

(Un)decidable Testing Relations for Infinitary Asynchronous **ccs**

Giuseppe Castagna Luca Padovani

Laboratoire PPS, CNRS, Université Paris Diderot

Istituto di Scienze e Tecnologie dell'Informazione
Università di Urbino "Carlo Bo"

INFINITY 2009

Outline

- ① Motivation
 - Session types
 - Types as processes
- ② Buffered ccs
 - Syntax and semantics
 - Undecidability results
- ③ Approximated testing relations
- ④ Concluding remarks

Outline

- 1 Motivation
 - Session types
 - Types as processes
- 2 Buffered ccs
 - Syntax and semantics
 - Undecidability results
- 3 Approximated testing relations
- 4 Concluding remarks

Session types in a nutshell

$c : ?\text{Int}.\text{?Int}.(!\text{Real} \oplus !\text{DivideByZero})$

$d : !\text{Int}.\text{!Int}.(?\text{Real} + ?\text{DivideByZero})$

- input actions
- output actions
- sequence
- internal choice
- external choice

Session types in a nutshell

$c : ?\text{Int}.\text{?Int}.\text{(!Real} \oplus \text{!DivideByZero)}$

$d : \text{!Int}.\text{!Int}.\text{(?Real} + \text{?DivideByZero)}$

- input actions
- **output actions**
- sequence
- internal choice
- external choice

Session types in a nutshell

$c : ?\text{Int}.?\text{Int}.(!\text{Real} \oplus !\text{DivideByZero})$



$d : !\text{Int}.!\text{Int}.(? \text{Real} + ? \text{DivideByZero})$

- input actions
- output actions
- **sequence**
- internal choice
- external choice

Session types in a nutshell

c : ?Int.?Int.(!Real \oplus !DivideByZero)


d : !Int.!Int.(?Real + ?DivideByZero)

- input actions
- output actions
- sequence
- **internal choice**
- external choice

Session types in a nutshell

$c : ?\text{Int}.\text{?Int}.(!\text{Real} \oplus !\text{DivideByZero})$

$d : !\text{Int}.\text{!Int}.(?\text{Real} + \text{?DivideByZero})$



- input actions
- output actions
- sequence
- internal choice
- **external choice**

Duality and subtyping

Duality = correct composition

$?Int. ?Int. (!Real \oplus !DivideByZero)$



$!Int. !Int. (?Real + ?DivideByZero)$

Subtyping = safe substitution

$?Int <: ?Int + ?Bool$
 $!Int \oplus !Bool <: !Int$

Duality and subtyping

Duality = correct composition

$$\begin{array}{ccc} ?\text{Int}.\text{?Int}.(!\text{Real} \oplus !\text{DivideByZero}) & & \\ \vdots & \times & \vdots \\ !\text{Int}.\text{!Int}.(?\text{Real} + ?\text{DivideByZero}) & & \end{array}$$

Subtyping = safe substitution

$$\begin{array}{l} ?\text{Int} <: ?\text{Int} + ?\text{Bool} \\ !\text{Int} \oplus !\text{Bool} <: !\text{Int} \end{array}$$

Duality and subtyping

Duality = correct composition

$$\begin{array}{ccc} ?\text{Int}.\text{?Int}.(!\text{Real} \oplus !\text{DivideByZero}) & & \\ \vdots & \times & \vdots \\ !\text{Int}.\text{!Int}.(?\text{Real} + ?\text{DivideByZero}) & & \end{array}$$

Subtyping = safe substitution

$$\begin{array}{l} ?\text{Int} <: ?\text{Int} + ?\text{Bool} \\ !\text{Int} \oplus !\text{Bool} <: !\text{Int} \end{array}$$

What are session types anyway?

