrow[ur, "t"'] & &
\end{tikzcd}
$$

Exercise 10
Give the LTS of a*(b+c)

RegExp & their operational semantics

We saw that we can define the language of an FSA $M = (Q, \Sigma, q_0, \delta, F)$ as

$$\mathcal{L}_M = \{ a_1 \ldots a_n \in \Sigma^* \mid \exists q_1, \ldots, q_n \mid q_0 \xrightarrow{a_1} \ldots q_{n-1} \xrightarrow{a_n} q_n \xrightarrow{\checkmark} \bullet \}$$

where $\rightarrow$ is the relation of the LTS corresponding to M

This can be generalised to ANY LTS e.g.

Since the TSS of reg exp induces an LTS, we can use the very same definition to define the language $\mathcal{L}_E$ of a reg exp E; so

$$\mathcal{L}_E = \{ a_1 \ldots a_n \in \Sigma^* \mid \exists E_1, \ldots, E_n : E \xrightarrow{a_1} E_1 \ldots E_{n-1} \xrightarrow{a_n} E_n = 1 \}$$

where now $\xrightarrow{a_i}$ are transition to be proved by applying the rules of our TSS!

Example .

Show that $a\,a\,b \in \mathcal{L}_E$ where $E = a^*(b + c)$ and $A = \{a, b, c, d\}$

1. find E_1, E_2 s.t. $E_1 \xrightarrow{a} E_2$ & there is E_3 s.t. $E_2 \xrightarrow{a} E_3 \xrightarrow{b} 1$
   - a candidate for E_3 is b since (Act) $\dfrac{b \in A}{b \xrightarrow{b} 1}$
   - likewise a candidate for E_2 is $a\,b$ WHY?

$$
(\mathrm{Act})\ \dfrac{\dfrac{a \in A}{a \xrightarrow{a} 1}}{a\,b \xrightarrow{a} b}\ (\mathrm{seq}_2)
$$

$$
\mathrm{Seq}_2\ \dfrac{\mathrm{act}\ \dfrac{a \in A}{a \xrightarrow{a} 1}}{a\,a\,b \xrightarrow{a} a\,b}
\qquad
\dfrac{\dfrac{a \in A}{a \xrightarrow{a} 1}\ \mathrm{act}}{a\,b \xrightarrow{a} b}\ \mathrm{Seq}_2
\qquad
\dfrac{b \in A}{b \xrightarrow{b} 1}\ \mathrm{act}
$$

A formal model of concurrency [Bergstra et al.]

From regular expressions to process algebras: a model of concurrency

$A_\tau = A \cup \{\tau\}$ $\qquad \tau \notin A$

Note the different yet equivalent definition wrt [Bergstra et al.]

$E ::= \ \dots \ \mid E \parallel E$ without Kleene star and 0

(Act) $\dfrac{a \in A_\tau}{a \xrightarrow{a} 1}$

(Cho1) $\dfrac{x \xrightarrow{a} x' \quad x' \neq 1}{x + y \xrightarrow{a} x'}$ $\qquad$ (Cho2) $\dfrac{x \xrightarrow{a} 1}{x + y \xrightarrow{a} 1}$

(Cho3) $\dfrac{y \xrightarrow{a} y' \quad y' \neq 1}{x + y \xrightarrow{a} y'}$ $\qquad$ (Cho4) $\dfrac{y \xrightarrow{a} 1}{x + y \xrightarrow{a} 1}$

(Seq1) $\dfrac{x \xrightarrow{a} x' \quad x' \neq 1}{x \cdot y \xrightarrow{a} x' \cdot y}$ $\qquad$ (Seq1) $\dfrac{x \xrightarrow{a} 1}{x \cdot y \xrightarrow{a} y}$

(per1) $\dfrac{x \xrightarrow{a} x' \quad x' \neq 1}{x \parallel y \xrightarrow{a} x' \parallel y}$

(per2) $\dfrac{y \xrightarrow{a} y' \quad y' \neq 1}{x \parallel y \xrightarrow{a} x \parallel y'}$

