Operators for Process Algebras

Rocco De Nicola

Dipartimento di Sistemi ed Informatica
Università di Firenze

Process Algebras and Concurrent Systems

Internal and External Actions

An elementary action of a system represents the atomic (non-interruptible) abstract step of a computation that is performed by a system to move from one state to the other.

Actions represent various activities of concurrent systems:

1. Sending a message
2. Receiving a message
3. Updating values
4. Synchronizing with other processes
5. ...

We have two main types of atomic actions:

- Visible Actions
- Internal Actions

Operators for Processes Modelling

Processes are composed via a number of basic operators

1. Basic Processes
2. Action Prefixing
3. Sequentialization
4. Choice
5. Parallel Composition
6. Abstraction
7. Infinite Behaviours

Operational Semantics

To each process, built using the above mentioned operators, an LTS is associated by relying on structural induction to define the meaning of each operator.

Inference Systems

An inference system is a set of inference rule of the form

$$\frac{p_1, \cdots, p_n}{q}$$

Transition Rules

For each operator *op*, we have a number of rules of the form below, where $\{i_1, \cdots, i_m\} \subseteq \{1, \cdots, n\}$.

$$\frac{E_{i_1} \xrightarrow{\alpha_1} E'_{i_1} \quad \cdots \quad E_{i_m} \xrightarrow{\alpha_m} E'_{i_m}}{op(E_1, \cdots, E_n) \xrightarrow{\alpha} op(E'_1, \cdots, E'_n)}$$

The Elegance of Operational Semantics

Automata as terms

Few SOS rules define all the automata that can ever be specified with the chosen operators. Given any term, the rules are used the derive the corresponding automaton. The set of rules is fixed once and for all.

Structural induction

The interaction of complex systems is defined in terms of the behavior of their components.

A remark

The LTS is the least one satisfying the inference rules.

Rule induction

A property is true for the whole LTS if whenever it holds for the premises of each rule, it holds also for the conclusion.

Basic Processes

Inactive Process

Is usually denoted by

- *nil*
- 0
- *stop*

The semantics of this process is characterized by the fact that there is no rule to define its transition: it has no transition.

A broken vending machine

$$nil$$

Does not accept coins and does not give any drink.

Basic Processes ctd

Termination

Termination is sometimes denoted by

- *exit*
- *skip*

that can only perform the special action $\surd$ ("tick") to indicate termination and become *nil*

$$\frac{}{exit \xrightarrow{\surd} stop}$$

A gentle broken vending machine

$$exit$$

Does not accept coins, does not gives drinks but says that everything is ok.

Action Prefixing

Prefixing

For each action μ there is a unary operator

- $\mu.\cdot$
- $\mu \rightarrow \cdot$

that builds from process E a new process $\mu.E$ that performs action μ and then behaves like E.

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

A "one shot" vending machine

$$coin \rightarrow choc \rightarrow stop$$

Accepts a coin and gives a chocolate, then stops.

Action Prefixing ctd

Action as processes

Instead of prefixing, some calculi rely on considering actions as basic processes.

$$\frac{}{a \xrightarrow{a} stop}$$

A dishonest vending machine

$$coin$$

Accepts a coin and stops.

Sequential Composition

Sequentialization

The binary operator for sequential composition is denoted by

- $\cdot ; \cdot$
- $\cdot \gg \cdot$

If E ed F are processes, process $E; F$ executes E and then behaves like F

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

Another "one shot" vending machine

$$coin;\ choc$$

Sequential Composition ctd

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

$$(coin \rightarrow choc \rightarrow stop) \; [> \; (bang \rightarrow choc \rightarrow stop)$$

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

Choice - 1

Nondeterministic Choice

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

User's Choice

$$coin \rightarrow (choc \rightarrow stop \; + \; water \rightarrow stop)$$

Machine's Choice

$$coin \rightarrow choc \rightarrow stop \quad + \quad coin \rightarrow water \rightarrow stop$$

Choice - 2

Internal Choice

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

Machine's Choice

$$coin \rightarrow (choc \rightarrow stop \;\; \oplus \;\; water \rightarrow stop)$$

