

# Switched Probabilistic I/O Automata

Ling Cheung<sup>1</sup>   Nancy Lynch<sup>2</sup>   Roberto Segala<sup>3</sup>  
Frits Vaandrager<sup>1</sup>

<sup>1</sup>Nijmegen Institute for Computing and Information Sciences  
University of Nijmegen, the Netherlands

<sup>2</sup>MIT Computer Science and Artificial Intelligence Laboratory, U.S.A.

<sup>3</sup>Dipartimento di Informatica, Università di Verona, Italy

ICTAC 2004, Guiyang, China

# Outline

- 1 Introduction
  - Basics
  - Randomization

# Outline

- 1 Introduction
  - Basics
  - Randomization
- 2 The trouble with composition
  - What is parallel composition?
  - How much does the daemon know?
  - Global choice vs local choice

# Outline

- 1 Introduction
  - Basics
  - Randomization
- 2 The trouble with composition
  - What is parallel composition?
  - How much does the daemon know?
  - Global choice vs local choice
- 3 Switched PIOA
  - The Switched PIOA model
  - Implementing parallel compositions

# Outline

- 1 Introduction
  - Basics
  - Randomization
- 2 The trouble with composition
  - What is parallel composition?
  - How much does the daemon know?
  - Global choice vs local choice
- 3 Switched PIOA
  - The Switched PIOA model
  - Implementing parallel compositions
- 4 Summary and future work
  - Summary
  - Future work

## For this talk ...

- We need very little probability theory: *discrete distributions*.  
Examples:
  - fair coin:  $\{\langle \text{Head}, \frac{1}{2} \rangle, \langle \text{Tail}, \frac{1}{2} \rangle\}$ ;
  - fair dice:  $\{\langle i, \frac{1}{6} \rangle \mid 1 \leq i \leq 6\}$ .

## For this talk ...

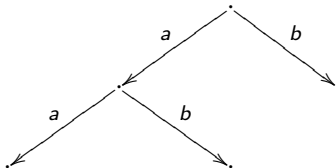
- We need very little probability theory: *discrete distributions*.  
Examples:
  - fair coin:  $\{\langle \text{Head}, \frac{1}{2} \rangle, \langle \text{Tail}, \frac{1}{2} \rangle\}$ ;
  - fair dice:  $\{\langle i, \frac{1}{6} \rangle \mid 1 \leq i \leq 6\}$ .
- Underlying model: nondeterministic automata with asynchronous composition.  
(In our paper: input/output distinction, combination of synchronous and asynchronous compositions, etc.)

## For this talk ...

- We need very little probability theory: *discrete distributions*.  
Examples:
  - fair coin:  $\{\langle \text{Head}, \frac{1}{2} \rangle, \langle \text{Tail}, \frac{1}{2} \rangle\}$ ;
  - fair dice:  $\{\langle i, \frac{1}{6} \rangle \mid 1 \leq i \leq 6\}$ .
- Underlying model: nondeterministic automata with asynchronous composition.  
(In our paper: input/output distinction, combination of synchronous and asynchronous compositions, etc.)
- Total order semantics: if both actions  $a$  and  $b$  occur, one must precede the other.

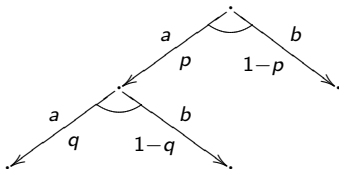
## Schedulers and trace distributions

- *History-dependent, randomized* schedulers transform nondeterministic choices into probabilistic choices.



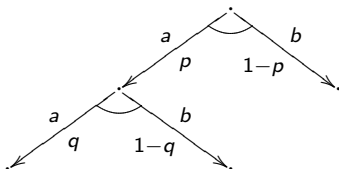
## Schedulers and trace distributions

- *History-dependent, randomized* schedulers transform nondeterministic choices into probabilistic choices.



## Schedulers and trace distributions

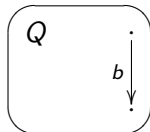
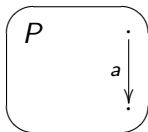
- *History-dependent, randomized* schedulers transform nondeterministic choices into probabilistic choices.