- Laneve, Padovani, “**The Pairing of Contracts and Session Types**”, 2008
- Padovani, “**Session Types at the Mirror**”, ICE 2009
- Castagna, Dezani-Ciancaglini, Giachino, Padovani, “**Foundations of Session Types**”, PPDP 2009

$?Int.?Int.(!Real \oplus !DivideByZero)$

$\stackrel{?}{=}$

$a.a.(\bar{b} \oplus \bar{c})$

The problem

Session types scale to asynchrony

- + reflects practice of distributed communication
- mismatches with **synchronous ccs**

Q: Why not asynchronous ccs? A: It's "too" asynchronous

- message order is not preserved

$$\bar{a} | \bar{b} \simeq \bar{b} | \bar{a}$$

- self synchronizations

$$\bar{a} | a \longrightarrow \mathbf{0} | \mathbf{0}$$

- no algorithms

The problem

Session types scale to asynchrony

- + reflects practice of distributed communication
- mismatches with **synchronous ccs**

Q: Why not asynchronous **ccs**? A: It's "too" asynchronous

- message order is not preserved

$$\bar{a} | \bar{b} \simeq \bar{b} | \bar{a}$$

- self synchronizations

$$\bar{a} | a \rightarrow \mathbf{0} | \mathbf{0}$$

- no algorithms

Outline

- ① Motivation
 - Session types
 - Types as processes
- ② Buffered ccs
 - Syntax and semantics
 - Undecidability results
- ③ Approximated testing relations
- ④ Concluding remarks

Proposal: asynchronous buffered ccs

$P ::=$	0	process
	1	(failure)
	$a.P$	(success)
	$\bar{a}.P$	(input)
	$P + P$	(buffer)
	$P \oplus P$	(external choice)
		(internal choice)

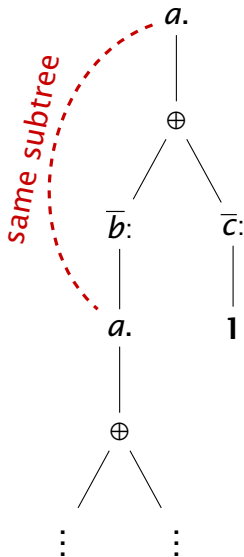
Streamlined design

- builtin buffers
- two choices

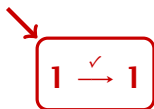
Infinite processes as infinite terms

$$P = a.(\bar{b}:P \oplus \bar{c}:1)$$

- regular trees
- contractivity



Labeled operational semantics



$$a.P \xrightarrow{a} P \quad \bar{a}.P \xrightarrow{\bar{a}} P$$

$$P \oplus Q \longrightarrow P \quad \frac{P \longrightarrow P'}{\bar{a}.P \longrightarrow \bar{a}.P'} \quad \frac{P \xrightarrow{b} P'}{\bar{a}.P \xrightarrow{b} \bar{a}.P'}$$

$$\frac{P \xrightarrow{a} P'}{P + Q \xrightarrow{a} P'} \quad \frac{P \xrightarrow{\bar{a}} P'}{P + Q \longrightarrow \bar{a}.P'}$$

$P = a.\bar{b}.P$ is regular but infinite state

$$P \xrightarrow{a} \bar{b}.P \xrightarrow{a} \bar{b}.\bar{b}.P \xrightarrow{a} \bar{b}.\bar{b}.\bar{b}.P \xrightarrow{a} \dots$$

Labeled operational semantics

$$1 \xrightarrow{\checkmark} 1 \quad a.P \xrightarrow{a} P \quad \bar{a}.P \xrightarrow{\bar{a}} P$$

$$P \oplus Q \longrightarrow P$$

$$\frac{P \longrightarrow P'}{\bar{a}.P \longrightarrow \bar{a}.P'}$$

$$\frac{P \xrightarrow{b} P'}{\bar{a}.P \xrightarrow{b} \bar{a}.P'}$$

$$\frac{P \xrightarrow{a} P'}{P + Q \xrightarrow{a} P'} \quad \frac{P \xrightarrow{\bar{a}} P'}{P + Q \longrightarrow \bar{a}.P'}$$

$P = a.\bar{b}.P$ is regular but infinite state

$$P \xrightarrow{a} \bar{b}.P \xrightarrow{a} \bar{b}.\bar{b}.P \xrightarrow{a} \bar{b}.\bar{b}.\bar{b}.P \xrightarrow{a} \dots$$

