

# probabilistic analysis of binary sessions

Omar Inverso, Gran Sasso Science Institute

Hernán Melgratti, Universidad de Buenos Aires

**Luca Padovani**, Università di Torino

Catia Trubiani, Gran Sasso Science Institute

Emilio Tuosto, Gran Sasso Science Institute

# context

- session type = protocol with branching points

*Accept & Reject*

- well-typed process = protocol fidelity **along all paths**

## This work

- different paths = different degrees of “success”

😊 *Accept & Reject* ☹️

- probabilistic analysis of the session success

# problem and contribution

Success probability of a session type: **easy**

*Accept <sub>$\rho$</sub>  & Reject*

Success probability of a process: **not so easy**

- arbitrary composition of parallel, interacting processes
- dynamic network topology
- unbounded number of states
- **local choices can propagate globally** through sessions

Our contribution: bridging the gap between types and processes

$x : \text{Accept}_{\rho} \& \text{Reject} \vdash P$

# processes

$P, Q ::=$	<b>idle</b>	inaction
	<b>done</b> $x$	success
	$x?(y).P$	message input
	$x!y.P$	message output
	<b>case</b> $x[P, Q]$	branch
	<b>inl</b> $x.P$	left selection
	<b>inr</b> $x.P$	right selection
	$P \mid Q$	parallel composition
	$(x)P$	session restriction
	$P \text{ }_p \boxplus Q$	probabilistic choice
	$A\langle\bar{x}\rangle$	process invocation

# processes

$P, Q ::=$	<code>idle</code>	inaction
	<code>done</code> $x$	success
	$x?(y).P$	message input
	$x!y.P$	message output
	<code>case</code> $x [P, Q]$	branch
	<code>inl</code> $x.P$	left selection
	<code>inr</code> $x.P$	right selection
	$P \mid Q$	parallel composition
	$(x)P$	session restriction
	$P \boxplus_p Q$	probabilistic choice
	$A\langle\bar{x}\rangle$	process invocation

# processes

$P, Q ::=$	<b>idle</b>	inaction
	<b>done</b> $x$	success
	$x?(y).P$	message input
	$x!y.P$	message output
	<b>case</b> $x[P, Q]$	branch
	<b>inl</b> $x.P$	left selection
	<b>inr</b> $x.P$	right selection
	$P \mid Q$	parallel composition
	$(x)P$	session restriction
	$P \text{ }_p \boxplus Q$	probabilistic choice
	$A\langle\bar{x}\rangle$	process invocation

# processes

$P, Q ::=$	<code>idle</code>	inaction
	<code>done x</code>	success
	<code>x?(y).P</code>	message input
	<code>x!y.P</code>	message output
	<code>case x [P, Q]</code>	branch
	<code>inl x.P</code>	left selection
	<code>inr x.P</code>	right selection
	<code>P   Q</code>	parallel composition
	<code>(x)P</code>	session restriction
	<code>P<sub>p</sub> ⊞ Q</code>	probabilistic choice
	<code>A⟨x̄⟩</code>	process invocation

# example of probabilistic choice propagation

$$(\text{inl } x \oplus \text{inr } x) \mid \text{case } x [\text{inr } y.\text{done } y, \text{inl } y]$$



# example of probabilistic choice propagation

$(\text{inl } x \text{ } _p \boxplus \text{inr } x) \mid \text{case } x [\text{inr } y.\text{done } y, \text{inl } y]$

$\rightsquigarrow$

$(\text{inl } x \mid \text{case } x [\text{inr } y.\text{done } y, \text{inl } y]) \text{ } _p \boxplus (\text{inr } x \mid \text{case } x [\dots, \dots])$

# example of probabilistic choice propagation

$$(\text{inl } x \text{ }_{\rho} \boxplus \text{inr } x) \mid \text{case } x [\text{inr } y.\text{done } y, \text{inl } y]$$
$$\rightsquigarrow$$
$$(\text{inl } x \mid \text{case } x [\text{inr } y.\text{done } y, \text{inl } y]) \text{ }_{\rho} \boxplus (\text{inr } x \mid \text{case } x [\dots, \dots])$$

$$\rightarrow$$
$$\text{inr } y.\text{done } y \text{ }_{\rho} \boxplus (\text{inr } x \mid \text{case } x [\text{inr } y.\text{done } y, \text{inl } y])$$

