Switched Probabilistic I/O Automata

Ling Cheung\textsuperscript{1} \hspace{0.5cm} Nancy Lynch\textsuperscript{2} \hspace{0.5cm} Roberto Segala\textsuperscript{3} \hspace{0.5cm} Frits Vaandrager\textsuperscript{1}

\textsuperscript{1}Radboud University Nijmegen, the Netherlands
\textsuperscript{2}MIT Computer Science and Artificial Intelligence Laboratory, U.S.A.
\textsuperscript{3}Dipartimento di Informatica, Università di Verona, Italy

FMCO 2004, Leiden, the Netherlands
Outline

1 Introduction
   • Basics
   • Motivation
Outline

1. Introduction
   - Basics
   - Motivation

2. The trouble with composition
   - How should we compose probabilistic behaviors?
   - How much does the daemon know?
   - Global choice vs local choice
Outline

1. Introduction
   - Basics
   - Motivation

2. The trouble with composition
   - How should we compose probabilistic behaviors?
   - 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
   - Motivation

2. The trouble with composition
   - How should we compose probabilistic behaviors?
   - 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
Probabilistic Automata

A probabilistic automaton is a tuple $A = (Q, \bar{q}, A, D)$, where

- $Q$ is a set of states,
Probabilistic Automata

A probabilistic automaton is a tuple \( A = (Q, \bar{q}, A, D) \), where

- \( Q \) is a set of states,
- \( \bar{q} \in Q \) is a start state,
Probabilistic Automata

A probabilistic automaton is a tuple $A = (Q, \bar{q}, A, D)$, where

- $Q$ is a set of states,
- $\bar{q} \in Q$ is a start state,
- $A$ is a set of actions,
A probabilistic automaton is a tuple $A = (Q, \bar{q}, A, D)$, where

- $Q$ is a set of states,
- $\bar{q} \in Q$ is a start state,
- $A$ is a set of actions,
- $D \subseteq Q \times A \times Disc(Q)$ is a transition relation.
A probabilistic automaton is a tuple $\mathcal{A} = (Q, \bar{q}, A, D)$, where

- $Q$ is a set of states,
- $\bar{q} \in Q$ is a start state,
- $A$ is a set of actions,
- $D \subseteq Q \times A \times \text{Disc}(Q)$ is a transition relation.

Notion has been used successfully for specification/verification of distributed algorithms and protocols.
Composition

The composition of two PAs $\mathcal{A}_1, \mathcal{A}_2$ is the PA

$$\mathcal{A} = (Q_1 \times Q_2, (\bar{q}_1, \bar{q}_2), A_1 \cup A_2, D)$$

where $D$ is the set of triples $(q, a, \mu_1 \times \mu_2)$ such that, for $i \in \{1, 2\}$:

$$a \in A_i \Rightarrow (\pi_i(q), a, \mu_i) \in D_i \text{ and } a \notin A_i \Rightarrow \mu_i = \text{Dirac}(\pi_i(q)).$$
The composition of two PAs $A_1, A_2$ is the PA

$$A = (Q_1 \times Q_2, (\bar{q}_1, \bar{q}_2), A_1 \cup A_2, D)$$

where $D$ is the set of triples $(q, a, \mu_1 \times \mu_2)$ such that, for $i \in \{1, 2\}$:

$$a \in A_i \Rightarrow (\pi_i(q), a, \mu_i) \in D_i \text{ and } a \notin A_i \Rightarrow \mu_i = Dirac(\pi_i(q)).$$

Probabilistic simulation relations induce a precongruence for parallel composition (Segala, 1995)
The Quest for Compositionality

Our goal is to find an

- abstract, trace based, compositional semantics for probabilistic (I/O) automata,
The Quest for Compositionality

Our goal is to find an

- abstract, trace based, compositional semantics for probabilistic (I/O) automata,
- that can be justified via some intuitive testing scenario along the lines of Stoelinga & Vaandrager (ICALP03), and
The Quest for Compositionality

Our goal is to find an

- abstract, trace based, compositional semantics for probabilistic (I/O) automata,
- that can be justified via some intuitive testing scenario along the lines of Stoelinga & Vaandrager (ICALP03), and
- conservatively extends trace semantics for nondeterministic automata.
Schedulers and trace distributions

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

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

\[
\{\langle a, pq \rangle, \langle ab, p(1-q) \rangle, \langle b, 1-p \rangle\}
\]
Schedulers and trace distributions

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

![Diagram of scheduling choices and probabilities](image-url)
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.

\[
\{\langle aa, pq \rangle, \langle ab, p(1-q) \rangle, \langle b, 1-p \rangle \}\]
Trace distribution semantics not compositional

Probabilistic simulation is coarsest precongruence that refines trace distribution preorder (Lynch, Segala & Vaandrager, CONCUR03).
Trace distribution semantics not compositional

Probabilistic simulation is coarest precongruence that refines trace distribution preorder (Lynch, Segala & Vaandrager, CONCUR03).
Nondeterministic parallel composition
Nondeterministic parallel composition

The *interleaving* axiom:
What is a probabilistic behavior of $P \parallel Q$?

Quick answer: bias factor $\theta$. Imagine a coin-flipping daemon.
Probabilistic parallel composition

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

Quick answer: bias factor $\theta$.

Imagine a coin-flipping daemon.

**FMCO 2004**

Switched Probabilistic I/O Automata
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 $\theta$?

$P \parallel Q$

$\theta$ $1-\theta$
What is the value of $\theta$?

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

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 (observable) 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 (observable) 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: $\langle R, 1 \rangle$. 

**FMCO 2004**

Switched Probabilistic I/O Automata
Non-associativity: $P\|(Q\|R)$

Context-independent composition

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

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

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: \(\{\langle cab, p \rangle, \langledba, 1-p \rangle\}\) not possible!

