

## Co je CCS proces pro své okolí?

CCS proces je výpočetní agent, který může komunikovat s okolním prostředím prostřednictvím svého rozhraní.

Rozhraní = Sada **komunikačních portů/kanálů**, spolu s informací, jestli jsou používány pro vstup nebo výstup.

## Příklad: počítačový vědec

Rozhraní procesu:

- coffee (vstupní port)
- coin (výstupní port)
- pub (výstupní port)

# Základy CCS (sekvenční fragment)

- *Nil* (or 0) proces (jediný atomický proces)
- akční prefix (*a.P*)
- jména a rekurzivní definice ( $\stackrel{\text{def}}{=}$ )
- nedeterministická volba (+)

Toto stačí k popisu sekvenčních procesů

Každý konečný LTS může být popsán za použití výše uvedených operací.

$$V_1 \stackrel{\text{def}}{=} 10k.10k.(coffee.collect.V_1 + tea.collect.V_1)$$

$$V_2 \stackrel{\text{def}}{=} 10k.10k.coffee.collect.V_2 + 10k.10k.tea.collect.V_2)$$

$$V_3 \stackrel{\text{def}}{=} (10k.10k + 20k).(coffee.collect.V_3 + tea.collect.V_3)$$

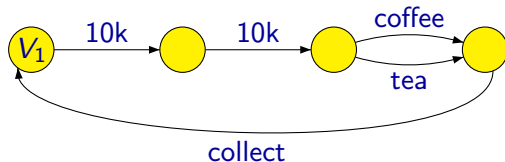
$$V_4 \stackrel{\text{def}}{=} 10k.10k.(coffee.collect.V_4 + tea.collect.V_4) + \\ 20k.(coffee.collect.V_4 + tea.collect.V_4)$$

$$V_5 \stackrel{\text{def}}{=} 10k.10k.V_6 + 20k.V_6$$

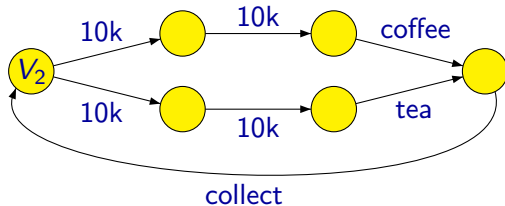
$$V_6 \stackrel{\text{def}}{=} coffee.collect.V_5 + tea.collect.V_5$$

# Nápojový automat

$$V_1 \stackrel{\text{def}}{=} 10k.10k.(coffee.collect.V_1 + tea.collect.V_1)$$

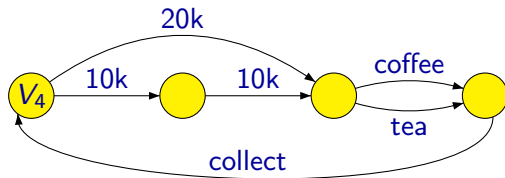


$$V_2 \stackrel{\text{def}}{=} 10k.10k.coffee.collect.V_2 + 10k.10k.tea.collect.V_2$$

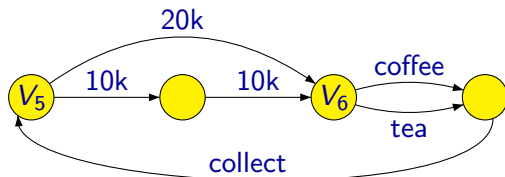


# Nápojový automat

$$V_4 \stackrel{\text{def}}{=} 10k.10k.(coffee.collect.V_4 + tea.collect.V_4) + 20k.(coffee.collect.V_4 + tea.collect.V_4)$$



$$V_5 \stackrel{\text{def}}{=} 10k.10k.V_6 + 20k.V_6, \quad V_6 \stackrel{\text{def}}{=} coffee.collect.V_5 + tea.collect.V_5$$



- paralelní kompozice ( $|$ )  
(synchronní komunikace mezi dvěma komponentami = handshake synchronizace)
- restrikce ( $P \setminus L$ )
- přejmenování ( $P[f]$ )

# Definice CCS (kanály, akce, jména procesů)

Označme:

- $\mathcal{A}$  množina **jmen kanálů** (channel names) (např. *tea*, *coffee* jsou kanály)
- $\mathcal{L} = \mathcal{A} \cup \overline{\mathcal{A}}$  množina **návěstí** (labels), kde
  - $\overline{\mathcal{A}} = \{\overline{a} \mid a \in \mathcal{A}\}$   
(prvky  $\mathcal{A}$  jsou jména (names),  
prvky  $\overline{\mathcal{A}}$  jsou doplňková jména (co-names))
  - podle konvence  $\overline{\overline{a}} = a$
- $Act = \mathcal{L} \cup \{\tau\}$  je množina **akcí** (actions), kde
  - $\tau$  je **vnitřní** (internal) nebo také **tichá** (silent) akce  
(např.  $\tau$ , *tea*, *coffee* jsou akce)
- $\mathcal{K}$  je množina **jmen procesů** nebo **konstant** (process names, constants) (např.  $V_5$ ).