# example of probabilistic choice propagation

$$(\text{inl } x \text{ } _p \boxplus \text{inr } x) \mid \text{case } x [\text{inr } y.\text{done } y, \text{inl } y]$$
$$\rightsquigarrow$$
$$(\text{inl } x \mid \text{case } x [\text{inr } y.\text{done } y, \text{inl } y]) \text{ } _p \boxplus (\text{inr } x \mid \text{case } x [\dots, \dots])$$
$$\text{inr } y.\text{done } y \text{ } _p \boxplus (\text{inr } x \mid \text{case } x [\text{inr } y.\text{done } y, \text{inl } y])$$
$$(\text{inr } x \mid \text{case } x [\text{inr } y.\text{done } y, \text{inl } y]) \text{ } _{1-p} \boxplus \text{inr } y.\text{done } y$$

# example of probabilistic choice propagation

$$(\text{inl } x \oplus_p \text{inr } x) \mid \text{case } x [\text{inr } y.\text{done } y, \text{inl } y]$$

$\rightsquigarrow$

$$(\text{inl } x \mid \text{case } x [\text{inr } y.\text{done } y, \text{inl } y]) \oplus_p (\text{inr } x \mid \text{case } x [\dots, \dots])$$

$\rightarrow$

$$\text{inr } y.\text{done } y \oplus_p (\text{inr } x \mid \text{case } x [\text{inr } y.\text{done } y, \text{inl } y])$$

$\rightsquigarrow$

$$(\text{inr } x \mid \text{case } x [\text{inr } y.\text{done } y, \text{inl } y])_{1-p} \oplus \text{inr } y.\text{done } y$$


$\rightarrow$

$$\text{inl } y \oplus_{1-p} \text{inr } y.\text{done } y$$

# probabilistic session types

$T, S ::=$	$\circ$	termination
	$\bullet$	success
	$?t.T$	input
	$!t.T$	output
	$T_p \& S$	branch
	$T_p \oplus S$	choice

- **plain** termination vs **successful** termination
- **probability annotations** in branches and choices
- infinite trees with finitely many distinct sub-trees (**regularity**)
- each sub-tree contains a reachable leaf  $\circ$  or  $\bullet$  (**reachability**)

# success probability of a session type

## Definition (success probability – informal)

$\llbracket T \rrbracket$  = cumulative probability of paths from  $T$  to  $\bullet$

Formally, solve this finite system of equations:

$$\begin{aligned} \llbracket \circ \rrbracket &= 0 \\ \llbracket \bullet \rrbracket &= 1 \\ \llbracket T \text{ }_p \& S \rrbracket = \llbracket T \text{ }_p \oplus S \rrbracket &= p \llbracket T \rrbracket + (1 - p) \llbracket S \rrbracket \end{aligned}$$

## Reasoning

- consider the Discrete-Time Markov Chain corresponding to  $T$
- regularity implies that the DTMC is **finite**
- reachability implies that the DTMC is **absorbing**
- the system of equations has **exactly one** solution

# success probability of a session type

## Definition (success probability – informal)

$\llbracket T \rrbracket$  = cumulative probability of paths from  $T$  to  $\bullet$

Formally, solve this finite system of equations:

$$\begin{aligned} \llbracket \circ \rrbracket &= 0 \\ \llbracket \bullet \rrbracket &= 1 \\ \llbracket T \text{ }_p \& S \rrbracket = \llbracket T \text{ }_p \oplus S \rrbracket &= p \llbracket T \rrbracket + (1 - p) \llbracket S \rrbracket \end{aligned}$$

## Reasoning

- consider the Discrete-Time Markov Chain corresponding to  $T$
- regularity implies that the DTMC is **finite**
- reachability implies that the DTMC is **absorbing**
- the system of equations has **exactly one** solution

