Formale Spezifikation und Verifikation

Mirco Tribastone

Institut für Informatik
Ludwig-Maximilians-Universität München
`tribastone@pst.ifi.lmu.de`

Process Algebras

Syntax and Semantics of Process Algebras

Acknowledgement

These slides are based on material kindly provided by:

Prof. Rocco De Nicola of the University of Florence, Italy

`http://www.dsi.unifi.it/~denicola/`

and by Dr. Roberto Bruni of the University of Pisa, Italy

`http://www.di.unipi.it/~bruni/`

Introduction

- Labelled transition systems have been proved a reasonable formal model of reactive systems as they capture:
  - Nondeterminism
  - Nonterminating behaviour
  - Concurrency
- However, this approach may become tedious and error-prone for nontrivial models.
- For instance, write down the labelled transition system if the condition $x < 2$ and $y < 2$ is replaced by $x < 20$ and $y < 20$... To some extent, a process algebra may be seen as a higher-level, compact representation of an LTS.

$x, y \leftarrow 0$

thread 1 do
  while $x < 2$ and $y < 2$ **do**
    $x \leftarrow x + 1$
  end while
end thread

thread 2 do
  while $x < 2$ and $y < 2$ **do**
    $y \leftarrow y + 1$
  end while
end thread

Intuitions Behind Process Algebra

- Process algebras formalize communication and computation in concurrent programs
- A program is modelled as a process that evolves through a sequence of (discrete) states by performing actions
- An action is the atomic unit of computation
- A model description is algebraic in that complex processes may be expressed as the result of operations over simpler ones.

In This Course

- A formal syntax defines the set of well-formed processes that can be defined in the calculus
- A structured operational semantics gives the meaning as a labelled transition system

Process Algebra Operators

Processes are composed via a number operators:

- Basic processes
- Action prefixing
- Sequential composition
- Choice (nondeterminism)
- Composition and interaction (parallelism)
- Abstraction (interaction delimiters)
- Recursion (infinite behaviour)

There are two kinds of atomic actions:

- Visible actions
- Internal (hidden) actions

Process Algebras and Labelled Transition Systems

Process Algebras and Labelled Transition Systems

- A formal syntax defines the set of well formed processes (a.k.a. terms).

Process Algebras and Labelled Transition Systems

- A formal syntax defines the set of well formed processes (a.k.a. terms).
- A structured operational semantics over the syntax induces a labelled transition system for a term.

Process Algebras and Labelled Transition Systems

- A formal syntax defines the set of well formed processes (a.k.a. terms).
- A structured operational semantics over the syntax induces a labelled transition system for a term.
- Transitions will be of kind

$$P \xrightarrow{a} P' ,$$

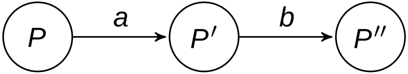
where P and P' are some well formed terms and a is some action (either visible or hidden).

Process Algebras and Labelled Transition Systems

- A formal syntax defines the set of well formed processes (a.k.a. terms).
- A structured operational semantics over the syntax induces a labelled transition system for a term.
- Transitions will be of kind

$$P \xrightarrow{a} P' \,,$$

  where P and P' are some well formed terms and a is some action (either visible or hidden).
- For instance, a process P that performs actions a and b in sequence, and then stops, will have the following LTS:



Basic Processes: Deadlock

The Inactive Process

It is usually denoted by one of the following terms:

- **0**
- *nil*
- *stop*

The semantics of this process is that there is no rule to define its transition: this process has no transition at all.

Basic Processes: Deadlock

The Inactive Process

It is usually denoted by one of the following terms:

- **0**
- *nil*
- *stop*

The semantics of this process is that there is no rule to define its transition: this process has no transition at all.

The Simplest Process Algebra Model

0

- A program that has terminated
- A teller that has left work for the day
- A vending machine that does not accept coins

Basic Processes: Termination

Termination is sometimes denoted by the term

- **1**
- *skip*
- *exit*
- $\surd$

It can only perform the special action $\surd$ (”tick”) to indicate termination and become **0**:

$$\frac{}{\mathbf{1} \xrightarrow{\surd} \mathbf{0}}$$

Basic Processes: Termination

Termination is sometimes denoted by the term

- $\mathbf{1}$
- *skip*
- *exit*
- $\surd$

It can only perform the special action $\surd$ (”tick”) to indicate termination and become **0**:

$$\frac{}{\mathbf{1} \xrightarrow{\surd} \mathbf{0}}$$

A Gentle Broken Vending Machine

$$\mathbf{1}$$

Does not accept coins but says that everything is ok.