Labeled operational semantics

$$\begin{array}{c}
 1 \xrightarrow{\checkmark} 1 \quad a.P \xrightarrow{a} P \quad \bar{a}.P \xrightarrow{\bar{a}} P \\
 \\
 P \oplus Q \longrightarrow P \quad \frac{P \longrightarrow P'}{\bar{a}.P \longrightarrow \bar{a}.P'} \\
 \\
 \boxed{\frac{P \xrightarrow{b} P'}{\bar{a}.P \xrightarrow{b} \bar{a}.P'}}
 \end{array}$$

$$\frac{P \xrightarrow{a} P'}{P + Q \xrightarrow{a} P'} \quad \frac{P \xrightarrow{\bar{a}} P'}{P + Q \longrightarrow \bar{a}.P'}$$

$P = a.\bar{b}.P$ is regular but infinite state

$$P \xrightarrow{a} \bar{b}.P \xrightarrow{a} \bar{b}.\bar{b}.P \xrightarrow{a} \bar{b}.\bar{b}.\bar{b}.P \xrightarrow{a} \dots$$

Testing relations

Systems

$$P \parallel Q$$

System reduction

$$\frac{P \rightarrow P'}{P \parallel Q \rightarrow P' \parallel Q}$$

$$\frac{Q \rightarrow Q'}{P \parallel Q \rightarrow P \parallel Q'}$$

$$\frac{P \xrightarrow{\bar{\alpha}} P' \quad Q \xrightarrow{\alpha} Q'}{P \parallel Q \rightarrow P' \parallel Q'}$$

Definition

- $P \bowtie Q$ if $P \parallel Q \Rightarrow P' \parallel Q'$ implies

$$P' \xRightarrow{\bar{\mu}} \text{ and } Q' \xRightarrow{\mu} \text{ for some } \mu$$

$$\overline{\checkmark} = \checkmark$$

- $P \preceq Q$ if $R \bowtie P$ implies $R \bowtie Q$ for every R

Testing relations

Systems

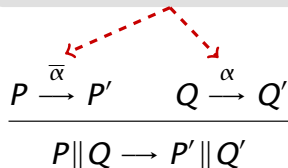
$P \parallel Q$

System reduction

$$\frac{P \rightarrow P'}{P \parallel Q \rightarrow P' \parallel Q}$$

$$\frac{Q \rightarrow Q'}{P \parallel Q \rightarrow P \parallel Q'}$$

complementary actions



The diagram shows two transitions: $P \xrightarrow{\bar{\alpha}} P'$ and $Q \xrightarrow{\alpha} Q'$. A red dashed double-headed arrow connects the two transitions, indicating they are complementary actions. Below these transitions is a horizontal line, and under the line is the equation $P \parallel Q \rightarrow P' \parallel Q'$.

$$\frac{P \xrightarrow{\bar{\alpha}} P' \quad Q \xrightarrow{\alpha} Q'}{P \parallel Q \rightarrow P' \parallel Q'}$$

Definition

- $P \bowtie Q$ if $P \parallel Q \Rightarrow P' \parallel Q'$ implies
 $P' \xRightarrow{\bar{\mu}}$ and $Q' \xRightarrow{\mu}$ for some μ

$\bar{\checkmark} = \checkmark$

- $P \preceq Q$ if $R \bowtie P$ implies $R \bowtie Q$ for every R

Testing relations

Systems

$$P \parallel Q$$

System reduction

$$\frac{P \rightarrow P'}{P \parallel Q \rightarrow P' \parallel Q} \quad \frac{Q \rightarrow Q'}{P \parallel Q \rightarrow P \parallel Q'} \quad \frac{P \xrightarrow{\bar{\alpha}} P' \quad Q \xrightarrow{\alpha} Q'}{P \parallel Q \rightarrow P' \parallel Q'}$$