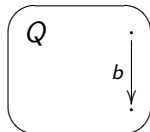
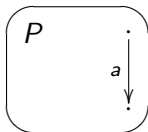
- Each scheduler induces a *trace distribution*: a discrete distributions on finite traces.

$$\{\langle aa, pq \rangle, \langle ab, p(1 - q) \rangle, \langle b, 1 - p \rangle\}$$

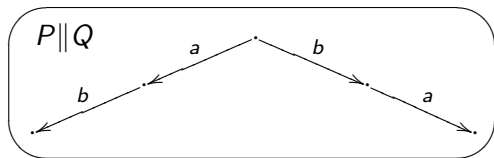
## Nondeterministic parallel composition



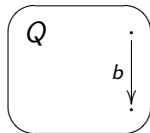
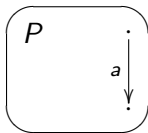
# Nondeterministic parallel composition



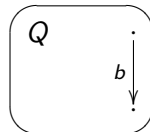
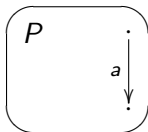
The *interleaving* axiom:



## Probabilistic parallel composition



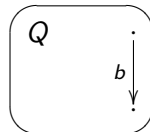
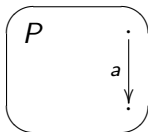
## Probabilistic parallel composition



What is a *probabilistic* behavior of  $P \parallel Q$ ?

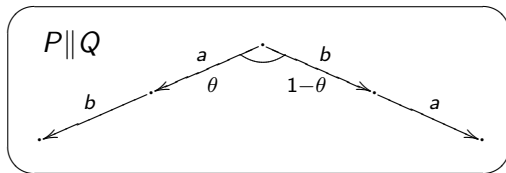


## Probabilistic parallel composition



What is a *probabilistic* behavior of  $P \parallel Q$ ?

Quick answer: *bias factor*  $\theta$ .



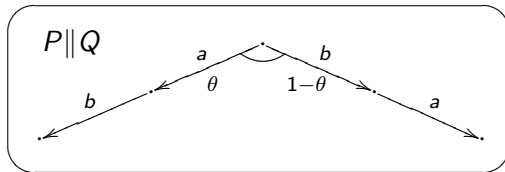
# Probabilistic parallel composition



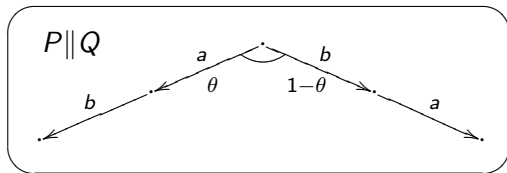
What is a *probabilistic* behavior of  $P \parallel Q$ ?

Quick answer: *bias factor*  $\theta$ .

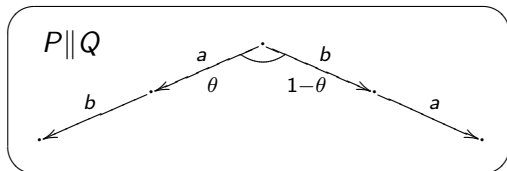
Imagine a **coin-flipping daemon**.



What is the value of  $\theta$ ?

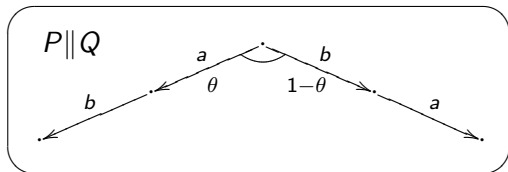


What is the value of  $\theta$ ?



Fixed  $\theta$ : *parameterized* composition operator  $\parallel^\theta$ .

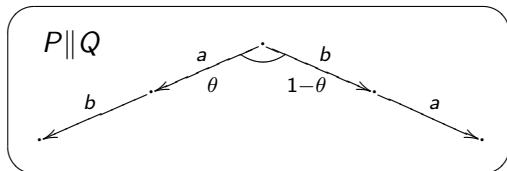
What is the value of  $\theta$ ?



