CIMPA-UNESCO School, Bangalore, 2005:

Models of Concurrency

Jean-Jacques Lévy

INRIA

jeanjacqueslevy.net/cimpa.html

Semantics

A semantics function $[\![\cdots]\!]$ assigns meaning $[\![M]\!]$ to terms M.

The induced relation $\simeq$ defined by $M \simeq N$ iff $[\![M]\!] = [\![N]\!]$ must be:

1. compositional, i.e.
   $M \simeq N$ implies $C[M] \simeq C[N]$ for any context $C[\,]$, i.e.
   $\simeq$ is a congruence
2. consistent with observation, i.e.
   if M produces α and $M \simeq N$, then N produces α
3. keeping choices (more specific to non-determinism), i.e.
   branching time semantics, i.e.
   bisimulation [Milner]

Last item is more ideologic than necessary.
Bisimulation are useful for proofs.

Examples

- λ-calculus [Church]
  $M, N ::= x \mid \lambda x.M \mid M N$
  $M \simeq_e N$ iff $\forall C[\,]\ C[M] \longrightarrow^* nf$ *implies* $C[N] \longrightarrow^* nf$
  $M \simeq_w N$ iff $\forall C[\,]\ C[M] \longrightarrow^* hnf$ *implies* $C[N] \longrightarrow^* hnf$
- PCF [Plotkin]
  $M, N ::=$ typed λ-calculus + recursion + arithmetic
  $M \simeq_p N$ iff $\forall C[\,]\ C[M] \longrightarrow^* \underline{\mathtt{n}}$ *implies* $C[N] \longrightarrow^* \underline{\mathtt{n}}$
  sequentiality
- Algol
  $M, N ::=$ valid Algol programs
  $M \simeq_p N$ iff $\forall C[\,]\ C[M] \longrightarrow^* \underline{\mathtt{n}}$ *implies* $C[N] \longrightarrow^* \underline{\mathtt{n}}$
- etc

Concurrency

Plan

1. Define a calculus for concurrency
2. Define directly semantics equivalence,
   instead of providing a semantics function.
3. Define observation
4. Context lemma for congruences
   (to reduce the set of contexts to consider)

Unfortunately, there are 2 calculi:

1. CCS, A calculus of communicating systems, `[Milner, 80]`
2. π-calculus, Communicating and mobile systems: the π-calculus, `[Milner et al, 90]`

Fortunately, the π-calculus is strong to express interaction,
and is useful in security.

Non-determinism

- x is a global variable. At beginning, $x = 0$
- Consider:

  $S = [x := 1;]$
  $T = [x := 2;]$

  After $S \,||\, T$, then $x \in \{1, 2\}$

- Result is not unique.
- Concurrent programs are not described by functions,
  $\Rightarrow$ relations.

Input-output behaviour

- x is a global variable. At beginning, $x = 0$
- Consider

  $S = [x := 1]$
  $T = [x := 0; x := x + 1]$

  $[\![S]\!]$ and $[\![T]\!]$ same functions on memory state.

- $S \,||\, S$ and $T \,||\, S$ are different relations on memory state.
  $\Rightarrow$ $[\![S]\!] \neq [\![T]\!]$ in any compositional semantics

- Conclusion: Interaction is important.

Atomicity

- x is a global variable. At beginning, $x = 0$
- Consider

  $S = [x := x + 1 \,||\, x := x + 1]$
  After S, then $x = 2$.

- However if

  $[x := x + 1]$ compiled into $[A := x + 1; x := A]$

- Then
  $S = [A := x + 1; x := A] \,||\, [B := x + 1; x := B]$
  After S, then $x \in \{1, 2\}$.

- Conclusion: define atomicity

Interaction

- A process is an atomic action, followed by a process. Ie.

$$\mathcal{P} \simeq \mathit{Null} + 2^{\mathit{action} \times \mathcal{P}}$$

  Is this equation meaningful?

- Answer: Scott's domains, denotational semantics. Remarkable and difficult theory of [Plotkin, 1976] (powerdomains for Scott's domains).
- Too difficult theory

Termination

- Concurrent processes are often non terminating.
- An operating system never terminates; same for the software of a vending machine, or a traffic-light controler, or a human, etc.
- Atomic steps usually terminate.

