

# An algebraic process calculus

Emmanuel Beffara



Institut de Mathématiques de Luminy  
CNRS & Université Aix-Marseille II



the **choco** project

LICS 2008, Pittsburgh

# Evaluation vs execution

Two related notions:

**evaluation** rules that compute the *value* of a term

**execution** rules that describe its operational *behaviour*

In the  $\lambda$ -calculus:

**evaluation** is  $\beta(\eta)$ -reduction

**execution** is evaluation strategies (CBN, CBV, lazy, linear)  
or even abstract machines (KAM, PAM, etc.)

In  $\pi$ -like calculi:

**execution** is what we all know (LTS, reaction rules...)

**evaluation** ?

# The problem with concurrency

Execution is non-deterministic.

So how can we define the value of a process?

**choose** a properly *denotational* (i.e. testing) semantics

**derive** an axiomatic system for equivalence

**deduce** reduction rules from the equations

If everything is done right, one should get

**confluence** of reductions

**termination** at least for processes with no infinite behaviour

**normal forms** that give a semantics

# Non-determinism as a formal sum

Take your favorite process calculus:  
(the paper uses the  $\pi$ -calculus)

$p, q := \alpha.p$	action prefix
$p \mid q$	parallel composition
$(\nu x)p$	hiding

In order to represent non-determinism, introduce

- ▶ a formal algebraic sum, written  $\oplus$
- ▶ its neutral element, written  $\emptyset$
- ▶ see what happens

Introduce a constant  $\emptyset$  as a testing token:

**acceptance**  $p$  is accepting if  $\emptyset$  is in active position

**success**  $p$  is successful if every maximal run of  $p$  reaches acceptance

**testing**  $p$  passes the test  $q$  if  $p \mid q$  is successful (write it  $p \perp q$ )

**preorder**  $p \sqsubseteq q$  if  $p \perp r$  implies  $q \perp r$

**equivalence**  $p \simeq q$  if  $p \sqsubseteq q$  and  $q \sqsubseteq p$

This is just standard must testing, it is congruent.

# The easy part

Introduce a sum that represents pure non-determinism, i.e.

$$p \oplus q \rightarrow p \qquad p \oplus q \rightarrow q$$

Compare with the typical situation

$$(\nu a)(a.p \mid a.q \mid \bar{a}) \rightarrow p \mid (\nu a)a.q \simeq p$$

$$(\nu a)(a.p \mid a.q \mid \bar{a}) \rightarrow (\nu a)a.p \mid q \simeq q$$

indeed we have

$$(\nu a)(a.p \mid a.q \mid \bar{a}) \simeq p \oplus q$$

# The meaning of zero, episode I

Remember that  $\emptyset$  is the testing token:

**neutrality**  $p \oplus \emptyset \simeq p$   
*in must testing, we consider the worst case*

**nullity**  $p \mid \emptyset \simeq \emptyset, \quad (\nu x)\emptyset \simeq \emptyset$   
*acceptance is stable*

This yields a nice structure:

**semiring** with product  $(\mid, 1)$  distributing over sum  $(\oplus, \emptyset)$

**linearity** of composition and hiding

**additivity** of actions:  $\alpha.(p \oplus q) \simeq \alpha.p \oplus \alpha.q$   
*you cannot control the choice  $p \oplus q$ , so it does not matter when the choice is made*

# The meaning of zero, episode II

So  $\oplus$  is pure non-deterministic choice and  $\emptyset$  is its neutral element. This suggests a different meaning of  $\emptyset$  in active position:

**inconsistency** *I cannot be in this state / this state does not exist*

**backtracking** *I made an invalid choice in some  $\oplus$ , I must make another choice*

Then we can interpret  $\alpha.\emptyset$ :

**inaction** *I could have done  $\alpha$ , but I know I will not*

This is *not* the same as being inactive.