Fixed  $\theta$ : *parameterized* composition operator  $\parallel^\theta$ .

Limitations: static parameter, **not** commutative, **not** associative.

## What is the value of $\theta$ ?



Fixed  $\theta$ : *parameterized* composition operator  $\parallel^\theta$ .

Limitations: static parameter, **not** commutative, **not** associative.

Variable  $\theta$ :

- a supply of coins with different biases;
- imaginary daemon chooses a coin based on his knowledge.

## How much does the daemon know?

There are two scenarios:

## How much does the daemon know?

There are two scenarios:

Scenario 1: *context-independent*



## How much does the daemon know?

There are two scenarios:

Scenario 1: *context-independent*



Scenario 2: *context-dependent*



## Scenario 1: context-independent composition

How much does the daemon know?

Daemon,  $P$  and  $Q$  all inside a big black box.



## Scenario 1: context-independent composition

How much does the daemon know?

Daemon,  $P$  and  $Q$  all inside a big black box.



Daemon knows the histories of  $P$  and  $Q$ , but **nothing** about the outside world.

## Scenario 1: context-independent composition

How much does the daemon know?

Daemon,  $P$  and  $Q$  all inside a big black box.

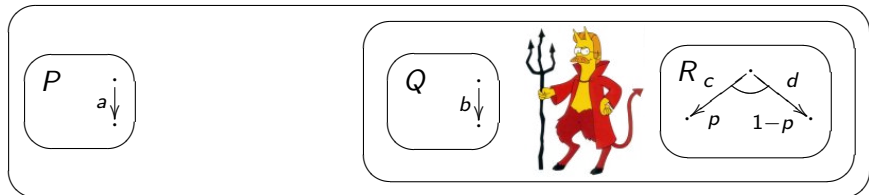


Daemon knows the histories of  $P$  and  $Q$ , but **nothing** about the outside world.

**Problem:** non-associativity.

# Non-associativity: $P \parallel (Q \parallel R)$

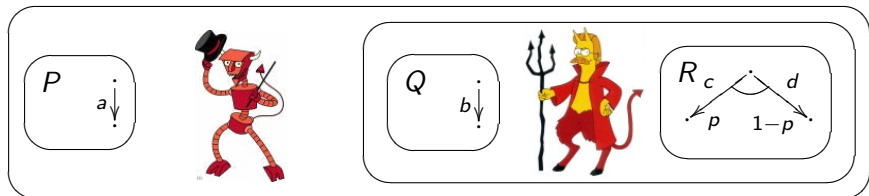
Context-independent composition



*Inner daemon:*  $\langle R, 1 \rangle$ .

# Non-associativity: $P \parallel (Q \parallel R)$

Context-independent composition

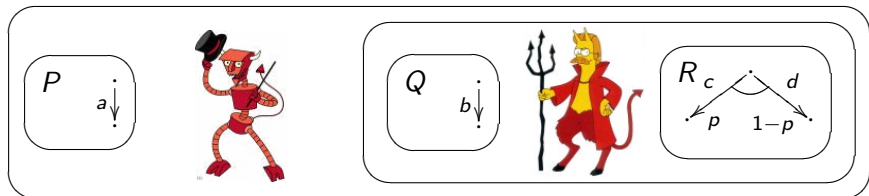


*Inner daemon:*  $\langle R, 1 \rangle$ .

*Outer daemon:*  $\langle Q \parallel R, 1 \rangle$ ; if  $c$ , then  $\langle P, 1 \rangle$ , else  $\langle Q \parallel R, 1 \rangle$ .

# Non-associativity: $P \parallel (Q \parallel R)$

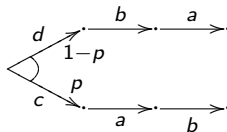
Context-independent composition



*Inner daemon:*  $\langle R, 1 \rangle$ .