Atomic Actions

Actions as Basic Processes

Sometimes, actions may be taken as basic processes themselves:

$$\frac{}{a \xrightarrow{a} \mathbf{0}}$$

When termination ticks are needed, a different variant can be used

$$\frac{}{a \xrightarrow{a} \surd}$$

A Dishonest Vending Machine: Accepts a Coin and Stops

$$coin$$

Sequential Behaviour

Sequential Behaviour

Prefixing

For each action μ there is a unary operator $\mu.$ such that given a process E, $\mu.E$ is a process that performs action μ and then behaves like E.

$$\frac{}{\mu.E \xrightarrow{\mu} E}$$

Sequential Behaviour

Prefixing

For each action μ there is a unary operator $\mu.$ such that given a process E, $\mu.E$ is a process that performs action μ and then behaves like E.

$$\frac{}{\mu.E \xrightarrow{\mu} E}$$

A One-Shot Vending Machine

$$\mathit{coin}.\mathit{choc}.\mathbf{0}$$

Accepts a coin and gives a chocolate (and then stops).

Sequential Behaviour

Prefixing

For each action μ there is a unary operator $\mu.$ such that given a process E, $\mu.E$ is a process that performs action μ and then behaves like E.

$$\overline{\mu.E \xrightarrow{\mu} E}$$

A One-Shot Vending Machine

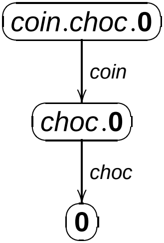
$$coin.choc.\mathbf{0}$$

Accepts a coin and gives a chocolate (and then stops).

Sometimes, for brevity, trailing zeros are omitted i.e., $P.\mathbf{0} \equiv P$ (e.g., $a.0 \equiv a$).

Labelled Transition System of $coin.choc.\mathbf{0}$

The labelled transition system



is obtained using the inference rule

$$\frac{}{\mu.E \xrightarrow{\mu} E}$$

and applying it for

$$\mu \equiv coin \quad \text{and} \quad E \equiv choc.\mathbf{0} \,,$$

and

$$\mu \equiv choc \quad \text{and} \quad E \equiv \mathbf{0} \,.$$

Choice - 1

Nondeterministic Choice

$$\frac{E \xrightarrow{\mu} E'}{E + F \xrightarrow{\mu} E'} \qquad \frac{F \xrightarrow{\mu} F'}{E + F \xrightarrow{\mu} F'}$$

Choice - 1

Nondeterministic Choice

$$\frac{E \xrightarrow{\mu} E'}{E + F \xrightarrow{\mu} E'} \qquad \frac{F \xrightarrow{\mu} F'}{E + F \xrightarrow{\mu} F'}$$

A Subtle Difference

Insert a coin, then make a choice of product:

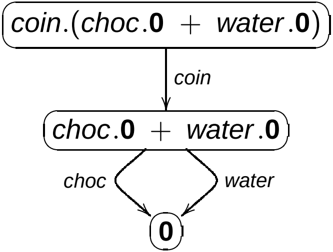
$$\mathit{coin}.(\mathit{choc}.\mathbf{0} \; + \; \mathit{water}.\mathbf{0})$$

Make a choice upon inserting the coin:

$$\mathit{coin}.\mathit{choc}.\mathbf{0} \;\; + \;\; \mathit{coin}.\mathit{water}.\mathbf{0}$$

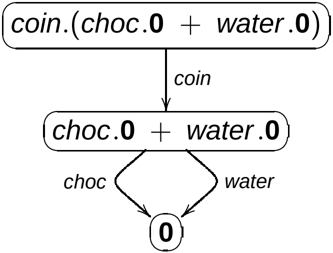
Subtly Different Labelled Transition Systems

LTS for $\mathit{coin}.(\mathit{choc}.\mathbf{0} + \mathit{water}.\mathbf{0})$

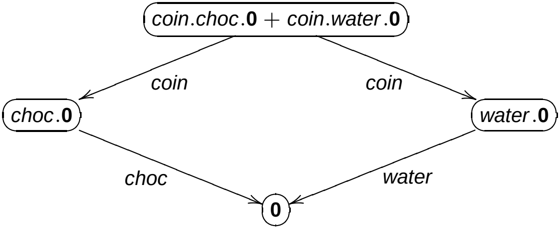


Subtly Different Labelled Transition Systems

LTS for $coin.(choc.\mathbf{0} \ + \ water.\mathbf{0})$

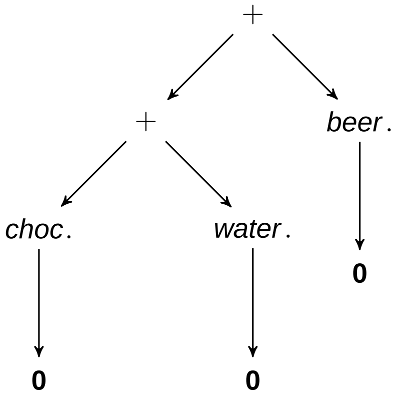


LTS for $coin.choc.\mathbf{0} + coin.water.\mathbf{0}$:



