Sound and Complete axiomatization of trace semantics for probabilistic systems

Alexandra Silva\textsuperscript{1}   Ana Sokolova\textsuperscript{2}

\textsuperscript{1}Centrum Wiskunde & Informatica (currently on leave at Cornell Univ.)
\textsuperscript{2}Computer Sciences Department, University of Salzburg

June 2011
Motivation

\[ P ::= 0 \mid a.P \mid P + P \mid \mu x.P^g \]

Kleene-like theorem: behaviours of LTS are characterized by P’s and vice-versa

Axiomatization:
\[ P + Q \equiv Q + P; P + 0 \equiv P; \mu x.P \equiv P[\mu x.P/x]; \ldots \]

Soundness and Completeness:
\[ P \equiv Q \iff P \sim Q \]

$P:: = 0 \mid a.P \mid P + P \mid \mu x.P^g$

**Kleene-like theorem:** behaviours of LTS are characterized by $P$’s and vice-versa

**Axiomatization:**

$P + Q \equiv Q + P; P + 0 \equiv P; \mu x.P \equiv P[\mu x.P/x]; \ldots$

**Soundness and Completeness:**

$P \equiv Q \iff P \sim Q$

Motivation

$P:: = 0 \mid a.P \mid P + P \mid \mu x.P^g$

Kleene-like theorem: behaviours of LTS are characterized by $P$'s and vice-versa

Axiomatization:
$P + Q \equiv Q + P; P + 0 \equiv P; \mu x.P \equiv P[\mu x.P/x]; \ldots$

Soundness and Completeness:
$P \equiv Q \iff P \sim Q$

Motivation

$P ::= 0 \mid a.P \mid P + P \mid \mu x.P^g$

**Kleene-like theorem:** behaviours of LTS are characterized by $P$’s and vice-versa

**Axiomatization:**

$P + Q \equiv Q + P; P + 0 \equiv P; \mu x.P \equiv P[\mu x.P/x]; \ldots$

**Soundness and Completeness:**

$P \equiv Q \iff P \sim Q$

Milner’s language + axiomatization +

\[ a.P + a.Q \equiv a.(P + Q) \]

Soundness and Completeness:

\[ P \equiv Q \iff \text{tr}(P) = \text{tr}(Q) \]
Motivation

Milner’s language + axiomatization +
\[ a.P + a.Q \equiv a.(P + Q) \]

**Soundness and Completeness:**
\[ P \equiv Q \iff tr(P) = tr(Q) \]

Alexander Rabinovich
Segala systems

\[
\begin{array}{c}
\text{a} \\
\frac{1}{2} \\
\frac{1}{2} \\
\frac{2}{3} \\
1/3 \\
1 \\
\end{array}
\]

\[
\begin{array}{c}
\text{a} \\
\frac{1}{2} \\
\frac{1}{2} \\
\frac{2}{3} \\
1/3 \\
1 \\
\end{array}
\]

Catuscia Palamidessi

Yuxin Deng
Probabilistic extensions of Milner’s work

Segala systems

\[ E :: = 0 \mid E \oplus E \mid \mu x.E \mid x \mid a \cdot E' \]
\[ E' :: = \bigoplus_{i \in 1 \ldots n} p_i \cdot E_i \]
where \( a \in A, p_i \in (0, 1] \) and \( \sum_{i \in 1 \ldots n} p_i = 1 \)
Probabilistic extensions of Milner’s work

Segala systems

\[
a \cdot (1/2 \cdot 0 \oplus 1/2 \cdot 0) \boxplus a \cdot (1/3 \cdot 0 \oplus 2/3 \cdot 0) \boxplus b \cdot 1 \cdot 0
\]

\[
E ::= 0 \mid E \boxplus E \mid \mu x. E \mid x \mid a \cdot E'
\]

\[
E' ::= \bigoplus_{i \in 1 \ldots n} p_i \cdot E_i
\]

where \( a \in A \), \( p_i \in (0, 1] \) and \( \sum_{i \in 1 \ldots n} p_i = 1 \)
Probabilistic extensions of Milner’s work