*Outer daemon:*  $\langle Q \parallel R, 1 \rangle$ ; if  $c$ , then  $\langle P, 1 \rangle$ , else  $\langle Q \parallel R, 1 \rangle$ .

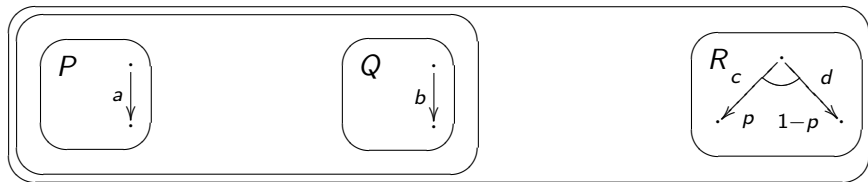
Result:  $\{ \langle cab, p \rangle, \langle dba, 1 - p \rangle \}$ .



# Non-associativity: $(P \parallel Q) \parallel R$

Context-independent composition

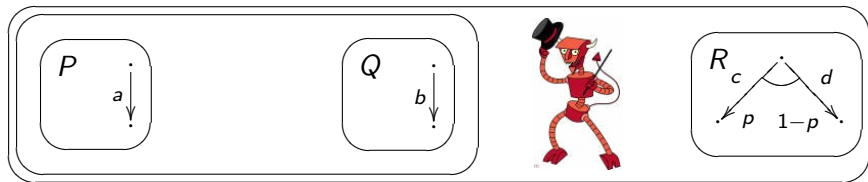
Claim:  $\{\langle cab, p \rangle, \langle dba, 1 - p \rangle\}$  **not** possible!



# Non-associativity: $(P \parallel Q) \parallel R$

Context-independent composition

Claim:  $\{\langle cab, p \rangle, \langle dba, 1 - p \rangle\}$  **not** possible!

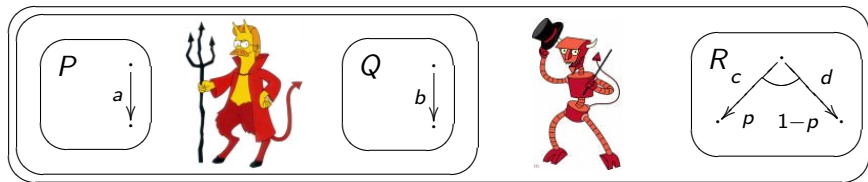


- *Outer daemon*:  $\langle R, 1 \rangle$ .

# Non-associativity: $(P \parallel Q) \parallel R$

Context-independent composition

Claim:  $\{\langle cab, p \rangle, \langle dba, 1 - p \rangle\}$  **not** possible!

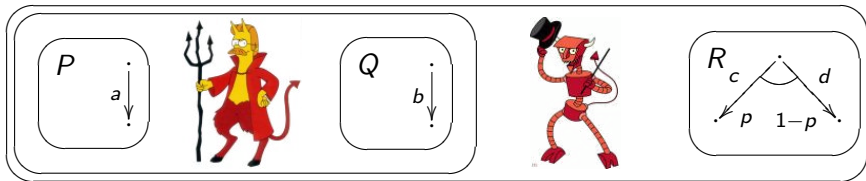


- *Outer* daemon:  $\langle R, 1 \rangle$ .
- *Inner* daemon:  
 $\{\langle P, q \rangle, \langle Q, 1 - q \rangle\}$ .

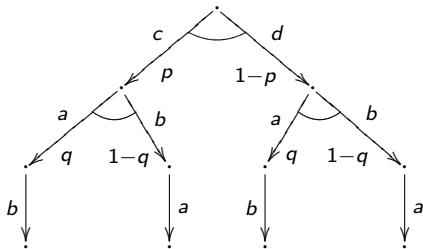
# Non-associativity: $(P \parallel Q) \parallel R$

Context-independent composition

Claim:  $\{\langle cab, p \rangle, \langle dba, 1 - p \rangle\}$  **not** possible!



