Concurrency 3

CCS

Jean-Jacques Lévy

jeanjacqueslevy.net/dea

Minimal language for concurrency

The λ-calculus is a minimal language for functional languages. It can also be used as a basis for imperative languages (via continuations).

What is a minimal language for concurrent processes ?

- CCS [Milner]
- π-calculus [Milner, Parrow, Walker, Sangiorgi]
- CSP [Hoare]
- Petri nets
- Mazurkiewitz traces
- Events structures $\Leftrightarrow$ True concurrency [Winskel]
- IO-automatas [Lynch, Tuttle]

CCS (1/4)

Language

a, b, c	$::=$	(channel) names	
$\overline{a}, \overline{b}, \overline{c}$	$::=$	co-names $\quad \overline{\overline{a}} = a$	
α	$::=$	$a \mid \overline{a} \mid \tau$	actions
P, Q, R	$::=$	$0 \mid \alpha.P \mid P + Q \mid (P \mid Q) \mid (\nu\alpha)P \mid K$	processes
$K \stackrel{\text{def}}{=} P$	$::=$	constant definitions	

$\mathcal{A}ct = \{a, b, c, \ldots\} \cup \{\overline{a}, \overline{b}, \overline{c}, \ldots\} \cup \{\tau\}$ Notation: α for $\alpha.0$

- 0 null process
- $\alpha.P$ sequential action
- $P + Q$ non-deterministic (external) choice
- $P \mid Q$ parallel composition
- $(\nu\alpha)P$ restriction on α
- K (recursively defined) constant

CCS (2/4)

Examples (coffee machine revisited)

$P_0 = A$ $\quad$ $P_0' = B$ $\quad$ $C = (k.d.D + t.d.D)$

$A = c.(k.d.A + t.d.A)$ $\quad$ $B = c.C$ $\quad$ $D = c.C$

$P_0'' = E$ $\quad$ $P_0''' = F$

$E = (c.k.d.E + c.t.d.E)$ $\quad$ $F = c + (c.k.d.F + c.t.d.F)$

Interaction with coffee machine

$P_0 \mid \overline{c}.\overline{k}.\overline{d}$ $\quad$ $P_0 \mid \overline{c}.\overline{k}.\overline{d} \mid \overline{c}.\overline{t}.\overline{d}$

$P_0 \mid$ Client1 $\quad$ $P_0' \mid$ Client1 $\quad$ $P_0'' \mid$ Client1 $\quad$ $P_0''' \mid$ Client1

$P_0 \mid$ Client2 $\quad$ $P_0'' \mid$ Client2 $\mid$ Client2 $\quad$ $P_0 \mid$ Client1 $\mid$ Client2

where

Client1 $\stackrel{\text{def}}{=} \overline{c}.\overline{k}.\overline{d}.$Client1 $\quad$ Client2 $\stackrel{\text{def}}{=} \overline{c}.\overline{t}.\overline{d}.$Client2

CCS (3/4)

Semantics (SOS)

[Act] $\alpha.P \xrightarrow{\alpha} P$

[Sum1] $\dfrac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'}$

[Sum2] $\dfrac{Q \xrightarrow{\alpha} Q'}{P + Q \xrightarrow{\alpha} Q'}$

[Par1] $\dfrac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}$

[Par2] $\dfrac{Q \xrightarrow{\alpha} Q'}{P \mid Q \xrightarrow{\alpha} P \mid Q'}$

[Com] $\dfrac{P \xrightarrow{a} P' \quad Q \xrightarrow{\overline{a}} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$

[Res] $\dfrac{P \xrightarrow{\alpha} P' \quad \alpha \notin \{a, \overline{a}\}}{(\nu a)P \xrightarrow{\alpha} (\nu a)P'}$

[Rec] $\dfrac{P \xrightarrow{\alpha} P' \quad K \overset{\text{def}}{=} P}{K \xrightarrow{\alpha} P'}$

$\xrightarrow{\tau}$ internal move
$\xrightarrow{\alpha}$ $(\alpha \neq \tau)$ interaction on external α-channel

By convention, $\xrightarrow{a}$ input on a-channel and $\xrightarrow{\overline{a}}$ output on a-channel

Sum $\neq$ internal choice $P + Q \xrightarrow{\tau} P$ or $P + Q \xrightarrow{\tau} Q$.

CCS (4/4)

At present time, no values passed on communication channels.
(see later for value passing calculi)

No buffering in communications. Different from TCP sockets, from Kahn/MacQueen flow systems.
$\Rightarrow$ communication by rendez-vous.
$\equiv$ more basic calculus.

Rendez-vous exist in Occam, Ada, CML, Ocaml's processes.

CCS and strong bisimulation (1/4)

Theorem 1 Following relations hold.

$$P + 0 \sim P \qquad\qquad P \mid 0 \sim P$$
$$P + Q \sim Q + P \qquad\qquad P \mid Q \sim Q \mid P$$
$$(P + Q) + R \sim P + (Q + R) \qquad\qquad (P \mid Q) \mid R \sim P \mid (Q \mid R)$$
$$P + P \sim P$$

$$(\nu a)(P \mid Q) \sim ((\nu a)P) \mid Q \qquad \text{if } a \text{ not free in } Q$$
$$(\nu a)(\nu b)P \sim (\nu b)(\nu a)P$$
$$(\nu a)P \sim (\nu b)P\{b/a\} \qquad \text{if } b \text{ not bound in } Q$$
$$(\nu a)\alpha.P \sim 0 \qquad \text{if } \alpha = a \text{ or } \alpha = \overline{a}$$
$$(\nu a)\alpha.P \sim \alpha.(\nu a).P \qquad \text{otherwise}$$

$$K \sim P \qquad \text{if } K \stackrel{\text{def}}{=} P$$

CCS and strong bisimulation (2/4)

Proof of previous theorem

- $P + 0 \sim P$. Take $\mathcal{R} = \{(P + 0, P), (P, P + 0), (P, P)\}$ and show $\mathcal{R}$ is a bisimulation.