Segala systems

\[
\begin{align*}
a \cdot (1/2 \cdot 0 \oplus 1/2 \cdot 0) & \oplus a \cdot (1/3 \cdot 0 \oplus 2/3 \cdot 0) \oplus b \cdot 1 \cdot 0 \\
E :: = & \ 0 \mid E \boxplus E \mid \mu x.E \mid x \mid a \cdot E' \\
E' :: = & \ \bigoplus_{i \in 1\ldots n} p_i \cdot E_i \\
\text{where } a & \in A, \ p_i \in (0,1] \text{ and } \sum_{i \in 1\ldots n} p_i = 1
\end{align*}
\]

\[
(E_1 \boxplus E_2) \boxplus E_3 \equiv E_1 \boxplus (E_2 \boxplus E_3) \\
\vdots \\
(p_1 \cdot E) \oplus (p_2 \cdot E) \equiv (p_1 + p_2) \cdot E
\]
(p_1 \cdot E) \oplus (p_2 \cdot E) \equiv (p_1 + p_2) \cdot E

a \cdot (1/2 \cdot 0 \oplus 1/2 \cdot 0) \boxplus a \cdot (1/3 \cdot 0 \oplus 2/3 \cdot 0) \boxplus b \cdot 1 \cdot 0
\[(p_1 \cdot E) \oplus (p_2 \cdot E) \equiv (p_1 + p_2) \cdot E\]

\[
a \cdot (1/2 \cdot 0 \oplus 1/2 \cdot 0) \oplus a \cdot (1/3 \cdot 0 \oplus 2/3 \cdot 0) \oplus b \cdot 1 \cdot 0
\equiv
a \cdot (1/2 + 1/2) \cdot 0 \oplus a \cdot (2/3 + 1/3) \cdot 0 \oplus b \cdot 1 \cdot 0
\]
\[(p_1 \cdot E) \oplus (p_2 \cdot E) \equiv (p_1 + p_2) \cdot E\]

\[a \cdot (1/2 \cdot 0 \oplus 1/2 \cdot 0) \boxplus a \cdot (1/3 \cdot 0 \oplus 2/3 \cdot 0) \boxplus b \cdot 1 \cdot 0 \equiv a \cdot (1/2 + 1/2) \cdot 0 \boxplus a \cdot (2/3 + 1/3) \cdot 0 \boxplus b \cdot 1 \cdot 0 \equiv a \cdot 1 \cdot 0 \boxplus a \cdot 1 \cdot 0 \boxplus b \cdot 1 \cdot 0\]
Extensions of Milner’s work uniformly to a large class of systems including Segala systems, generative systems, alternating systems, . . .

Key idea: $S \rightarrow GS$

The type $G$ is enough to derive:

1. canonical notion of equivalence (bisimilarity)
2. syntax
3. sound and complete axiomatizations
Extensions of Milner’s work uniformly to a large class of systems including Segala systems, generative systems, alternating systems, . . .

Key idea: $S \rightarrow GS$

The type $G$ is enough to derive:

1. canonical notion of equivalence (bisimilarity)
2. syntax
3. sound and complete axiomatizations
How about trace?

Many people think that bisimilarity is not the right equivalence. . .

. . . and that trace equivalence is more appropriate to reason about systems.

\[
\begin{align*}
\bullet & \xrightarrow{\frac{1}{2}} \bullet & \xrightarrow{\frac{1}{3}} \bullet & \xrightarrow{1} * \\
\bullet & \xrightarrow{\frac{1}{3}} \bullet & \xrightarrow{\frac{1}{2}} \bullet & \xrightarrow{1} * 
\end{align*}
\]

1. How far can Rabinovich’s method be extended?

2. Can we extend sound and complete axiomatizations of probabilistic systems for bisimilarity to trace equivalence?
How about trace?