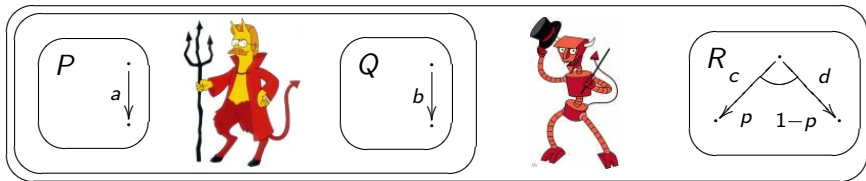
- *Outer daemon*:  $\langle R, 1 \rangle$ .
- *Inner daemon*:  $\{\langle P, q \rangle, \langle Q, 1 - q \rangle\}$ .



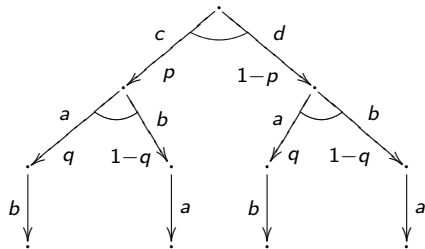
# Non-associativity: $(P \parallel Q) \parallel R$

Context-independent composition

Claim:  $\{\langle cab, p \rangle, \langle dba, 1 - p \rangle\}$  **not** possible!



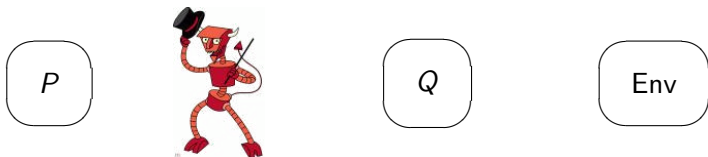
- *Outer* daemon:  $\langle R, 1 \rangle$ .
- *Inner* daemon:  $\{\langle P, q \rangle, \langle Q, 1 - q \rangle\}$ .
- Conclusion: inner daemon **doesn't** know enough.



## Scenario 2: context-dependent composition

How much does the daemon know?

Daemon sees the outside world.



## Scenario 2: context-dependent composition

How much does the daemon know?

Daemon sees the outside world.

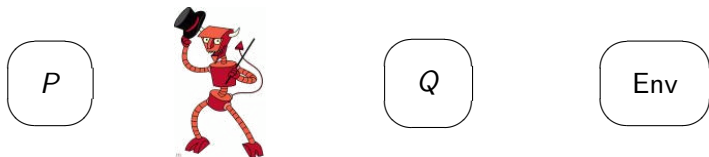


Daemon knows the histories of  $P$ ,  $Q$  and  $Env$ .

## Scenario 2: context-dependent composition

How much does the daemon know?

Daemon sees the outside world.



Daemon knows the histories of  $P$ ,  $Q$  and  $\text{Env}$ .

**Problem:** violation of the interleaving axiom!

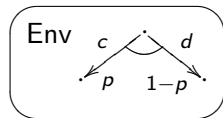
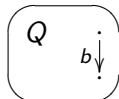
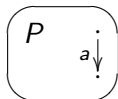
I.e., there exists  $\text{Env}$  such that

$$(a\|b)\| \text{Env} \not\approx (a.b + b.a)\| \text{Env} .$$

# Non-interleaving semantics

Context-dependent composition

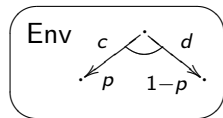
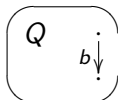
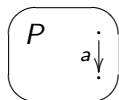
$(a \parallel b) \parallel \text{Env}$ :



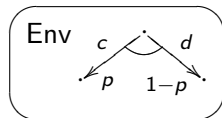
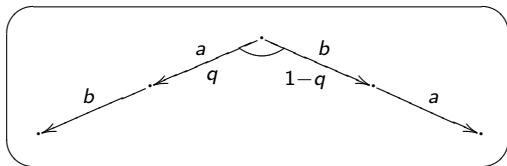
# Non-interleaving semantics

Context-dependent composition

$(a \parallel b) \parallel \text{Env}$ :

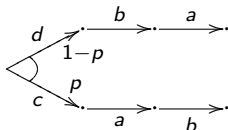


$(a.b + b.a) \parallel \text{Env}$ :



# The (same) counterexample

Context-dependent composition



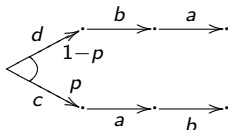
The trace distribution

$$\{\langle cab, p \rangle, \langle dba, 1 - p \rangle\}$$

is possible in  $(a \parallel b) \parallel \text{Env}$ , but **not** in  $(a.b + b.a) \parallel \text{Env}$ .