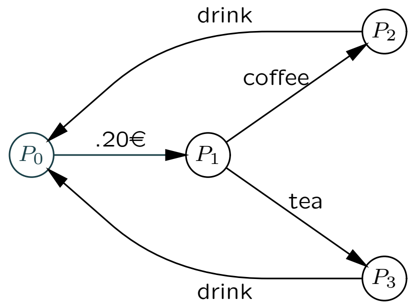
Transition Graphs

A transition graph is a triple $(\mathcal{P}, \mathcal{A}ct, \mathcal{T})$ where

- $\mathcal{P}$ is the set of processes
- $\mathcal{A}ct$ is the set of (atomic) actions
- $\mathcal{T} \subseteq \mathcal{P} \times \mathcal{A}ct \times \mathcal{P}$ is the transition relation

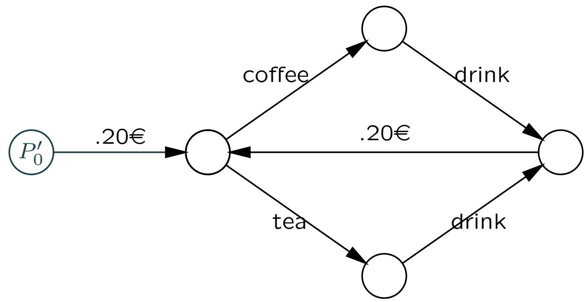
Example (1/3)

A vending machine for coffee/tea. At beginning, P_0



Example (2/3)

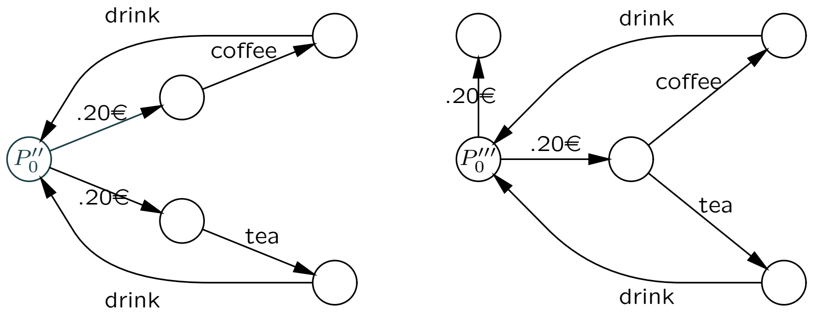
A different vending machine for coffee/tea. At beginning, P'_0



Is this graph equivalent to previous one?

Example (3/3)

Two new vending machines P_0'' and P_0'''



Why these graphs are not equivalent to previous ones?

CCS

CCS (1/2)

P, Q	$::=$			process
		$\sum_{i \in I} \alpha_i . P_i$	I finite set	guarded sum
		$P \mid Q$		composition
		$(\nu a) P$		restriction
		$A\langle a_1, a_2, \ldots a_n \rangle$	$n \geq 0$	function call
0	$=$	$\sum_{i \in \emptyset} P_i$		
α	$::=$	$a \mid \overline{a}$		guard
$\overline{\overline{a}}$	$=$	a		
		$A\langle x_1, x_2, \ldots x_n \rangle \overset{\text{def}}{=} P$		function definition
				$\{x_1, x_2, \ldots x_n\} = fn(P)$
$C[\,]$	$::=$	$[\,] \mid \alpha . C[\,] + M \mid (\nu a) C[\,] \mid P \mid C[\,] \mid C[\,] \mid Q$		context

Process α abbreviates process $\alpha.0$

CCS (2/2)

$P_0\langle\rangle \overset{\text{def}}{=} coin.(coffee.\overline{drink}.P_0\langle\rangle + tea.\overline{drink}.P_0\langle\rangle)$

or simply

$P_0 \overset{\text{def}}{=} coin.(coffee.\overline{drink}.P_0 + tea.\overline{drink}.P_0)$

$P_0' \overset{\text{def}}{=} coin.P_1' \qquad P_1' \overset{\text{def}}{=} coffee.\overline{drink}.P_2' + tea.\overline{drink}.P_2'$

$P_2' \overset{\text{def}}{=} coin.P_0'$

$P_0''' \overset{\text{def}}{=} coin.(coffee.\overline{drink}.P_0 + tea.\overline{drink}.P_0) + coin.0$

$Drinker \overset{\text{def}}{=} \overline{coin}.\overline{coffee}.drink.\overline{coin}.\overline{tea}.drink.0$

$Drinker \mid P_0$

$Drinker \mid P_0'$

$Drinker \mid P_0''$

Structural equivalence

- monoid laws

$$
\begin{array}{ll}
P + Q \equiv Q + P & P \mid Q \equiv Q \mid P \\
P + (Q + R) \equiv (P + Q) + R & P \mid (Q \mid R) \equiv (P \mid Q) \mid R \\
P + 0 \equiv P & P \mid 0 \equiv P
\end{array}
$$