Many people think that bisimilarity is not the right equivalence . . .
. . . and that trace equivalence is more appropriate to reason about systems.

\[ \bullet \xrightarrow{a, \frac{1}{2}} \bullet \xrightarrow{b, \frac{1}{3}} \bullet \xrightarrow{1} \ast \]

\[ \bullet \xrightarrow{a, \frac{1}{3}} \bullet \xrightarrow{b, \frac{1}{2}} \bullet \xrightarrow{1} \ast \]

1. How far can Rabinovich’s method be extended?

2. Can we extend sound and complete axiomatizations of probabilistic systems for bisimilarity to trace equivalence?
How about trace?

Many people think that bisimilarity is not the right equivalence. . .
. . . and that trace equivalence is more appropriate to reason about systems.

\begin{align*}
\bullet & \xrightarrow{a, \frac{1}{2}} \bullet & \xrightarrow{b, \frac{1}{3}} & \bullet & \xrightarrow{1} & * \\
\bullet & \xrightarrow{a, \frac{1}{3}} & \bullet & \xrightarrow{b, \frac{1}{2}} & \bullet & \xrightarrow{1} & * 
\end{align*}

1. How far can Rabinovich’s method be extended?

2. Can we extend sound and complete axiomatizations of probabilistic systems for bisimilarity to trace equivalence?
How about trace?

Many people think that bisimilarity is not the right equivalence... and that trace equivalence is more appropriate to reason about systems.

\[ \bullet \xrightarrow{a, \frac{1}{2}} \bullet \xrightarrow{b, \frac{1}{3}} \bullet \xrightarrow{1} \ast \]

\[ \bullet \xrightarrow{a, \frac{1}{3}} \bullet \xrightarrow{b, \frac{1}{2}} \bullet \xrightarrow{1} \ast \]

1. How far can Rabinovich’s method be extended?

2. Can we extend sound and complete axiomatizations of probabilistic systems for bisimilarity to trace equivalence?
First observation: this is a general phenomenon!

Theorem (Bonsangue&Milius&Silva 2011)

Sound and complete axiomatizations for bisimilarity can always be extended to sound and complete axiomatizations for trace semantics.

The theorem is valid for a large class of systems including LTS and weighted automata, but...not for probabilistic systems.

This talk: the method also works for probabilistic systems!
First observation: this is a general phenomenon!

**Theorem (Bonsangue&Milius&Silva 2011)**

*Sound and complete axiomatizations for bisimilarity can always be extended to sound and complete axiomatizations for trace semantics.*

The theorem is valid for a large class of systems including LTS and weighted automata, but... not for probabilistic systems.

**This talk:** the method also works for probabilistic systems!
Generative systems

This talk will be about one type of probabilistic systems: generative systems.

\((S, \alpha : X \rightarrow D_\omega(1 + A \times X))\)

- \(x \xrightarrow{p} \ast \) if \(\alpha(x)(\ast) = p\),
  i.e., \(x\) successfully terminates with probability \(p\), and

- \(x \xrightarrow{a,p} y \) if \(\alpha(x)(a, y) = p\),
  i.e., if \(x\) can make an \(a\)-labelled step to \(y\) with weight \(p\).
This talk will be about one type of probabilistic systems: generative systems.

\[(S, \alpha: \mathbb{X} \rightarrow \mathcal{D}_\omega(1 + A \times \mathbb{X}))\]

\[x \xrightarrow{p} * \quad \text{if} \quad \alpha(x)(*) = p,\]

i.e., \(x\) successfully terminates with probability \(p\), and

\[x \xrightarrow{a,p} y \quad \text{if} \quad \alpha(x)(a, y) = p,\]

i.e., if \(x\) can make an \(a\)-labelled step to \(y\) with weight \(p\).
Starting point

\[ E := \bigoplus_{i \in I} p_i \cdot F_i \mid \mu x. E^g \mid x \quad (p_i \in [0, 1], \sum_{i \in I} p_i \leq 1) \]

\[ E^g := \bigoplus_{i \in I} p_i \cdot F_i \mid \mu x. E^g \quad (p_i \in [0, 1], \sum_{i \in I} p_i \leq 1) \]

\[ F_i := \ast \mid a \cdot E \]