# Linear actions

The opposite of  $\alpha.\emptyset$  is linear actions:

$\hat{\alpha}.p$  *I can do  $\alpha$  and I know I will*

To give a meaning to this, we need:

**consistency** a run of a process is consistent if all linear actions are eventually triggered

**success** a process is successful if any *consistent* maximal run reaches acceptance

Then these actions are properly linear:

$\hat{\alpha}.(p \oplus q) \simeq \hat{\alpha}.p \oplus \hat{\alpha}.q$

the same as for standard actions

$\hat{\alpha}.\emptyset \simeq \emptyset$  *I know I must do  $\alpha$  and then be successful*

# Decomposing things

Linear actions yield a decomposition of affine actions:

$$\alpha.p \simeq \hat{\alpha}.p \oplus \alpha.\emptyset$$

*either I will do  $\alpha$  at some point, or I will never do it*

$\hat{\alpha}.p$  linear part

$\alpha.\emptyset$  constant part

External choice can be decomposed too:

$$\alpha.p + \beta.q \simeq \hat{\alpha}.p \oplus \hat{\beta}.q \oplus \alpha.\emptyset \mid \beta.\emptyset$$

*read: either I will do  $\alpha$ , or I will do  $\beta$ , or I will never do either of them*

Linear actions and  $\emptyset$  are used to make **assumptions** on the behaviour of the environment.

# Normalising things

These assumptions allows many equivalence rules:

$$\hat{\alpha}.p \mid \hat{\beta}.q \simeq \hat{\alpha}.(p \mid \hat{\beta}.q) \oplus \hat{\beta}.(\hat{\alpha}.p \mid q)$$

*either  $\hat{\alpha}$  or  $\hat{\beta}$  acts first*

$$\hat{\alpha}.p \mid \hat{\hat{\alpha}}.q \simeq \hat{\alpha}.(p \mid \hat{\hat{\alpha}}.q) \oplus \hat{\hat{\alpha}}.(\hat{\alpha}.p \mid q) \oplus (\nu x)(p \mid q)$$

*... or maybe they interact*

$$(\nu u)\hat{u}(x).p \simeq \emptyset$$

*a linear action with no one to talk to is inconsistent*

$$\hat{\alpha}.p \mid \beta.\emptyset \simeq \hat{\alpha}.(p \mid \beta.\emptyset)$$

*actions occur before inactions, of course...*

$$\alpha.\emptyset \mid \bar{\alpha}.\emptyset \simeq \emptyset$$

*because if an action can happen, then it will*

# Normalising things

These assumptions allows many reduction rules:

$$\hat{\alpha}.p \mid \hat{\beta}.q \succ \hat{\alpha}.(p \mid \hat{\beta}.q) \oplus \hat{\beta}.(\hat{\alpha}.p \mid q)$$

*either  $\hat{\alpha}$  or  $\hat{\beta}$  acts first*

$$\hat{\alpha}.p \mid \hat{\hat{\alpha}}.q \succ \hat{\alpha}.(p \mid \hat{\hat{\alpha}}.q) \oplus \hat{\hat{\alpha}}.(\hat{\alpha}.p \mid q) \oplus (\nu x)(p \mid q)$$

*... or maybe they interact*

$$(\nu u)\hat{u}(x).p \succ \emptyset$$

*a linear action with no one to talk to is inconsistent*

$$\hat{\alpha}.p \mid \beta.\emptyset \succ \hat{\alpha}.(p \mid \beta.\emptyset)$$

*actions occur before inactions, of course...*

$$\alpha.\emptyset \mid \bar{\alpha}.\emptyset \succ \emptyset$$

*because if an action can happen, then it will*

# Example

We can detect deadlocks:

$$\begin{aligned} a.\bar{b} \mid b.\bar{a}.p &\simeq a.\emptyset \mid b.\emptyset \\ &\oplus \hat{a}.(\bar{b} \mid b.\bar{a}.p) \\ &\oplus \hat{b}.(a.\bar{b} \mid \bar{a}.p) \end{aligned}$$

# Example

We can detect deadlocks:

$$\begin{aligned}(\nu a)(a.\bar{b} \mid b.\bar{a}.p) &\simeq (\nu a)(a.\emptyset \mid b.\emptyset) \\ &\oplus (\nu a)\hat{a}.\bar{b} \mid b.\bar{a}.p \\ &\oplus (\nu a)\hat{b}.(a.\bar{b} \mid \bar{a}.p)\end{aligned}$$