Choice - 3

External Choice

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

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

User's Choice

$$coin \rightarrow ((choc \rightarrow stop \;\oplus\; water \rightarrow stop) \;\Box\; water \rightarrow stop)$$

Different Transitions

External Choice

$$coin \rightarrow ((choc \rightarrow stop \ \oplus \ water \rightarrow stop) \ \square \ water \rightarrow stop)$$
$$\xrightarrow{coin}$$
$$(choc \rightarrow stop \ \oplus \ water \rightarrow stop) \ \square \ water \rightarrow stop$$
$$\xrightarrow{\tau}$$
$$(choc \rightarrow stop \ \square \ water \rightarrow stop)$$

Internal Choice

$$coin \rightarrow ((choc \rightarrow stop \ \oplus \ water \rightarrow stop) \ \oplus \ water \rightarrow stop)$$
$$\xrightarrow{coin}$$
$$(choc \rightarrow stop \ \oplus \ water \rightarrow stop) \ \oplus \ water \rightarrow stop$$
$$\xrightarrow{\tau}$$
$$choc \rightarrow stop \ \oplus \ water \rightarrow stop$$
$$\xrightarrow{\tau}$$
$$choc \rightarrow stop$$

Parallel Composition - 1

Milner's Parallel

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

User-Machine interaction

$$\big(coin \rightarrow (\overline{choc} \rightarrow stop \;\oplus\; \overline{water} \rightarrow stop)\big) \;\mid\; \big(\overline{coin} \rightarrow choc \rightarrow stop\big)$$

We can have different interactions

Appropriate Interaction

$$(coin \rightarrow (\overline{choc} \rightarrow stop \oplus \overline{water} \rightarrow stop)) \mid (\overline{coin} \rightarrow choc \rightarrow stop)$$

$$\xrightarrow{\tau}$$

$$(\overline{choc} \rightarrow stop \oplus \overline{water} \rightarrow stop) \mid (choc \rightarrow stop)$$

$$\xrightarrow{\tau}$$

$$(\overline{choc} \rightarrow stop\ ) \mid (choc \rightarrow stop)$$

$$\xrightarrow{\tau}$$

$$stop \mid stop$$

Inappropriate Interaction - Coin thrown away

$$(coin \rightarrow (\overline{choc} \rightarrow stop \oplus \overline{water} \rightarrow stop)) \mid (\overline{coin} \rightarrow choc \rightarrow stop)$$

$$\xrightarrow{\tau}$$

$$(\overline{choc} \rightarrow stop \oplus \overline{water} \rightarrow stop) \mid (choc \rightarrow stop)$$

$$\xrightarrow{\tau}$$

$$(\overline{water} \rightarrow stop) \mid (choc \rightarrow stop)$$

Parallel Composition - 2

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

$$getCoin.(giveChoc.nil \; + \; giveWater.nil) \; \parallel \; putCoin.getChoc.nil$$

with $\gamma(getCoin, putCoin) = ok$ e $\gamma(giveChoc, getChoc) = ok$.

Parallel Composition - 3

Communication Merge

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

Left Merge

$$\frac{E \xrightarrow{\mu} E'}{E \mathbin{|\!\lfloor} F \xrightarrow{\mu} E' \parallel F}$$

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 - 4

Hoare's Parallel

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

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

The operator $|[L]|$ is strongly related with some of the operators seen before.

1. $|[L]|$ and $\|$ are equivalent if $\gamma(a, a) = a, \; \forall a \in L$,
2. $|[L]|$ and $|||$ are equivalent if $L = \emptyset$,

Interaction via Synchronization Algebra

Most operators for parallel composition can be expressed in terms of suitable synchronization algebras (assume $E \xrightarrow{*} E$ for all E).