Algorithmic Inference

$$choc.\mathbf{0} + water.\mathbf{0} + beer.\mathbf{0}$$



Choice - 2

Internal Choice

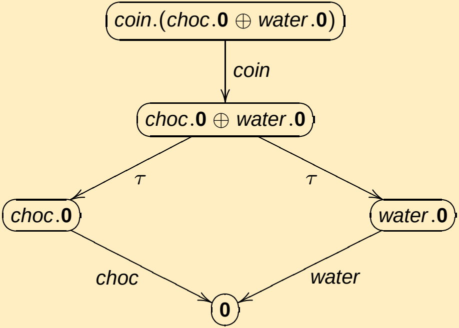
$$\frac{}{E \oplus F \xrightarrow{\tau} E} \qquad \frac{}{E \oplus F \xrightarrow{\tau} F}$$

Choice - 2

Internal Choice

$$\frac{}{E \oplus F \xrightarrow{\tau} E} \qquad \frac{}{E \oplus F \xrightarrow{\tau} F}$$

$coin.(choc.\mathbf{0} \oplus water.\mathbf{0})$



Choice - 3

External Choice

$$\frac{E \xrightarrow{\alpha} E'}{E \,\square\, F \xrightarrow{\alpha} E'} (\alpha \neq \tau) \qquad \frac{F \xrightarrow{\alpha} F'}{E \,\square\, F \xrightarrow{\alpha} F'} (\alpha \neq \tau)$$

$$\frac{E \xrightarrow{\tau} E'}{E \,\square\, F \xrightarrow{\tau} E' \,\square\, F} \qquad \frac{F \xrightarrow{\tau} F'}{E \,\square\, F \xrightarrow{\tau} E \,\square\, F'}$$

Choice - 3

External Choice

$$\frac{E \xrightarrow{\alpha} E'}{E \,\square\, F \xrightarrow{\alpha} E'} (\alpha \neq \tau) \qquad \frac{F \xrightarrow{\alpha} F'}{E \,\square\, F \xrightarrow{\alpha} F'} (\alpha \neq \tau)$$

$$\frac{E \xrightarrow{\tau} E'}{E \,\square\, F \xrightarrow{\tau} E' \,\square\, F} \qquad \frac{F \xrightarrow{\tau} F'}{E \,\square\, F \xrightarrow{\tau} E \,\square\, F'}$$

Mix of Choices

$$\mathit{coin}.\big((\mathit{choc}.\mathbf{0} \;\oplus\; \mathit{water}.\mathbf{0}) \;\square\; \mathit{water}.\mathbf{0}\big)$$

Different Transitions

External Choice

$$coin.\big((choc.\mathbf{0} \oplus water.\mathbf{0}) \ \Box \ water.\mathbf{0}\big) \xrightarrow{coin}$$
$$(choc.\mathbf{0} \oplus water.\mathbf{0}) \ \Box \ water.\mathbf{0} \xrightarrow{\tau}$$
$$(choc.\mathbf{0} \ \Box \ water.\mathbf{0})$$

Different Transitions

External Choice

$$coin.((choc.\mathbf{0} \oplus water.\mathbf{0}) \;\Box\; water.\mathbf{0}) \xrightarrow{coin}$$

$$(choc.\mathbf{0} \oplus water.\mathbf{0}) \;\Box\; water.\mathbf{0} \xrightarrow{\tau}$$

$$(choc.\mathbf{0} \;\Box\; water.\mathbf{0})$$

Internal Choice

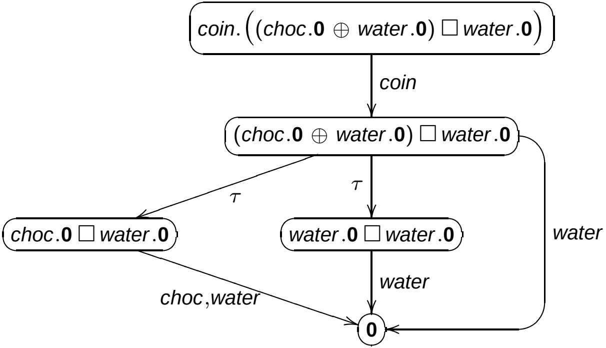
$$coin.((choc.\mathbf{0} \oplus water.\mathbf{0}) \oplus water.\mathbf{0}) \xrightarrow{coin}$$

$$(choc.\mathbf{0} \oplus water.\mathbf{0}) \oplus water.\mathbf{0} \xrightarrow{\tau}$$