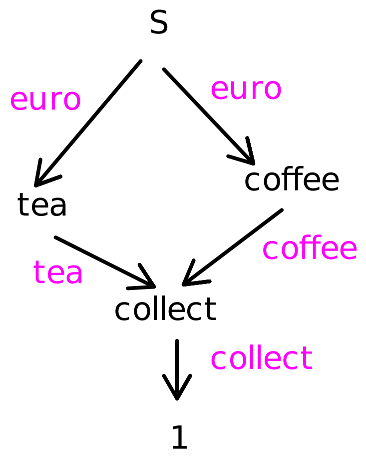
(per3) $\dfrac{x \xrightarrow{a} 1}{x \parallel y \xrightarrow{a} y}$

(per4) $\dfrac{y \xrightarrow{a} 1}{x \parallel y \xrightarrow{a} x}$

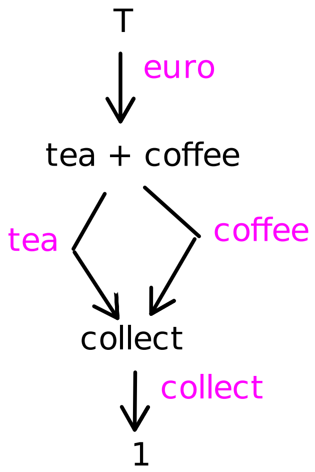
Interleaving semantics

Equivalences of concurrent programs

S = (euro.tea + euro.coffee).collect



T = euro.(tea + coffee).collect



- S and T have the same traces (words), but they differ if interpreted as reactive systems
- For reactive systems, bisimulation is a better notion of equivalence than language (trace) equivalence

Def. Given an LTS T, a binary relation B on the states of T is a bisimulation if whenever (s1,s2) are in B
- for all s1 --a--> s1' there is s2 --a--> s2' such that (s1',s2') is in B and
- for all s2 --a--> s2' there is s1 --a--> s1' such that (s1',s2') is in B

Exercise 11
Let S and T as in the example of the vending machine above. Show that there is no bisimulation containing the pair (S,T).

Recap of the previous class

Structural Operational Semantics
- regexp
- bpa

Concurrency as interleaving

Equivalences for concurrency

What about communication?

Let $A_\perp = A \cup \{\perp\}$ $\quad \perp \notin A$ and fix a communication function

$$\_\circ\_ : A_\perp \times A_\perp \longrightarrow A_\perp \quad \begin{cases} \circ \text{ commutative} \\ \circ \text{ associative} \\ \forall a \in A_\perp : a \circ \perp = \perp \circ a = \perp \end{cases}$$

$$(\mathrm{com}_1)\ \frac{x \xrightarrow{a} x' \quad y \xrightarrow{b} y' \quad a \circ b \in A}{x \parallel y \xrightarrow{a \circ b} x' \parallel y'} \qquad (\mathrm{com}_2)\ \frac{x \xrightarrow{a} 1 \quad y \xrightarrow{b} y' \quad a \circ b \in A}{x \parallel y \xrightarrow{a \circ b} y'}$$

$$(\mathrm{com}_3)\ \frac{x \xrightarrow{a} x' \quad y \xrightarrow{b} 1 \quad a \circ b \in A}{x \parallel y \xrightarrow{a \circ b} x'} \qquad (\mathrm{com}_4)\ \frac{x \xrightarrow{a} 1 \quad y \xrightarrow{b} 1 \quad a \circ b \in A}{x \parallel y \xrightarrow{a \circ b} 1}$$

Example

Show that $ax + by \parallel cz \xrightarrow{b} x \parallel z$ if $a \circ c = b$, $x \neq 1$, and $y \neq 1$

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{a \in A}{a \xrightarrow{a} 1}\,\text{Act}
}{ax \xrightarrow{a} x}\,\text{Seq}_1
}{ax + by \xrightarrow{a} x}\,\text{Cho}_1
\qquad\qquad
\dfrac{
\dfrac{c \in A}{c \xrightarrow{c} 1}\,\text{Act}
}{cz \xrightarrow{c} z}\,\text{Seq}_2
}{ax + by \parallel cz \xrightarrow{b} x \parallel z}\,\text{Com}_1
$$