Definition

- $P \bowtie Q$ if $P \parallel Q \Rightarrow P' \parallel Q'$ implies $P' \xRightarrow{\bar{\mu}}$ and $Q' \xRightarrow{\mu}$ for some μ
- $P \preceq Q$ if $R \bowtie P$ implies $R \bowtie Q$ for every R

$$\overline{\checkmark} = \checkmark$$

Examples

Duality

$$\begin{array}{l} a.1 \not\bowtie \bar{a}:1 \oplus \bar{b}:1 \\ a.1 + b.1 \bowtie \bar{a}:1 \oplus \bar{b}:1 \end{array}$$

$$\begin{array}{l} a.1 \bowtie \bar{a}:1 \\ a.1 + b.1 \bowtie \bar{a}:1 \end{array}$$

Refinement

$$\begin{array}{l} \bar{a}:P \oplus \bar{b}:Q \preceq \bar{a}:P \\ a.P \preceq a.P + b.Q \end{array}$$

About undecidability of duality

- duality \sim DDP for CFSMs
- DDP for CSFMs is undecidable

Outline

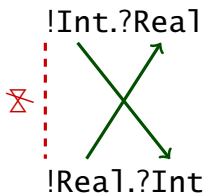
- ① Motivation
 - Session types
 - Types as processes
- ② Buffered ccs
 - Syntax and semantics
 - Undecidability results
- ③ **Approximated testing relations**
- ④ Concluding remarks

Syntactic vs semantic duality

Semantic duality: output messages may accumulate

$$\begin{array}{ccccccc} P \stackrel{\text{def}}{=} \bar{a}:P & \xRightarrow{\bar{a}} P & \xRightarrow{\bar{a}} P & \xRightarrow{\bar{a}} P & \dots \\ Q \stackrel{\text{def}}{=} a.\bar{b}:Q & \xRightarrow{a} \bar{b}:Q & \xRightarrow{a} \bar{b}:\bar{b}:Q & \xRightarrow{a} \bar{b}:\bar{b}:\bar{b}:Q & \dots \end{array}$$

Syntactic duality: simultaneous outputs are disallowed



Approximated duality

Definition

- $P \bowtie_a Q$ if $P \parallel Q \Rightarrow P' \parallel Q'$ implies

① $P' \xRightarrow{\bar{\mu}}$ and $Q' \xRightarrow{\mu}$ for some μ

$$\bar{\checkmark} = \checkmark$$

② $P' \parallel Q' \equiv \bar{s}:P'' \parallel \bar{t}:Q''$ implies either $s = \varepsilon$ or $t = \varepsilon$

$$P \stackrel{\text{def}}{=} \bar{a}:P$$

$$\xrightarrow{\bar{a}}$$

P

~~\bowtie_a~~

$$Q \stackrel{\text{def}}{=} a.\bar{b}:Q$$

$$\xrightarrow{a}$$

$\bar{b}:Q$

$$P \stackrel{\text{def}}{=} \bar{a}:P \oplus \bar{c}:1$$

$$\Rightarrow \bar{a}^n:P$$

$$\Rightarrow \bar{c}:1$$

$$\xrightarrow{\bar{c}} 1$$

$$Q \stackrel{\text{def}}{=} a.Q + c.1$$

$$\Rightarrow a^n:Q$$

$$\xrightarrow{c} 1$$

Deciding approximated duality

Definition

- $P \bowtie_a Q$ if $P \parallel Q \Rightarrow P' \parallel Q'$ implies

① $P' \xRightarrow{\bar{\mu}}$ and $Q' \xRightarrow{\mu}$ for some μ

$\bar{\nu} = \check{\nu}$

② $P' \parallel Q' \equiv \bar{s}:P'' \parallel \bar{t}:Q''$ implies either $s = \varepsilon$ or $t = \varepsilon$

- input actions do not depend on buffers
- output actions may accumulate in buffers, but only the first one matters

Residuals

subtrees of P

$$\mathcal{R}(P) \stackrel{\text{def}}{=} \{\bar{a}_1:P' \mid \exists \varphi : P \xRightarrow{\varphi} \bar{a}_1:\dots:\bar{a}_n:P', P' \in \mathcal{T}(P)\}$$