# The (same) counterexample

Context-dependent composition



The trace distribution

$$\{\langle cab, p \rangle, \langle dba, 1 - p \rangle\}$$

is possible in  $(a \parallel b) \parallel \text{Env}$ , but **not** in  $(a.b + b.a) \parallel \text{Env}$ .

Conclusion: we have a **non-interleaving**, but **total order** semantics.

## What's wrong?

Something is wrong with our understanding of parallel composition.

## What's wrong?

Something is wrong with our understanding of parallel composition.

In context-independent composition, the problem shows up as non-associativity.

## What's wrong?

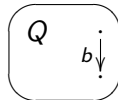
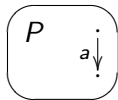
Something is wrong with our understanding of parallel composition.

In context-independent composition, the problem shows up as non-associativity.

In context-dependent composition, the **same** problem leads to difference between  $a \parallel b$  and  $a.b + b.a$ .

## Two types of nondeterministic choices: global vs. local

**Global** choice:  $a||b$ , resolved by a daemon.

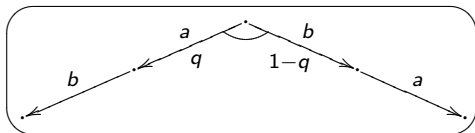


## Two types of nondeterministic choices: global vs. local

**Global** choice:  $a||b$ , resolved by a daemon.



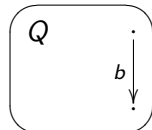
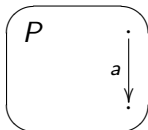
**Local** choice:  $a.b + b.a$ , resolved by a local scheduler.



**Behavior varies depending on the perspective!**

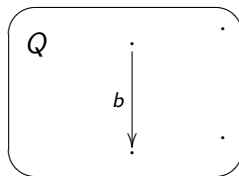
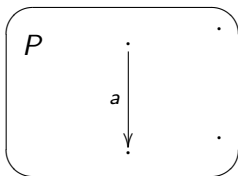
# Dissecting the problem, Part I: eliminate global choices.

To better understand the problem, we developed the model of *Switched PIOA*.



# Dissecting the problem, Part I: eliminate global choices.

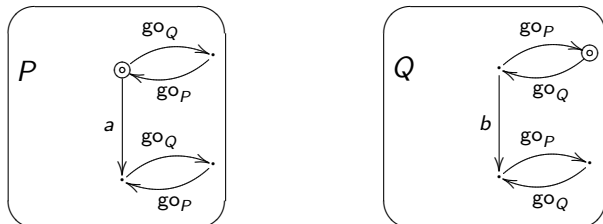
To better understand the problem, we developed the model of *Switched PIOA*.



- **active** states (foreground) vs. **inactive** states (background);

# Dissecting the problem, Part I: eliminate global choices.

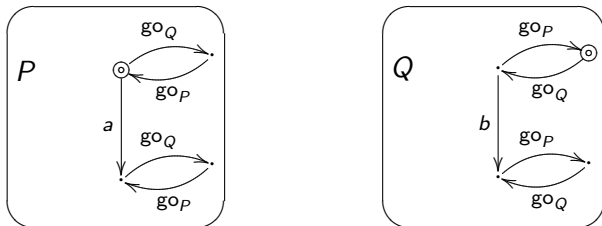
To better understand the problem, we developed the model of *Switched PIOA*.



- **active** states (foreground) vs. **inactive** states (background);
- **control exchange** via special actions (e.g.  $go_P$ ,  $go_Q$ );

# Dissecting the problem, Part I: eliminate global choices.

To better understand the problem, we developed the model of *Switched PIOA*.