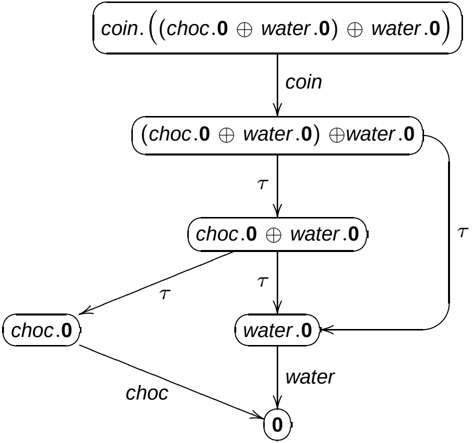
$$choc.\mathbf{0} \oplus water.\mathbf{0} \xrightarrow{\tau}$$

$$choc.\mathbf{0}$$

Labelled Transition System of $coin.\big((choc.\mathbf{0} \oplus water.\mathbf{0}) \;\square\; water.\mathbf{0}\big)$



Labelled Transition System of $coin.((choc.\mathbf{0} \oplus water.\mathbf{0}) \oplus water.\mathbf{0})$



Parallel Composition – 1

Interleaving

$$\frac{E \xrightarrow{\mu} E'}{E \parallel F \xrightarrow{\mu} E' \parallel F} \qquad \frac{F \xrightarrow{\mu} F'}{E \parallel F \xrightarrow{\mu} E \parallel F'}$$

Parallel Composition – 1

Interleaving

$$\frac{E \xrightarrow{\mu} E'}{E \parallel F \xrightarrow{\mu} E' \parallel F} \qquad \frac{F \xrightarrow{\mu} F'}{E \parallel F \xrightarrow{\mu} E \parallel F'}$$

Anagrams as Traces

Draw the LTS for

$$a.\mathbf{0} \parallel e.\mathbf{0} \parallel m.\mathbf{0} \parallel r.\mathbf{0}$$

Parallel Composition – 2

We assume there is a *co-action* $\overline{\alpha}$ for each visible α, and let $\overline{\overline{\alpha}} = \alpha$.

Milner's Parallel (Two-Party Synchronization)

$$\frac{E \xrightarrow{\mu} E'}{E \mid F \xrightarrow{\mu} E' \mid F} \qquad \frac{F \xrightarrow{\mu} F'}{E \mid F \xrightarrow{\mu} E \mid F'} \qquad \frac{E \xrightarrow{\alpha} E' \qquad F \xrightarrow{\overline{\alpha}} F'}{E \mid F \xrightarrow{\tau} E' \mid F'} (\alpha \neq \tau)$$

Parallel Composition – 2

We assume there is a *co-action* $\overline{\alpha}$ for each visible α, and let $\overline{\overline{\alpha}} = \alpha$.

Milner's Parallel (Two-Party Synchronization)

$$\frac{E \xrightarrow{\mu} E'}{E \mid F \xrightarrow{\mu} E' \mid F} \qquad \frac{F \xrightarrow{\mu} F'}{E \mid F \xrightarrow{\mu} E \mid F'} \qquad \frac{E \xrightarrow{\alpha} E' \qquad F \xrightarrow{\overline{\alpha}} F'}{E \mid F \xrightarrow{\tau} E' \mid F'} (\alpha \neq \tau)$$

User-Machine Interaction

$$\big(coin.(\overline{choc}.\mathbf{0} \oplus \overline{water}.\mathbf{0})\big) \mid (\overline{coin}.choc.\mathbf{0})$$

Different Interactions

Appropriate Interaction

$$\big(coin.(\overline{choc}.\mathbf{0} \oplus \overline{water}.\mathbf{0})\big) \mid (\overline{coin}.choc.\mathbf{0}) \xrightarrow{\tau}$$
$$(\overline{choc}.\mathbf{0} \oplus \overline{water}.\mathbf{0}) \mid (choc.\mathbf{0}) \xrightarrow{\tau}$$
$$(\overline{choc}.\mathbf{0}\ ) \mid (choc.\mathbf{0}) \xrightarrow{\tau}$$
$$\mathbf{0} \mid \mathbf{0}$$