- $A\langle y_1, y_2, \ldots y_n\rangle \equiv P[y_1/x_1, y_2/x_2, \ldots y_n/x_n]$
  when $A\langle x_1, x_2, \ldots x_n\rangle \stackrel{\text{def}}{=} P$
- congruence: $P \equiv Q \quad \Rightarrow \quad C[P] \equiv C[Q]$
- scope extrusion: $(\nu a)P \mid Q \equiv (\nu a)(P \mid Q)$ when $a \notin fn(Q)$
- $(\nu a)(\nu b)P \equiv (\nu b)(\nu a)P$
- $(\nu a)0 \equiv 0$
- α-renaming

Reduction rules (1/2)

[React] $(a.P + M) \mid (\overline{a}.Q + N) \longrightarrow P \mid Q$

[Par] $\dfrac{P \longrightarrow P'}{P \mid Q \longrightarrow P' \mid Q}$ [Res] $\dfrac{P \longrightarrow P'}{(\nu a)P \longrightarrow (\nu a)P'}$

[Struct] $\dfrac{P \equiv\longrightarrow\equiv Q}{P \longrightarrow Q}$

Reduction rules (2/2)

$P_0 \stackrel{\text{def}}{=} coin.(coffee.\overline{drink}.P_0 + tea.\overline{drink}.P_0)$

$Drinker \stackrel{\text{def}}{=} \overline{coin}.\overline{coffee}.drink.\overline{coin}.\overline{tea}.drink.0$

$P_0 \mid Drinker$

$\equiv$

$(coin.(coffee.\overline{drink}.P_0 + tea.\overline{drink}.P_0)) \mid (\overline{coin}.\overline{coffee}.drink.\overline{coin}.\overline{tea}.drink.0)$

$\longrightarrow$

$(coffee.\overline{drink}.P_0 + tea.\overline{drink}.P_0) \mid \overline{coffee}.drink.\overline{coin}.\overline{tea}.drink.0$

$\longrightarrow$

$\overline{drink}.P_0 \mid drink.\overline{coin}.\overline{tea}.drink.0$

$\longrightarrow$

$P_0 \mid \overline{coin}.\overline{tea}.drink.0$

Semantic equivalences

- $\mathcal{R}$ is a congruence:
  $P \mathrel{\mathcal{R}} Q \quad \Rightarrow \quad C[P] \mathrel{\mathcal{R}} C[Q]$
- preserving observation on any α:
  $P \mathrel{\mathcal{R}} Q \quad \Rightarrow \quad (P \downarrow \alpha \Leftrightarrow Q \downarrow \alpha)$
  where
  Definition 1 [barb] $P \downarrow \alpha$ iff $P \equiv (\nu\widetilde{\beta})(\alpha.Q + M \mid S)$ where $\alpha \notin \widetilde{\beta}$
  Definition 2 [weak barb] $P \Downarrow \alpha$ iff $P \longrightarrow^* Q \downarrow \alpha$
- preserving choices (branching time):
  $P \mathrel{\mathcal{R}} Q \wedge P \longrightarrow P' \quad \Rightarrow \quad \exists Q'$ s.t. $Q \longrightarrow Q' \wedge P' \mathrel{\mathcal{R}} Q'$
  $P \mathrel{\mathcal{R}} Q \wedge Q \longrightarrow Q' \quad \Rightarrow \quad \exists P'$ s.t. $Q \longrightarrow Q' \wedge P' \mathrel{\mathcal{R}} Q'$
  Such a relation is named a bisimulation

Many recursive definitions. In which order? Are there well-founded?
`[Park,Milner]` defined bisimulations as maximal fixpoints.
`[Fournet,Gonthier]` proved order is irrelevant.

Labelled Transition Systems

Reducing contexts ($\sim$ critical pairs in TRS):

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

[Sum1] $\dfrac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'}$ [Sum2] $\dfrac{Q \xrightarrow{\alpha} Q'}{P + Q \xrightarrow{\alpha} Q'}$

[Com] $\dfrac{P \xrightarrow{a} P' \quad Q \xrightarrow{\overline{a}} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$ [Par1] $\dfrac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}$ [Par2] $\dfrac{Q \xrightarrow{\alpha} Q'}{P \mid Q \xrightarrow{\alpha} P \mid Q'}$

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

[Rec] $\dfrac{P[\vec{a}/\vec{x}] \xrightarrow{\alpha} P' \quad A\langle\vec{x}\rangle \stackrel{\text{def}}{=} P}{A\langle\vec{a}\rangle \xrightarrow{\alpha} P'}$

Proposition 3 $P \xrightarrow{\tau}\equiv Q$ iff $P \longrightarrow Q$

Proposition 4 $P \equiv\xrightarrow{\alpha} Q$ implies $P \xrightarrow{\alpha}\equiv Q$

Proposition 5 $P \xrightarrow{\alpha} Q$ iff $P \downarrow \alpha \quad (\alpha \neq \tau)$

Strong bisimulation (1/4)

Definition 6 P strongly bisimilar to Q (we write $P \sim Q$) if whenever

- $P \xrightarrow{\alpha} P'$, there is Q' such that $Q \xrightarrow{\alpha} Q'$ and $P' \sim Q'$.
- $Q \xrightarrow{\alpha} Q'$, there is P' such that $P \xrightarrow{\alpha} P'$ and $P' \sim Q'$.