- **Outer daemon:** \(\langle R, 1 \rangle\).
- **Inner daemon:** 
  \(\{\langle P, q \rangle, \langle Q, 1-q \rangle\}\).
Non-associativity: \((P ∥ Q) ∥ R\)

Context-independent composition

Claim: \(\{⟨cab, p⟩, ⟨dba, 1−p⟩\}\) not possible!

- **Outer daemon:** \(⟨R, 1⟩\).
- **Inner daemon:** \(\{⟨P, q⟩, ⟨Q, 1−q⟩\}\).
- Conclusion: inner daemon doesn’t know enough.
Scenario 2: context-dependent composition
How much does the daemon know?

Daemon sees the outside world.

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

Problem: violation of the interleaving axiom!
I.e., there exists $Env$ such that $(a \parallel b) \parallel Env \not\sim (a. b + b. a) \parallel Env$. 
Scenario 2: context-dependent composition
How much does the daemon know?

Daemon sees the outside world.

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

Problem: violation of the interleaving axiom!
I.e., there exists $Env$ such that

$((a \parallel b) \parallel Env) \neq (a \cdot b + b \cdot a) \parallel Env$. 

FMCO 2004 Switched Probabilistic I/O Automata
Scenario 2: context-dependent composition
How much does the daemon know?

Daemon sees the outside world.

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

Problem: violation of the interleaving axiom!
I.e., there exists $Env$ such that

$$(a \parallel b) \parallel Env \not\sim (a.b + b.a) \parallel Env.$$
Non-interleaving semantics
Context-dependent composition

\[(a \parallel b) \parallel \text{Env}:
\]

\[
\begin{array}{c}
\text{P} \\
\begin{array}{c}
\cdot \\
\downarrow a
\end{array}
\end{array}
\begin{array}{c}
\text{Q} \\
\begin{array}{c}
\cdot \\
\downarrow b
\end{array}
\end{array}
\begin{array}{c}
\text{Env} \\
\begin{array}{c}
\cdot \\
\downarrow c \\
\ wedge \\
\cdot \\
\downarrow d \\
\quad \\
\cdot \\
\quad \\
\downarrow p \\
\quad \\
\cdot \\
\quad \\
\downarrow 1-p
\end{array}
\end{array}
\end{array}
\]
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

\[
\begin{array}{c}
\bullet \\
\downarrow p \\
\bullet \\
\downarrow 1-p \\
\bullet
\end{array}
\]

\[d \rightarrow b \rightarrow a \rightarrow.
\]

\[c \rightarrow a \rightarrow b \rightarrow.
\]

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:** $a \parallel b$, resolved by a daemon.

![Diagram showing two automata $P$ and $Q$ with nondeterministic choices $a$ and $b$](image)
Two types of nondeterministic choices: global vs. local

Global choice: $a \parallel b$, resolved by a daemon.

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

Behavior varies depending on the perspective!
Dissecting the problem, Part I: eliminate global choices.

Several authors establish compositional trace based semantics for models without global choice:

*Wu, Smolka, Stark (CONCUR94)*
Components draw delay from exponential distribution
Dissecting the problem, Part I: eliminate global choices.

Several authors establish compositional trace based semantics for models without global choice:

Wu, Smolka, Stark (CONCUR94)  
Components draw delay from exponential distribution

De Alfaro, Henzinger & Jhala (CONCUR01)  
Synchronous parallel composition
Dissecting the problem, Part I: eliminate global choices.

Several authors establish compositional trace based semantics for models without global choice:

*Wu, Smolka, Stark (CONCUR94)*
Components draw delay from exponential distribution

*De Alfaro, Henzinger & Jhala (CONCUR01)*
Synchronous parallel composition

*Cheung, Lynch, Segala, Vaandrager (ICTAC04)*
At most one component has control at any time
Switched Probabilistic I/O Automata

To better understand the problem, we developed the model of Switched PIOA.
Switched Probabilistic I/O Automata

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

- active states (foreground) vs. inactive states (background);
Switched Probabilistic I/O Automata

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. $go_P$, $go_Q$);
Switched Probabilistic I/O Automata

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

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 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 *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**

**Diagram:**
- **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 \( \{\langle \text{go}_P, \theta \rangle, \langle \text{go}_Q, 1 - \theta \rangle\} \);
  - if \( \text{final}_P \) then \( \langle \text{go}_Q, 1 \rangle \) and vice versa.

Examples:
- \( \langle \text{go}_P \cdot a. \text{final}_P \cdot \text{go}_Q \cdot b. \text{final}_Q, \theta \rangle; \)
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 \cdot a \cdot \text{final}_P \cdot go_Q \cdot b \cdot \text{final}_Q, \theta \rangle;
- \langle go_Q \cdot b \cdot \text{final}_Q \cdot go_P \cdot a \cdot \text{final}_P, 1 - \theta \rangle.

FMCO 2004 Switched Probabilistic I/O Automata
Implementing context-independent composition

$P \xleftrightarrow{\text{go}_P} \text{Arb} \xleftrightarrow{\text{go}_Q} Q$

Local schedulers: no scheduling restrictions ("run to completion").

Arbiter: if final $P$ then $\langle \text{go}_Q, 1 \rangle$ and vice versa.
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: complete the quest.
Future work

- Philosophical: is there a “most intuitive” parallel composition operator?
- Technical: complete the quest.
- Practical: modeling communication and/or security protocols in Switched PIOA.
Future work

- Philosophical: is there a “most intuitive” parallel composition operator?
- Technical: complete the quest.
- Practical: modeling communication and/or security protocols in Switched PIOA.

– End –