Bonchi et al. 2009

There is a sound and complete axiomatization w.r.t. \( \sim \).
Starting point

\[ E ::= \bigoplus_{i \in I} p_i \cdot F_i \mid \mu x.E^g \mid x \quad (p_i \in [0, 1], \sum_{i \in I} p_i \leq 1) \]

\[ E^g ::= \bigoplus_{i \in I} p_i \cdot F_i \mid \mu x.E^g \quad (p_i \in [0, 1], \sum_{i \in I} p_i \leq 1) \]

\[ F_i ::= \ast \mid a \cdot E \]

**Bonchi et al. 2009**

There is a sound and complete axiomatization w.r.t. \( \sim \).
Examples

\[
\begin{align*}
\text{(1)} & \quad a, \frac{1}{2} \quad \bullet \quad a, \frac{1}{4} \\
& \quad \quad \bullet \quad \bullet \\
& \quad b, \frac{1}{3} \downarrow \quad \downarrow c, \frac{1}{2} \\
& \quad 1 \downarrow \quad 1 \\
& \quad * \quad * \\
\frac{1}{2} \cdot a \cdot \frac{1}{3} \cdot b \cdot 1 \cdot * \oplus \frac{1}{4} \cdot a \cdot \frac{1}{2} \cdot c \cdot 1 \cdot *
\end{align*}
\]
Examples

Top states are not bisimilar but they are trace equivalent.

\[
\frac{1}{2} \cdot a \cdot \frac{1}{3} \cdot b \cdot 1 \cdot * \oplus \frac{1}{4} \cdot a \cdot \frac{1}{2} \cdot c \cdot 1 \cdot * \quad \frac{1}{2} \cdot a \cdot (\frac{1}{3} \cdot b \cdot 1 \cdot * \oplus \frac{1}{4} \cdot c \cdot 1 \cdot *)
\]
Top states are not bisimilar but they are trace equivalent.

\[ ab \leftrightarrow \frac{1}{6}; \quad ab \leftrightarrow \frac{1}{8} \]
Axiomatization

all the axioms for \( \sim \)

\[
p \cdot a \cdot (p_1 E_1 \oplus p_2 E_2) \equiv p_1 \cdot a \cdot pE_1 \oplus p_2 \cdot a \cdot pE_2 \quad (D)
\]
Part of (D) is about multiplying probabilities

We define a notion of *scalar product* for expressions:

\[ p \left( \bigoplus_{i \in I} p_i \cdot F_i \right) = \bigoplus_{i \in I} (pp_i) \cdot F_i \]

\[ \frac{1}{2} \cdot a \cdot \frac{1}{3} \cdot b \cdot 1 \cdot * \equiv \frac{1}{3} \cdot \frac{3}{2} (a \cdot \frac{1}{3} \cdot b \cdot 1 \cdot *) = \frac{1}{3} \cdot a \cdot \frac{1}{2} \cdot b \cdot 1 \cdot * \]
Part of (D) is about multiplying probabilities

We define a notion of *scalar product* for expressions:

\[ p \left( \bigoplus_{i \in I} p_i \cdot F_i \right) = \bigoplus_{i \in I} (pp_i) \cdot F_i \]

Silva&Sokolova (CWI&USalz)  Sound and Complete axiomatization of trace semantics for probabilistic systems  June 2011  13 / 20
(D) is also about eliminating branching

\[
\begin{align*}
(\frac{1}{2} \cdot a \cdot \frac{1}{3} \cdot b \cdot 1 \cdot *) & \oplus (\frac{1}{4} \cdot a \cdot \frac{1}{2} \cdot c \cdot 1 \cdot *) & (\frac{1}{2} \cdot a \cdot (\frac{1}{3} \cdot b \cdot 1 \cdot *) & \oplus \frac{1}{4} \cdot c \cdot 1 \cdot * \\
\end{align*}
\]

\[
\equiv
\]

\[
\frac{1}{2} \cdot a \cdot (\frac{1}{2} (\frac{2}{3} \cdot b \cdot 1 \cdot *) \oplus \frac{1}{4} (1 \cdot c \cdot 1 \cdot *)
\]
Soundness and completeness proofs often boil down to find normal forms;

Rabinovich’s proof uses the fact that every finite LTS can be changed to a finite trace-equivalent LTS that is deterministic.