Definition

A Synchronization Algebra una 4-tuple $\langle \Lambda, *, 0, \bullet \rangle$ where

1. Λ is a set of labels containing the special labels $*$ e 0,
2. $\bullet$ is an associative and commutative binary operation over Λ (i.e. $\bullet : \Lambda \times \Lambda \rightarrow \Lambda$) that satisfies:
   1. $a \bullet 0 = 0$ for all $a \in \Lambda$,
   2. $* \bullet * = *$,
   3. $a \bullet b = *$ implies $a = b = *$, for all $a, b \in \Lambda$.

$$\frac{E \xrightarrow{\alpha} E' \quad F \xrightarrow{\beta} F'}{E \bullet F \xrightarrow{\alpha \bullet \beta} E' \bullet F'} \quad (\alpha \bullet \beta \neq 0)$$

$\bullet$	$*$	α	0
$*$	$*$	α	0
α	α	0	0
0	0	0	0

Interaction with Value Passing

Single Evolutions

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

$$\frac{}{\bar{a}\, e.E \xrightarrow{\bar{a}\;\; val(e)} E}$$

Interaction

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

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

Conditional Execution

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

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

$$coin(x).\ if\ x \geq 20\ then\ choc.nil\ else\ nil$$

The user interacts with the machine as follows:

$$coin(x).\ if\ x \geq 20\ then\ \overline{choc}.nil\ else\ nil \ \mid\ \overline{coin}\,40.choc.nil$$
$$\xrightarrow{\tau}$$
$$if\ 40 \geq 20\ then\ \overline{choc}.nil\ else\ nil \ \mid\ choc.nil$$
$$\xrightarrow{\tau}$$
$$nil \ \mid\ nil$$

Abstraction - 1

Restriction

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

Forcing Interaction

$$\begin{array}{c}
(\ (coin.\overline{ok}.nil) \mid ok.(\overline{choc}.nil + \overline{water}.nil)\ ) \setminus ok \quad \mid \quad \overline{coin}.choc.nil \\
\xrightarrow{\tau} \\
(\ (\overline{ok}.nil) \mid ok.(\overline{choc}.nil + \overline{water}.nil)\ ) \setminus ok \quad \mid \quad choc.nil \\
\xrightarrow{\tau} \\
(\ nil \mid (\overline{choc}.nil + \overline{water}.nil)\ ) \setminus ok \quad \mid \quad choc.nil \\
\xrightarrow{\tau} \\
(\ nil \mid nil\ ) \setminus ok \quad \mid \quad nil
\end{array}$$

A malicious user executing $\overline{ok}.choc.nil$ 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

$$\big(\ (coin.ok.nil) \ \ |[ok]| \ \ ok.(choc.nil + water.nil)\ \big)/\ ok$$

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

$$\overline{soldo}.\, acqua.\, nil$$

can interact with the machine with English indication by applying:

$$(\, \overline{soldo}.\, acqua.\, nil\,)\, [coin/soldo,\ water/acqua]$$

Infinite Behaviour - 1

Recursion

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

Long Lasting Vending Machine

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

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

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

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

Infinite Behaviour - 2

Replication

$$\frac{E \xrightarrow{\mu} E'}{!\,E \xrightarrow{\mu} E' |\, !\,E} \qquad \text{or, equivalently} \qquad \frac{E |\, !\,E \xrightarrow{\mu} E'}{!\,E \xrightarrow{\mu} E'}$$

The replication operator can be defined by the following equation $!\,E \triangleq E |!\,E$ that can be expressed in terms of rec as follows: $recX.(E|X)$

Chocolate ad libitum

$$\begin{array}{ll}
!\,coin.\,\overline{choc}.\,nil & \xrightarrow{coin} \\
\overline{choc}.\,nil \;|\; !\,coin.\,\overline{choc}.\,nil & \xrightarrow{coin} \\
\overline{choc}.\,nil \;|\; \overline{choc}.\,nil \;|\; !\,coin.\,\overline{choc}.\,nil & \xrightarrow{\overline{choc}} \\
nil \;|\; \overline{choc}.\,nil \;|\; !\,coin.\,\overline{choc}.\,nil & \xrightarrow{\overline{choc}} \\
nil \;|\; nil \;|\; !\,coin.\,\overline{choc}.\,nil &
\end{array}$$

Infinite Behaviour - 3

Iteration

$$\frac{}{E^* \xrightarrow{\epsilon} \surd} \quad \text{and} \quad \frac{E \xrightarrow{\mu} E'}{E^* \xrightarrow{\mu} E'; E^*}$$

This iteration operator is the classical one of regular expressions.