# type system

$$\Gamma \vdash P$$

- $\Gamma$  is a behavioural abstraction of  $P$  (including probabilities)
- $x : T \in \Gamma \Rightarrow P$  successfully terminates  $x$  with probability  $\llbracket T \rrbracket$



# successful termination

$$\frac{}{x : \bullet \vdash \text{done } x}$$

# deterministic choices

$$\frac{x : T \vdash P}{x : T_1 \oplus S \vdash \text{inl } x.P} \qquad \frac{x : S \vdash P}{x : T_0 \oplus S \vdash \text{inr } x.P}$$

- deterministic process  $\Rightarrow$  trivial probability

# probabilistic choices

$$\frac{x : T_1 \vdash P \quad x : T_2 \vdash Q}{x : T_1 \oplus_p T_2 \vdash P \oplus_p Q}$$

probabilistic combination of  $T_1$  and  $T_2$

$$\begin{aligned} T \oplus_p T &= T \\ (T_q \oplus S) \oplus_p (T_r \oplus S) &= T_{pq+(1-p)r} \oplus S \end{aligned}$$

■ combination is **undefined** otherwise

# probabilistic choices

$$\frac{x : T_1 \vdash P \quad x : T_2 \vdash Q}{x : T_1 \oplus_p T_2 \vdash P \oplus_p Q}$$

probabilistic combination of  $T_1$  and  $T_2$

$$\begin{aligned} T \oplus_p T &= T \\ (T \oplus_q S) \oplus_p (T \oplus_r S) &= T \oplus_{pq+(1-p)r} S \end{aligned}$$

- combination is **undefined** otherwise

# branches and choice propagation

$$\frac{\Gamma, x : T \vdash P \quad \Delta, x : S \vdash Q}{\Gamma \boxplus_p \Delta, x : T \& S \vdash \text{case } x [P, Q]}$$

- $\Gamma$  and  $\Delta$  are nearly the same
- choices in  $\Gamma$  and  $\Delta$  affected by information received from  $x$
- choices in  $\Gamma$  and  $\Delta$  **weighed** by  $p$

# parallel composition

endpoint

endpoint

$$\frac{\Gamma, x : T \vdash P \quad \Delta, x : \bar{T} \vdash Q}{\Gamma, \Delta, x : \langle \llbracket T \rrbracket \rangle \vdash P \mid Q}$$

whole session

- $\langle p \rangle$  = type of a session with success probability  $p$

# example

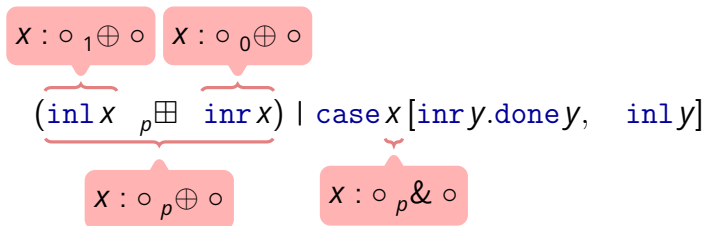
$$\begin{array}{c} \underbrace{X : \circ_1 \oplus \circ} \quad \underbrace{X : \circ_0 \oplus \circ} \\ (\text{inl } x \quad \text{inr } x) \mid \text{case } x [\text{inr } y.\text{done } y, \quad \text{inl } y] \end{array}$$

# example

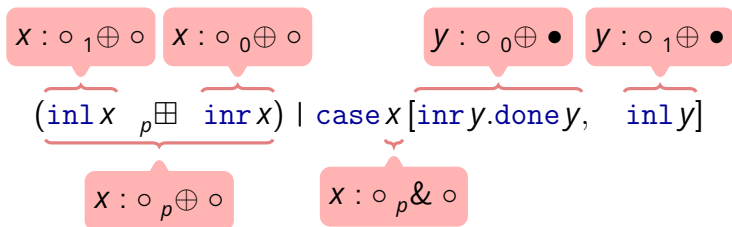
$$\begin{array}{c} X : \circ_1 \oplus \circ \quad X : \circ_0 \oplus \circ \\ \underbrace{\quad \quad \quad} \quad \underbrace{\quad \quad \quad} \\ (\text{inl } x \text{ } \boxplus \text{ } \text{inr } x) \mid \text{case } x [\text{inr } y.\text{done } y, \text{ inl } y] \\ \underbrace{\quad \quad \quad} \\ X : \circ_p \oplus \circ \end{array}$$