- **active** states (foreground) vs. **inactive** states (background);
- **control exchange** via special actions (e.g.  $go_P$ ,  $go_Q$ );

Every decision is made locally, so no more daemons.

## Due to the absence of global choices ...

Parallel composition in Switched PIOA:

- easy to define;

## Due to the absence of global choices . . .

Parallel composition in Switched PIOA:

- easy to define;
- **commutative** and **associative**;

## Due to the absence of global choices . . .

Parallel composition in Switched PIOA:

- easy to define;
- **commutative** and **associative**;
- **deep/semantic** compositionality of trace distribution semantics;

## Due to the absence of global choices ...

Parallel composition in Switched PIOA:

- easy to define;
- **commutative** and **associative**;
- **deep/semantic** compositionality of trace distribution semantics;

That's all very nice, but parallel processes don't really exchange control ...

## Dissecting the problem, Part II: reintroduce global choices.

- Control-exchange should **not** be taken semantically.

## Dissecting the problem, Part II: reintroduce global choices.

- Control-exchange should **not** be taken semantically.
- Switch PIOA is an **implementation tool** for various composition operators.

## Dissecting the problem, Part II: reintroduce global choices.

- Control-exchange should **not** be taken semantically.
- Switch PIOA is an **implementation tool** for various composition operators.

Examples:

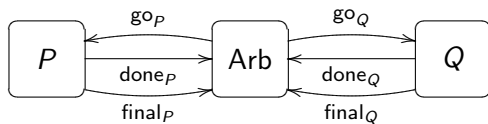
- fixed bias factor  $\theta$ ;

## Dissecting the problem, Part II: reintroduce global choices.

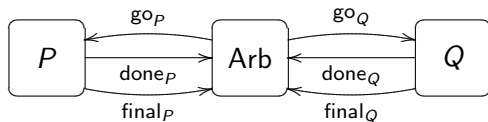
- Control-exchange should **not** be taken semantically.
- Switch PIOA is an **implementation tool** for various composition operators.

Examples:

- fixed bias factor  $\theta$ ;
- context-independent.

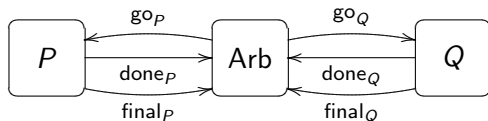
Implementing **biased** composition

# Implementing **biased** composition



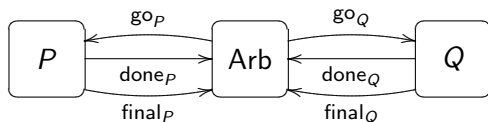
- Local schedulers: always return control after **one** local move.

# Implementing **biased** composition



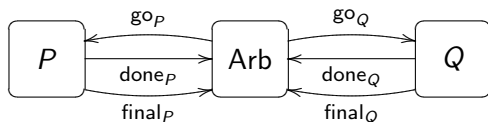
- Local schedulers: always return control after **one** local move.
- Arbiter:
  - usually schedule  $\{\langle go_P, \theta \rangle, \langle go_Q, 1 - \theta \rangle\}$ ;

# Implementing biased composition



- Local schedulers: always return control after **one** local move.
- Arbiter:
  - usually schedule  $\{\langle \text{go}_P, \theta \rangle, \langle \text{go}_Q, 1 - \theta \rangle\}$ ;
  - if  $\text{final}_P$  then  $\langle \text{go}_Q, 1 \rangle$  and vice versa.

# Implementing biased composition

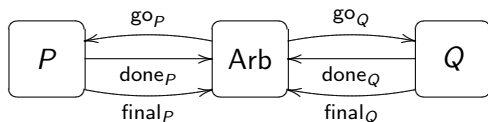


- Local schedulers: always return control after **one** local move.
- Arbiter:
  - usually schedule  $\{\langle go_P, \theta \rangle, \langle go_Q, 1 - \theta \rangle\}$ ;
  - if  $final_P$  then  $\langle go_Q, 1 \rangle$  and vice versa.