Different Interactions

Appropriate Interaction

$$\big(coin.(\overline{choc}.\mathbf{0} \;\oplus\; \overline{water}.\mathbf{0})\big) \mid (\overline{coin}.choc.\mathbf{0}) \xrightarrow{\tau}$$

$$(\overline{choc}.\mathbf{0} \;\oplus\; \overline{water}.\mathbf{0}) \mid (choc.\mathbf{0}) \xrightarrow{\tau}$$

$$(\overline{choc}.\mathbf{0}\ ) \mid (choc.\mathbf{0}) \xrightarrow{\tau}$$

$$\mathbf{0} \mid \mathbf{0}$$

Inappropriate Interaction — Coin Thrown Away

$$\big(coin.(\overline{choc}.\mathbf{0} \;\oplus\; \overline{water}.\mathbf{0})\big) \mid (\overline{coin}.choc.\mathbf{0}) \xrightarrow{\tau}$$

$$(\overline{choc}.\mathbf{0} \;\oplus\; \overline{water}.\mathbf{0}) \mid (choc.\mathbf{0}) \xrightarrow{\tau}$$

$$(\overline{water}.\mathbf{0}) \mid (choc.\mathbf{0})$$

Parallel Composition – 3

Merge Operator with Synchronization Function

$$\frac{E \xrightarrow{\mu} E'}{E \parallel F \xrightarrow{\mu} E' \parallel F} \qquad \frac{F \xrightarrow{\mu} F'}{E \parallel F \xrightarrow{\mu} E \parallel F'} \qquad \frac{E \xrightarrow{a} E' \qquad F \xrightarrow{b} F'}{E \parallel F \xrightarrow{\gamma(a,b)} E' \parallel F'}$$

with $\mu \in \Lambda \cup \{\tau\}$

Parallel Composition – 3

Merge Operator with Synchronization Function

$$\frac{E \xrightarrow{\mu} E'}{E \parallel F \xrightarrow{\mu} E' \parallel F} \qquad \frac{F \xrightarrow{\mu} F'}{E \parallel F \xrightarrow{\mu} E \parallel F'} \qquad \frac{E \xrightarrow{a} E' \qquad F \xrightarrow{b} F'}{E \parallel F \xrightarrow{\gamma(a,b)} E' \parallel F'}$$

with $\mu \in \Lambda \cup \{\tau\}$

Another Interaction

$$\mathit{getCoin}.(\mathit{giveChoc}.\mathbf{0} + \mathit{giveWater}.\mathbf{0}) \parallel \mathit{putCoin}.\mathit{getChoc}.\mathbf{0}\ ,$$

with $\gamma(\mathit{getCoin}, \mathit{putCoin}) = \mathbf{ok}$ and $\gamma(\mathit{giveChoc}, \mathit{getChoc}) = \mathbf{ok}$.

Parallel Composition – 4

Communication Merge

$$\frac{E \xrightarrow{a} E' \qquad F \xrightarrow{b} F'}{E \,|_c\, F \xrightarrow{\gamma(a,b)} E' \parallel F'}$$

Parallel Composition – 4

Communication Merge

$$\frac{E \xrightarrow{a} E' \qquad F \xrightarrow{b} F'}{E \,|_c\, F \xrightarrow{\gamma(a,b)} E' \parallel F'}$$

How is $E \,|_c\, F$ different with respect to $E \parallel F$?

Parallel Composition – 4

Communication Merge

$$\frac{E \xrightarrow{a} E' \qquad F \xrightarrow{b} F'}{E \,|_c\, F \xrightarrow{\gamma(a,b)} E' \parallel F'}$$

How is $E \,|_c\, F$ different with respect to $E \parallel F$?

Left Merge

$$\frac{E \xrightarrow{\mu} E'}{E \,⌊\!\lfloor\, F \xrightarrow{\mu} E' \parallel F}$$

Parallel Composition - 5

Hoare's Parallel (Multi-Party Synchronization)

$$\frac{E \xrightarrow{\mu} E'}{E \;\|[L]\|\; F \xrightarrow{\mu} E' \;\|[L]\|\; F}\,(\mu \notin L) \qquad \frac{F \xrightarrow{\mu} F'}{E \;\|[L]\|\; F \xrightarrow{\mu} E \;\|[L]\|\; F'}\,(\mu \notin L)$$

$$\frac{E \xrightarrow{\alpha} E' \qquad F \xrightarrow{\alpha} F'}{E \;\|[L]\|\; F \xrightarrow{\alpha} E' \;\|[L]\|\; F'}\,(\alpha \in L)$$