Example

forget content of buffer except possibly for a_1

$$P \stackrel{\text{def}}{=} \bar{a}:P$$

$$\mathcal{R}(P) = \{P\}$$

$$Q \stackrel{\text{def}}{=} a.\bar{b}:Q$$

$$\mathcal{R}(Q) = \{Q, \bar{b}:Q\}$$

Proposition

$\mathcal{R}(P)$ is finite for every P

Residuals

$$\mathcal{R}(P) \stackrel{\text{def}}{=} \{\bar{a}_1:P' \mid \exists \varphi : P \xRightarrow{\varphi} \bar{a}_1:\dots:\bar{a}_n:P', P' \in \mathcal{T}(P)\}$$

Example

$$P \stackrel{\text{def}}{=} \bar{a}:P \qquad \mathcal{R}(P) = \{P\}$$

$$Q \stackrel{\text{def}}{=} a.\bar{b}:Q \qquad \mathcal{R}(Q) = \{Q, \bar{b}:Q\}$$

Proposition

$\mathcal{R}(P)$ is finite for every P

Bad pairs of residuals

$(\bar{s}:P', \bar{t}:Q') \in \mathcal{R}(P) \times \mathcal{R}(Q)$ is bad if:

- $s = a$ and $Q' \Rightarrow Q'' \stackrel{a}{\not\Rightarrow}$
 - $t = a$ and $P' \Rightarrow P'' \stackrel{a}{\not\Rightarrow}$
 - $s = t = \varepsilon$, $P, Q \notin \text{Out}$, $P \stackrel{\check{}}{\not\Rightarrow}$ or $Q \stackrel{\check{}}{\not\Rightarrow}$
 - $s \neq \varepsilon$ and $t \neq \varepsilon$
-
- Duality reduces to checking decidable properties of bad pairs
 - The set of bad pairs is finite
 - Duality can be decided with a finite number of tests

Which bad pairs are reachable?

Suppose

$$(P, Q) \Rightarrow (p, q) \text{ bad!}$$

Either...

- there exists

$$(P, Q) \Rightarrow (\bar{s}:P', \bar{t}:Q') \text{ bad!} \Rightarrow (p, q)$$

...or

- there exists

$$(P, Q) \Rightarrow \begin{array}{c} (P', Q') \\ \in \\ \mathcal{T}(P) \times \mathcal{T}(Q) \end{array} \rightarrow (p, q)$$

Which bad pairs are reachable?

Suppose

$$(P, Q) \Rightarrow (p, q) \text{ bad!}$$

Either...

- there exists

$$(P, Q) \Rightarrow (\bar{s}:P', \bar{t}:Q') \text{ bad!} \Rightarrow (p, q)$$

...or

- there exists

$$(P, Q) \Rightarrow \begin{array}{c} (P', Q') \\ \in \\ \mathcal{T}(P) \times \mathcal{T}(Q) \end{array} \rightarrow (p, q)$$

Approximated refinement

Definition

- $P \preceq_a Q$ if $R \bowtie_a P$ implies $R \bowtie_a Q$ for every R

Properties

$$P \oplus Q \preceq_a P \quad (1)$$

$$a.P \preceq_a a.P + b.Q \quad (2)$$

Approximated refinement is decidable

- it reduces to reachability of residuals

Outline

- 1 Motivation
 - Session types
 - Types as processes
- 2 Buffered ccs
 - Syntax and semantics
 - Undecidability results
- 3 Approximated testing relations
- 4 Concluding remarks

Concluding remarks

Contributions

- ✓ natural restriction of asynchronous **ccs**
- ✓ semantic foundations to theories of session types

Open issues

- algorithms
- can \bowtie_a be relaxed?

Concluding remarks

Contributions

- ✓ natural restriction of asynchronous **ccs**
- ✓ semantic foundations to theories of session types

Open issues

- algorithms
- can \bowtie_a be relaxed?

Thank you.