Examples:

- $\langle go_P . a . final_P . go_Q . b . final_Q, \theta \rangle$ ;

# Implementing biased composition

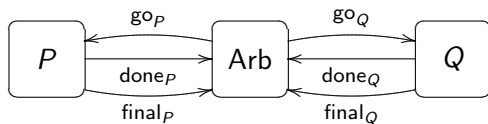


- Local schedulers: always return control after **one** local move.
- Arbiter:
  - usually schedule  $\{\langle \text{go}_P, \theta \rangle, \langle \text{go}_Q, 1 - \theta \rangle\}$ ;
  - if  $\text{final}_P$  then  $\langle \text{go}_Q, 1 \rangle$  and vice versa.

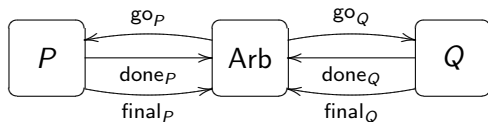
Examples:

- $\langle \text{go}_P . a . \text{final}_P . \text{go}_Q . b . \text{final}_Q, \theta \rangle$ ;
- $\langle \text{go}_Q . b . \text{final}_Q . \text{go}_P . a . \text{final}_P, 1 - \theta \rangle$ .

## Implementing context-independent composition

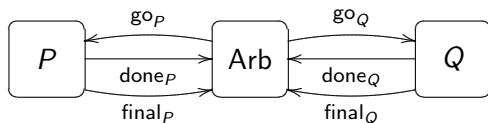


## Implementing context-independent composition



- Local schedulers: no scheduling restrictions (“run to completion”).

## Implementing context-independent composition



- Local schedulers: no scheduling restrictions (“run to completion”).
- Arbiter: if  $\text{final}_P$  then  $\langle \text{go}_Q, 1 \rangle$  and vice versa.

## To summarize ...

- Parallel composition is trickier than we thought.

## To summarize ...

- Parallel composition is trickier than we thought.
- Switched PIOA is a probabilistic model without global choices.

## To summarize ...

- Parallel composition is trickier than we thought.
- Switched PIOA is a probabilistic model without global choices.
- Parallel composition in Switched PIOA is well-behaved.

## To summarize ...

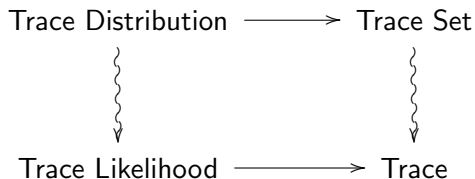
- Parallel composition is trickier than we thought.
- Switched PIOA is a probabilistic model without global choices.
- Parallel composition in Switched PIOA is well-behaved.
- Switched PIOA can be used to study various “real” parallel composition operators.

## Future work

- Philosophical: is there a “most intuitive” parallel composition operator?

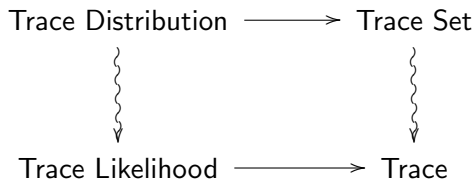
## Future work

- Philosophical: is there a “most intuitive” parallel composition operator?
- Technical: “decomposing” trace distribution semantics.



## Future work

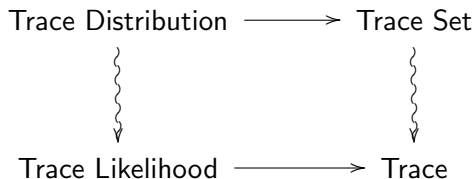
- Philosophical: is there a “most intuitive” parallel composition operator?
- Technical: “decomposing” trace distribution semantics.



- Practical: modeling communication and/or security protocols in Switched PIOA.

## Future work

- Philosophical: is there a “most intuitive” parallel composition operator?
- Technical: “decomposing” trace distribution semantics.



- Practical: modeling communication and/or security protocols in Switched PIOA.

– End –