Parallel Composition - 5

Hoare's Parallel (Multi-Party Synchronization)

$$\frac{E \xrightarrow{\mu} E'}{E \;\|[L]\|\; F \xrightarrow{\mu} E' \;\|[L]\|\; F} \,(\mu \notin L) \qquad \frac{F \xrightarrow{\mu} F'}{E \;\|[L]\|\; F \xrightarrow{\mu} E \;\|[L]\|\; F'} \,(\mu \notin L)$$

$$\frac{E \xrightarrow{\alpha} E' \qquad F \xrightarrow{\alpha} F'}{E \;\|[L]\|\; F \xrightarrow{\alpha} E' \;\|[L]\|\; F'} \,(\alpha \in L)$$

- The operators $\cdot \;\|[\emptyset]\|\; \cdot$ and $\cdot \;\|\; \cdot$ are equivalent.
- Let us compare Milner's and Hoare's synchronisation operators…

Abstraction – 1

Restriction

$$\frac{E \xrightarrow{\alpha} E'}{E \setminus L \xrightarrow{\alpha} E' \setminus L} \ (\alpha\,,\ \overline{\alpha} \notin L)$$

Abstraction – 1

Restriction

$$\frac{E \xrightarrow{\alpha} E'}{E \setminus L \xrightarrow{\alpha} E' \setminus L} \; (\alpha \, , \, \overline{\alpha} \, \notin \, L)$$

Forcing Interaction

$$\begin{aligned} \big(\,(coin.\overline{ok}.\mathbf{0}) \mid ok.(\overline{choc}.\mathbf{0} + \overline{water}.\mathbf{0})\,\big)\setminus\{ok\} \mid \overline{coin}.choc.\mathbf{0} &\xrightarrow{\tau} \\ \big(\,(\overline{ok}.\mathbf{0}) \mid ok.(\overline{choc}.\mathbf{0} + \overline{water}.\mathbf{0})\,\big)\setminus\{ok\} \mid choc.\mathbf{0} &\xrightarrow{\tau} \\ \big(\,\mathbf{0} \mid (\overline{choc}.\mathbf{0} + \overline{water}.\mathbf{0})\,\big)\setminus\{ok\} \mid choc.\mathbf{0} &\xrightarrow{\tau} \\ \big(\,\mathbf{0} \mid \mathbf{0}\,\big)\setminus\{ok\} \mid \mathbf{0} & \end{aligned}$$

A malicious user executing $\overline{ok}.choc.\mathbf{0}$ would be stopped.

Abstraction – 2

Hiding

$$\frac{E \xrightarrow{\alpha} E'}{E/L \xrightarrow{\alpha} E'/L} (\alpha \notin L) \qquad \frac{E \xrightarrow{\alpha} E'}{E/L \xrightarrow{\tau} E'/L} (\alpha \in L)$$

Avoiding Interaction

$$\big((coin.ok.\mathbf{0}) \ \lvert[ok]\rvert \ ok.(choc.\mathbf{0} + water.\mathbf{0}) \big)/\{ok\}$$

The *ok* signal is internalized thus it cannot be used by a dishonest user.

Abstraction – 3

Renaming

$$\frac{E \xrightarrow{\mu} E'}{E[f] \xrightarrow{f(\mu)} E'[f]}$$

Multilingual Interaction

An Italian user

$$\overline{moneta}.\, acqua.\, \mathbf{0}$$

can interact with an English-speaking machine by applying:

$$(\,\overline{moneta}.\, acqua.\, \mathbf{0}\,)\,[coin/moneta,\; water/acqua]$$

Infinite Behaviour – 1

Recursion

$$\frac{E\{rec\,X.E/X\} \xrightarrow{\mu} E'}{rec\,X.E \xrightarrow{\mu} E'}$$

Infinite Behaviour – 1

Recursion

$$\frac{E\{rec\,X.E/X\} \xrightarrow{\mu} E'}{rec\,X.E \xrightarrow{\mu} E'}$$

Example : $rec\,D.\,coin.\,(\overline{choc}.\,D \;+\; \overline{water}.\,D)$

Infinite Behaviour – 1

Recursion

$$\frac{E\{rec\, X.E/X\} \xrightarrow{\mu} E'}{rec\, X.E \xrightarrow{\mu} E'}$$

Example : $rec\, D.\, coin.\, (\overline{choc}.\, D \;+\; \overline{water}.\, D)$

$$rec\, D.\, coin.\, (\overline{choc}.\, D \;+\; \overline{water}.\, D) \quad \Big\} \quad \xrightarrow{coin}$$

$$\left.\begin{array}{l} \overline{choc}.\, rec\, D.\, coin.\, (\overline{choc}.\, D \;+\; \overline{water}.\, D) \\ + \\ \overline{water}.\, rec\, D.\, coin.\, (\overline{choc}.\, D \;+\; \overline{water}.\, D) \end{array}\right\} \quad \xrightarrow{\overline{choc}}$$

$$rec\, D.\, coin.\, (\overline{choc}.\, D \;+\; \overline{water}.\, D) \quad \Big\} \quad \xrightarrow{coin} \quad \ldots$$

Infinite Behaviour – 2

The notation $rec\,X.E$ for recursion makes the process expressions more difficult to parse and less pleasant to read.