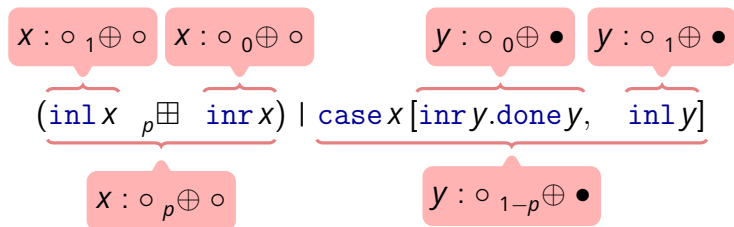
# example



# example



# example



# subject reduction

## Informally

Types – not just typing – are preserved by reductions.

## Theorem

*If  $\Gamma \vdash P$  and  $P \rightarrow Q$ , then  $\Gamma \vdash Q$ .*

## Particular instance

If  $x : \langle p \rangle \vdash P$  and  $P \rightarrow Q$ , then  $x : \langle p \rangle \vdash Q$ .

- **unresolved** prob. choices  $\Rightarrow$  **steady** success probabilities
- suitable design choice for specifying **invariants**

# soundness

## Definition

We write  $P \uparrow_p^x$  iff  $P$  has a top-level **done**  $x$  with probability  $p$ , that is

$$P \uparrow_p^x \iff P \preceq \text{done } x \text{ }_p \boxplus Q$$

## Theorem

*If  $x : \langle p \rangle \vdash P$  and  $P \dashv\rightarrow$ , then  $P \uparrow_p^x$ .*

## Remarks

- deadlock freedom is a **necessary condition**
  - type system enforces an acyclic (tree-like) network topology
- useless when  $P$  reduces forever

# soundness

## Definition

We write  $P \uparrow_p^x$  iff  $P$  has a top-level `done`  $x$  with probability  $p$ , that is

$$P \uparrow_p^x \iff P \preceq \text{done } x \text{ }_p \boxplus Q$$

## Theorem

*If  $x : \langle p \rangle \vdash P$  and  $P \dashv\rightarrow$ , then  $P \uparrow_p^x$ .*

## Remarks

- deadlock freedom is a **necessary condition**
  - type system enforces an acyclic (tree-like) network topology
- useless when  $P$  reduces forever

# probabilistic termination

$$A(x) := \text{done } x \text{ } \boxplus \text{ } A(x)$$

Fact:  $A(x)$  reduces forever

$$A(x) \rightarrow \text{done } x \text{ } \boxplus (\text{done } x \text{ } \boxplus A(x)) \rightarrow \dots$$

Fact: if  $p > 0$ , the probability of reaching an irreducible state is 1

$$p + (1 - p)p + (1 - p)^2 p + \dots = \frac{p}{1 - (1 - p)} = 1$$

# limit soundness

## Definition (eventual success of a session)

Let  $P \uparrow_p^x$  if  $P = P_0 \rightarrow P_1 \rightarrow P_2 \rightarrow \dots$  and

- 1  $P_n \uparrow_{p_n}^x$  for all  $n \in \mathbb{N}$
- 2  $\lim_{n \rightarrow \infty} p_n = p$

## Theorem

*If  $x : \langle p \rangle \vdash P$  and  $P$  terminates with probability 1, then  $P \uparrow_p^x$ .*

- Conclusion holds also if the termination probability is  $p < 1$  and the successful completion probability is 1 (see paper).



# Wrap up

With which probability  $P$  terminates session  $x$  successfully?

- easy to do from session types, less so for processes
- type system fills the gap

Future work

- type inference
- subtyping

# Wrap up

With which probability  $P$  terminates session  $x$  successfully?

- easy to do from session types, less so for processes
- type system fills the gap

Future work

- type inference
- subtyping

**thank you**