We can detect deadlocks:

$$\begin{aligned}(\nu a)(a.\bar{b} \mid b.\bar{a}.p) &\simeq (\nu a)(a.\emptyset \mid b.\emptyset) \\ &\oplus (\nu a)\hat{a}.\bar{b} \mid b.\bar{a}.p \\ &\oplus (\nu a)\hat{b}.(a.\bar{b} \mid \bar{a}.p) \\ &\simeq b.\emptyset \oplus \hat{b}.(\nu a)(a.\bar{b} \mid \bar{a}.p) \\ &\simeq b.\emptyset \oplus \hat{b}.\bar{b} \mid p \\ &\simeq b.(\bar{b} \mid p)\end{aligned}$$

# Example

We can detect deadlocks:

$$\begin{aligned}(\nu ab)(a.\bar{b} \mid b.\bar{a}.p) &\simeq (\nu ab)(a.\emptyset \mid b.\emptyset) \\ &\oplus (\nu ab)\hat{a}.\bar{b} \mid b.\bar{a}.p \\ &\oplus (\nu ab)\hat{b}.(a.\bar{b} \mid \bar{a}.p) \\ &\simeq (\nu b)b.\emptyset \oplus (\nu b)\hat{b}.(\nu a)(a.\bar{b} \mid \bar{a}.p) \\ &\simeq (\nu b)b.\emptyset \oplus (\nu b)\hat{b}.(\bar{b} \mid p)\end{aligned}$$

# Example

We can detect deadlocks:

$$\begin{aligned}(\nu ab)(a.\bar{b} \mid b.\bar{a}.p) &\simeq (\nu ab)(a.\emptyset \mid b.\emptyset) \\ &\oplus (\nu ab)\hat{a}.\bar{b} \mid b.\bar{a}.p \\ &\oplus (\nu ab)\hat{b}.\bar{a}.p \\ &\simeq (\nu b)b.\emptyset \oplus (\nu b)\hat{b}.\bar{a}.p \\ &\simeq (\nu b)b.\emptyset \oplus (\nu b)\hat{b}.\bar{b} \mid p \\ &\simeq 1 \oplus \emptyset \\ &\simeq 1\end{aligned}$$

## Theorem

*Relation  $\succ$  preserves testing semantics.*

## Theorem

*Relation  $\succ$  is locally confluent.*

## Theorem

*Relation  $\succ$  is well-founded on terms without replication.*

Normal forms are sums of **traces**:

$$\hat{\alpha}_1.\hat{\alpha}_2\dots\hat{\alpha}_n.(\beta_1.\emptyset \mid \dots \mid \beta_k.\emptyset)$$

$\hat{\alpha}_i$  are linear actions,

$\beta_j.\emptyset$  are inactions, with  $\beta_i \neq \bar{\beta}_j$ .

The testing preorder has a straightforward characterisation:

$$\vec{\alpha}.M \sqsubseteq \vec{\beta}.N \quad \text{iff} \quad \vec{\alpha} = \vec{\beta} \text{ and } M \subseteq N$$

Consider sets of traces and let

$$\llbracket p \rrbracket = \{ t \mid t \text{ is a trace and } p \sqsubseteq t \}$$

## Theorem

$p \sqsubseteq q$  if and only if  $\llbracket p \rrbracket \subseteq \llbracket q \rrbracket$ .

*this coincides with Olderog and Hoare's readiness semantics*

## Proof.

Linear actions do not change the expressiveness of the calculus:

$$(\nu u)(\alpha.u.p \mid u.\emptyset \mid \bar{u}) \simeq \hat{\alpha}.p$$

□

We have

**normalisation** strongly normalising system for processes without losing any concurrent feature

**presentation** a purely axiomatic system for testing semantics

**structure** a monoid structure that is more primitive than non-determinism and choice: expresses both of them with extra niceness properties

... all in the finitary  $\pi$ -calculus.

# What next

**name passing** works if we add *matching* assumptions:

$$\hat{u}(x).p \mid \hat{v}\langle y \rangle.q \\ \simeq \hat{u}(x).(p \mid \hat{v}\langle y \rangle.q) + \hat{v}\langle y \rangle.(u(x).p \mid q) + \{u=v\} \mid p[y/x] \mid q$$

**replication** works if we add infinite traces, but termination of the rules is immediately lost  $\rightarrow$  more cleverness required

**typing** allows further reductions

**coefficients** could be added to represent more subtle features (probabilistic choice? quantum superposition?)