Graphically,

$$\begin{tikzcd} P \arrow[r, "\alpha"] \arrow[d, dash, "\mathcal{R}"'] & P' \arrow[d, dotted, dash, "\mathcal{R}"] \\ Q \arrow[r, dotted, "\alpha"] & Q' \end{tikzcd}$$

Exercise 1 Give intuition for $P_0 \lesssim P_0''' \lesssim P_0$

Exercise 2 Give intuition for $P_0 \sim P_0'$, $P_0 \not\sim P_0''$, $P_0 \not\sim P_0'''$

($\lesssim$ is strong simulation, i.e. half of strong bisimulation)

Exercise 3 Show that $(\nu a)(P + M) \sim (\nu a)P + (\nu a)M$.

Exercise 4 Show that $(\nu a)(P \mid Q) \not\sim (\nu a)P \mid (\nu a)M$.

Strong bisimulation (2/4)

Proposition 7 Strong bisimulation is a congruence

$$P \sim Q \quad \Rightarrow \quad C[P] \sim C[Q]$$

So $\sim$ is a semantics for $\downarrow \alpha$ (strong observation)

Exercise 5 (difficult) Show that it is the semantics induced by strong observation.

How to prove previous proposition ?

Typical (co-inductive) proof about bisimulation:

We want to show $P \sim Q$.
As $\sim$ is a maximal fixpoint,
$\sim$ is the the largest relation $\mathcal{R}$
satisfying the fixpoint equations of definition 5;
find $\mathcal{R}$ such that $P \mathrel{\mathcal{R}} Q$
show it satisfies the fixpoint equations of definition 5,
we say "we show that $\mathcal{R}$ is a bisimulation".

Strong bisimulation (3/4)

Proof of previous proposition.

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

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

CCS and strong bisimulation (4/4)

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 ...

Exercise 6 Give full proof of theorem.

Weak bisimulation (1/2)

Only visible actions are interesting $\Rightarrow$ Skip internal moves $\xrightarrow{\tau}$

Definition 8 $P \overset{\alpha}{\Longrightarrow} Q$ iff $P \longrightarrow^* \xrightarrow{\alpha_1} \longrightarrow^* \xrightarrow{\alpha_2} \cdots \longrightarrow^* \xrightarrow{\alpha_n} \longrightarrow^* Q \quad (n \geq 0)$
and $\alpha = \alpha_1 \alpha_2 \cdots \alpha_n$.

Definition 9 $\widehat{\alpha}$ is α where τ has been eliminitated.

Definition 10 P weakly bisimilar to Q (we write $P \approx Q$) if whenever

- $P \xrightarrow{\alpha} P'$, there is Q' such that $Q \overset{\widehat{\alpha}}{\Longrightarrow} Q'$ and $P' \approx Q'$.
- $Q \xrightarrow{\alpha} Q'$, there is P' such that $P \overset{\widehat{\alpha}}{\Longrightarrow} P'$ and $P' \approx Q'$.

Nearly a congruence, except for $+$ (partial commitment problem).

Definition 11 [observation-congruence] P observation-congruent to Q (we write $P \cong Q$) if, for any $\alpha \in \mathcal{A}ct$, whenever

- $P \xrightarrow{\alpha} P'$, there is Q' such that $Q \overset{\alpha}{\Longrightarrow} Q'$ and $P' \approx Q'$.
- $Q \xrightarrow{\alpha} Q'$, there is P' such that $P \overset{\alpha}{\Longrightarrow} P'$ and $P' \approx Q'$.

(differs from weak bisimulation in first step)

Weak bisimulation (2/2)

Exercise 7 Show that $\cong$ is the semantics induced by observation of weak barbs $\Downarrow \alpha$.

Conclusion

- axiomatization of (weak) bisimulations
- algorithms to compute bisimulations
- model checkers for bisimulations
- temporal logic: Hennessy-Milner logic
- missing reconfigurable networks of processes

$\Rightarrow$ the π-calculus