Recursion Via Constant Declaration

A fixed set of constants can be used as some sort of procedure calls inside processes.

Let $\Gamma = \left\{X_1 \triangleq E_1, X_2 \triangleq E_2, \ldots, X_n \triangleq E_n\right\}$ be the set of definitions, then

$$\frac{X \triangleq E \in \Gamma \quad E \xrightarrow{\mu} E'}{X \xrightarrow{\mu} E'}$$

Long Lasting Vending Machine

$$D \triangleq coin.(\overline{choc}.D + \overline{water}.D)$$

Infinite Behaviour – 3

Replication

$$\frac{E \,|\, !E \xrightarrow{\mu} E'}{!E \xrightarrow{\mu} E'}$$

Chocolate *Ad Libitum*

$$!\,coin.\overline{choc}.\mathbf{0} \xrightarrow{coin}$$

$$\overline{choc}.\mathbf{0} \;|\; !\,coin.\overline{choc}.\mathbf{0} \xrightarrow{coin}$$

$$\overline{choc}.\mathbf{0} \;|\; \overline{choc}.\mathbf{0} \;|\; !\,coin.\overline{choc}.\mathbf{0} \xrightarrow{\overline{choc}}$$

$$\mathbf{0} \;|\; \overline{choc}.\mathbf{0} \;|\; !\,coin.\overline{choc}.\mathbf{0} \xrightarrow{\overline{choc}}$$

$$\mathbf{0} \;|\; \mathbf{0} \;|\; !\,coin.\overline{choc}.\mathbf{0}$$

Interaction with Value Passing

$$B \triangleq in(x).B(x)$$
$$B(x) \triangleq \overline{out}(x+1).B$$

- Assume x is a nonnegative integer
- Process B accepts an action parametrized with a value
- Then, it behaves as a process that outputs the input value incremented by one
- For instance,

$$B \xrightarrow{in(0)} B(0) \xrightarrow{\overline{out}(1)} B \xrightarrow{in(3)} B(3) \xrightarrow{\overline{out}(4)} B \rightarrow \ldots$$

Interaction with Value Passing

Interaction with Value Passing

Single Evolutions

$$\frac{}{a(x).E \xrightarrow{a(v)} E\{v/x\}} \ (v \text{ is a value})$$

$$\frac{}{\overline{a}(e).E \xrightarrow{\overline{a}(val(e))} E}$$

Interaction with Value Passing

Single Evolutions

$$\frac{}{a(x).E \xrightarrow{a(v)} E\{v/x\}} \ (v \text{ is a value}) \qquad \frac{}{\overline{a}(e).E \xrightarrow{\overline{a}(val(e))} E}$$

Constant

$$\frac{P\{v_1/x_1, v_2/x_2, \ldots, v_n/x_n\} \xrightarrow{\alpha} P'}{A(e_1, e_2, \ldots, e_n) \xrightarrow{\alpha} P'}, \qquad A(x_1, x_2, \ldots, x_n) \triangleq P$$

and each e_i evaluates to v_i .

Interaction with Value Passing

Single Evolutions

$$\frac{}{a(x).E \xrightarrow{a(v)} E\{v/x\}} \; (v \text{ is a value}) \qquad \frac{}{\overline{a}(e).E \xrightarrow{\overline{a}(val(e))} E}$$

Constant

$$\frac{P\{v_1/x_1, v_2/x_2, \ldots, v_n/x_n\} \xrightarrow{\alpha} P'}{A(e_1, e_2, \ldots, e_n) \xrightarrow{\alpha} P'}, \qquad A(x_1, x_2, \ldots, x_n) \triangleq P$$

and each e_i evaluates to v_i .