# Definice CCS (výrazy)

$P := K$		procesní konstanty ( $K \in \mathcal{K}$ )
$\alpha.P$		akční prefixy ( $\alpha \in Act$ )
$\sum_{i \in I} P_i$		sumy ( $I$ je libovolná množina indexů)
$P_1   P_2$		paralelní kompozice (parallel composition)
$P \setminus L$		restrikce (restriction) ( $L \subseteq \mathcal{A}$ )
$P[f]$		přejmenování (relabelling) ( $f : Act \rightarrow Act$ ) tž. <ul style="list-style-type: none"><li>• <math>f(\tau) = \tau</math></li><li>• <math>f(\bar{a}) = \overline{f(a)}</math></li></ul>

Termy vygenerovatelné touto gramatikou nazýváme **procesní výrazy CCS** (CCS process expressions) a množinu všech těchto výrazů značíme  $\mathcal{P}$ .

## Notace

$$P_1 + P_2 = \sum_{i \in \{1,2\}} P_i$$

$$Nil = 0 = \sum_{i \in \emptyset} P_i$$



## Priority operátorů

- 1 restrikce a přejmenování (nejvyšší priorita)
- 2 akční prefix
- 3 paralelní kompozice
- 4 suma

Příklad:  $R + a.P|b.Q \setminus L$  znamená  $R + ((a.P)|(b.(Q \setminus L)))$ .

# Definice CCS (definující rovnice)

## CCS program

**CCS programem** nazýváme sadu **definujících rovnic** (defining equations) tvaru

$$K \stackrel{\text{def}}{=} P,$$

kde  $K \in \mathcal{K}$  je procesní konstanta a  $P \in \mathcal{P}$  je procesní výraz CCS.

- Povolena jen jedna definující rovnice pro každou procesní konstantu.
- Je povolena rekurze – např.  $A \stackrel{\text{def}}{=} \bar{a}.A \mid A$ .

Syntaxe

CCS

(sada definujících rovnic)



Sémantika

LTS

(ohodnocené přechodové  
systémy)

JAK?

## Strukturální operační sémantika (SOS - Structural Operational Semantics) – G. Plotkin 1981

Tzv. small-step operační sémantika sémantika, kde chování systému je odvozeno s využitím pravidel řízených syntaxí.

Pro danou sadu definujících rovnic definujeme následující LTS  
( $Proc, Act, \{\xrightarrow{a} \mid a \in Act\}$ ):

- $Proc = \mathcal{P}$  (množina všech procesních výrazů CCS)
- $Act = \mathcal{L} \cup \{\tau\}$  (množina všech CCS akcí včetně  $\tau$ )
- přechodová relace je dána **SOS pravidly** tvaru:

PRAVIDLO  $\frac{\text{předpoklady}}{\text{závěr}}$  podmínky

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

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

$$\text{COM1} \frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q}$$

$$\text{COM2} \frac{Q \xrightarrow{\alpha} Q'}{P|Q \xrightarrow{\alpha} P|Q'}$$

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

$$\text{RES} \frac{P \xrightarrow{\alpha} P'}{P \setminus L \xrightarrow{\alpha} P' \setminus L} \quad \alpha, \bar{\alpha} \notin L$$

$$\text{REL} \frac{P \xrightarrow{\alpha} P'}{P[f] \xrightarrow{f(\alpha)} P'[f]}$$

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

# Odvození přechodu v CCS

Nechť  $A \stackrel{\text{def}}{=} a.A$ . Potom

$$((A | \bar{a}.Nil) | b.Nil)[c/a] \xrightarrow{c} ((A | \bar{a}.Nil) | b.Nil)[c/a].$$

$$\begin{array}{c} \text{ACT} \frac{}{a.A \xrightarrow{a} A} \\ \text{CON} \frac{a.A \xrightarrow{a} A}{A \xrightarrow{a} A} \quad A \stackrel{\text{def}}{=} a.A \\ \text{COM1} \frac{}{A | \bar{a}.Nil \xrightarrow{a} A | \bar{a}.Nil} \\ \text{COM1} \frac{}{(A | \bar{a}.Nil) | b.Nil \xrightarrow{a} (A | \bar{a}.Nil) | b.Nil} \\ \text{REL} \frac{}{((A | \bar{a}.Nil) | b.Nil)[c/a] \xrightarrow{c} ((A | \bar{a}.Nil) | b.Nil)[c/a]} \end{array}$$