  Let $P + 0 \xrightarrow{\alpha} P'$. Then $P \xrightarrow{\alpha} P'$ by rule [Sum1] since $0 \xrightarrow{\alpha} P'$ is not possible. And $P' \mathcal{R} P'$.

  Conversely let $P \xrightarrow{\alpha} P'$. Then $P + 0 \xrightarrow{\alpha} P'$ by rule [Sum1] . And again $P' \mathcal{R} P'$.

- $P + Q \sim Q + P$. Show following $\mathcal{R}$ is a bisimulation. Take $\mathcal{R} = \{P + Q, Q + P, (P, P)\}$.

  Let $P + Q \xrightarrow{\alpha} S$.

  - Case 1: let $P + Q \xrightarrow{\alpha} S$ using [Sum1] . Then $P \xrightarrow{\alpha} S$.
    But $Q + P \xrightarrow{\alpha} S$ using [Sum2] .
    QED since $S \mathcal{R} S$.
  - Case 2: let $P + Q \xrightarrow{\alpha} S$ using [Sum2] . Then $Q \xrightarrow{\alpha} S$.
    But $Q + P \xrightarrow{\alpha} S$ using [Sum1] .
    QED since $S \mathcal{R} S$.

  Conversely let $Q + P \xrightarrow{\alpha} S$. QED by symmetry.

CCS and strong bisimulation (3/4)

Proof of theorem (continued)

- $(P+Q)+R \sim P+(Q+R)$. Show following $\mathcal{R}$ is a bisimulation.
  Take $\mathcal{R} = \{(P+Q)+R, P+(Q+R), (P,P)\}$.
  Let $(P+Q)+R \xrightarrow{\alpha} S$.
  - Case 1: let $(P+Q) \xrightarrow{\alpha} S$ using [Sum1] .
    * Case 1.1: let $P \xrightarrow{\alpha} S$ using [Sum1] .
      Then $P+(Q+R) \xrightarrow{\alpha} S$ by [Sum1] .
      QED since $S \mathcal{R} S$.
    * Case 1.2: Let $Q \xrightarrow{\alpha} S$. Then $(Q+R) \xrightarrow{\alpha} S$ by [Sum1] , and $P+(Q+R) \xrightarrow{\alpha} S$ by [Sum2] .
      QED since $S \mathcal{R} S$.
  - Case 2: Let $R \xrightarrow{\alpha} S$ by [Sum2] . Then $(Q+R) \xrightarrow{\alpha} S$ by [Sum2] , and $P+(Q+R) \xrightarrow{\alpha} S$ by [Sum2] .
    QED since $S \mathcal{R} S$.

  By symmetry when $P+(Q+R) \xrightarrow{\alpha} S$.
- other equations ...

Exercice 1 Give full proof of theorem.

CCS and strong bisimulation (4/4)

Theorem 2 [Expansion]

$$a.P \mid b.Q \;\sim\; a.(P \mid b.Q) + b.(a.P \mid Q)$$
$$a.P \mid \overline{a}.Q \;\sim\; a.(P \mid \overline{a}.Q) + \overline{a}.(a.P \mid Q) + \tau.(P \mid Q)$$

Exercice 2 Prove it.

Concurrency in CCS relies on interleaving. Never two actions occur at same time. Different from "true concurrency".

Exercice 3 Draw LTS for following processes:

$$P \;=\; (\nu a)((a+b) \mid \overline{a}) \qquad\qquad K_2 \stackrel{\text{def}}{=} \tau.(\nu a)(a \mid (\overline{a}+b)) + c.K_3$$
$$K_1 \stackrel{\text{def}}{=} a.(\tau.K_1 + b) + \tau.a.K_1 \qquad\qquad K_3 \stackrel{\text{def}}{=} d.K_3$$

Exercice 4 Draw LTS for $(\nu c)(K_1 \mid K_2)$ where

$$K_1 \stackrel{\text{def}}{=} a.\overline{c}.K_1 \qquad\qquad K_2 \stackrel{\text{def}}{=} b.c.K_2$$

Exercice 5 Give a CCS term for boolean semaphores.
Exercice 6 Give a CCS term for n-ary semaphores.

Strong bisimulation and congruence

Theorem 3 Strong bisimulation $\sim$ is a congruence. Namely:

$$P \sim Q \quad \Rightarrow \quad C[P] \sim C[Q] \quad \text{for any context } C[\,].$$

Exercice 7 Prove it.

This means that $\sim$ can be used as standard equations.

Exercice 8 Prove by using equations of Theorems 1 and 2 that:
$(\nu b)(a.(b \mid c) + \tau.(b \mid \overline{b}.c)) \sim a.c + \tau.\tau.c$

Exercice 9 Show $K \mid K \sim K$ when $K \stackrel{\text{def}}{=} a.K$.

Exercice 10 Show $K \sim K'$ when $K \stackrel{\text{def}}{=} a.K$ and $K' \stackrel{\text{def}}{=} a.a.K'$.

Exercice 11 Show $K \sim a.K'$ when $K \stackrel{\text{def}}{=} a.b.K$ and $K' \stackrel{\text{def}}{=} b.a.K'$.

Exercice 12 Show that $a.(b + c) \not\sim ab + ac$.