Interaction

$$\frac{E \xrightarrow{\overline{a}(v)} E' \quad F \xrightarrow{a(v)} F'}{E|F \xrightarrow{\tau} E'|F'} \qquad \frac{E \xrightarrow{a(v)} E' \quad F \xrightarrow{\overline{a}(v)} F'}{E|F \xrightarrow{\tau} E'|F'}$$

Conditional Execution

if-then-else construct

$$\frac{val(e) = \textbf{true} \quad E \xrightarrow{\mu} E'}{\textbf{if } e \textbf{ then } E \textbf{ else } F \xrightarrow{\mu} E'} \qquad \frac{val(e) = \textbf{false} \quad F \xrightarrow{\mu} F'}{\textbf{if } e \textbf{ then } E \textbf{ else } F \xrightarrow{\mu} F'}$$

Conditional Execution

if-then-else construct

$$\frac{val(e) = \textbf{true} \quad E \xrightarrow{\mu} E'}{\textbf{if } e \textbf{ then } E \textbf{ else } F \xrightarrow{\mu} E'} \qquad \frac{val(e) = \textbf{false} \quad F \xrightarrow{\mu} F'}{\textbf{if } e \textbf{ then } E \textbf{ else } F \xrightarrow{\mu} F'}$$

Let us consider a vending machine that accept 20 cents coins (or higher) and offers a chocolate:

$$coin(x).\,\textbf{if } x \geq 20 \textbf{ then } choc.\mathbf{0} \textbf{ else } \mathbf{0}$$

The user interacts with the machine as follows:

$$\begin{aligned} coin(x).\,\textbf{if } x \geq 20 \textbf{ then } \overline{choc}.\mathbf{0} \textbf{ else } \mathbf{0} \mid \overline{coin}\,40.choc.\mathbf{0} &\xrightarrow{\tau} \\ \textbf{if } 40 \geq 20 \textbf{ then } \overline{choc}.\mathbf{0} \textbf{ else } \mathbf{0} \mid choc.\mathbf{0} &\xrightarrow{\tau} \\ \mathbf{0} \mid \mathbf{0}\,. \end{aligned}$$

Pipelining

Pipeline

The binary operator for pipeline is denoted by $>$.
If E ed F are processes, process $E > F$ spawns a copy of F whenever E succeeds.

$$\frac{E \xrightarrow{\mu} E'}{E > F \xrightarrow{\mu} E' > F} \quad (\mu \neq \surd) \qquad\qquad \frac{E \xrightarrow{\surd} E'}{E > F \xrightarrow{\tau} (E' > F)|F}$$

Pipelining

Pipeline

The binary operator for pipeline is denoted by $>$.
If E ed F are processes, process $E > F$ spawns a copy of F whenever E succeeds.

$$\frac{E \xrightarrow{\mu} E'}{E > F \xrightarrow{\mu} E' > F} \quad (\mu \neq \surd) \qquad\qquad \frac{E \xrightarrow{\surd} E'}{E > F \xrightarrow{\tau} (E' > F) | F}$$

A Three-Shot Vending Machine

$$(coin.\mathbf{1} \mid coin.\mathbf{1} \mid coin.\mathbf{1}) > \; choc.\mathbf{0}$$

Interruption – 1

Disabling Operator

The disabling binary operator $[>$ permits to interrupt some actions when specific events happen.

$$\frac{E \xrightarrow{\mu} E'}{E \; [> F \xrightarrow{\mu} E' \; [> F} \quad (\mu \neq \surd) \qquad \frac{E \xrightarrow{\surd} E'}{E \; [> F \xrightarrow{\tau} E'} \qquad \frac{F \xrightarrow{\mu} F'}{E \; [> F \xrightarrow{\mu} F'}$$

Interruption – 1

Disabling Operator

The disabling binary operator $[>$ permits to interrupt some actions when specific events happen.

$$\frac{E \xrightarrow{\mu} E'}{E \; [> F \xrightarrow{\mu} E' \; [> F} \quad (\mu \neq \surd) \qquad \frac{E \xrightarrow{\surd} E'}{E \; [> F \xrightarrow{\tau} E'} \qquad \frac{F \xrightarrow{\mu} F'}{E \; [> F \xrightarrow{\mu} F'}$$

A Cheating Customer

$$(coin.choc.\mathbf{1}) \; [> \; (bang.choc.\mathbf{0})$$

This describes a vending machine that when“banged” gives away a chocolate without getting the coin

Interruption – 2

Try Catch

The try-catch operator

- $\texttt{try}\ \_\ \texttt{catch}(A)\ \_$

permits to handle specific events (maybe less natural to use in PA).

$$\frac{E \xrightarrow{\mu} E'}{\texttt{try}\ E\ \texttt{catch}(A)\ F \xrightarrow{\mu} \texttt{try}\ E'\ \texttt{catch}(A)\ F} \quad (\mu \notin A)$$

$$\frac{E \xrightarrow{\mu} E'}{\texttt{try}\ E\ \texttt{catch}(A)\ F \xrightarrow{\tau} F} \quad (\mu \in A)$$

$$\frac{E \xrightarrow{\surd} E'}{\texttt{try}\ E\ \texttt{catch}(A)\ F \xrightarrow{\surd} E'}$$