This is not so trivial for probabilistic systems: for a finite system, there may be no finite deterministic system that is trace equivalent to it.

We will use an (infinite) determinization of a probabilistic transition system but avoid reasoning about normal forms by using a coinductive approach.
Soundness and completeness proofs often boil down to find normal forms;

Rabinovich’s proof uses the fact that every finite LTS can be changed to a finite trace-equivalent LTS that is deterministic.

This is not so trivial for probabilistic systems: for a finite system, there may be no finite deterministic system that is trace equivalent to it.

We will use an (infinite) determinization of a probabilistic transition system but avoid reasoning about normal forms by using a coinductive approach.
Soundness and completeness proofs often boil down to find *normal forms*;

Rabinovich’s proof uses the fact that every finite LTS can be changed to a finite trace-equivalent LTS that is deterministic.

This is not so trivial for probabilistic systems: for a finite system, there may be no finite deterministic system that is trace equivalent to it.

We will use an (infinite) determinization of a probabilistic transition system but avoid reasoning about normal forms by using a coinductive approach.
General strategy

- Determinization

\[
\begin{align*}
X & \xrightarrow{\eta_X} \mathcal{D}_\omega(X) \\
\mathcal{D}_\omega(1 + A \times X) & \xrightarrow{(\delta \circ \alpha)^\#} \\
[0, 1] \times \mathcal{D}_\omega(X)^A
\end{align*}
\]
General strategy

- Determinization and semantics by finality

\[
\begin{align*}
X \xrightarrow{\eta_X} \mathcal{D}_\omega(X) &\xrightarrow{\text{out}} [0, 1]^{A^*} \\
\mathcal{D}_\omega(1 + A \times X) &\xrightarrow{(\delta \circ \alpha)^\sharp} [0, 1] \times \mathcal{D}_\omega(X)^A &\xrightarrow{id \times \text{out}^A} [0, 1] \times ([0, 1]^{A^*})^A
\end{align*}
\]
Theorem

For any $x \in X$, $tr(x) = out(\eta(x))$. For $E \in \text{Exp}$, $tr(x) = out_{\equiv}([E])$

This actually means that the image of $out$ is a distribution on words.
# Soundness and completeness

<table>
<thead>
<tr>
<th>Soundness</th>
<th>Completeness</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_1 \equiv E_2$</td>
<td>$E_1 \sim_{\text{tr}} E_2$</td>
</tr>
<tr>
<td>$\iff [E_1] = [E_2]$</td>
<td>$\iff \text{tr}(E_1) = \text{tr}(E_2)$</td>
</tr>
<tr>
<td>$(\star)$: $\text{out}<em>\equiv([E_1]) = \text{out}</em>\equiv([E_2])$</td>
<td>$(\star)$: $\text{out}<em>\equiv([E_1]) = \text{out}</em>\equiv([E_2])$</td>
</tr>
<tr>
<td>$(\triangle)$: $\text{tr}(E_1) = \text{tr}(E_2)$</td>
<td>$(\triangle)$: $[E_1] = [E_2]$</td>
</tr>
<tr>
<td>$\iff E_1 \sim_{\text{tr}} E_2$</td>
<td>$\iff E_1 \equiv E_2$</td>
</tr>
</tbody>
</table>

$(\star)$: existence of $\text{out}_\equiv$, $(\triangle)$: $\text{out}_\equiv \circ [-] = \text{tr}$, $(\heartsuit)$: $\text{out}_\equiv$ is injective.

The proof of $(\heartsuit)$ is where the difficulties arose.
### Soundness and completeness

<table>
<thead>
<tr>
<th>Soundness</th>
<th>Completeness</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E_1 \equiv E_2 )</td>
<td>( E_1 \sim_{\text{tr}} E_2 )</td>
</tr>
<tr>
<td>( [E_1] = [E_2] )</td>
<td>( \text{tr}(E_1) = \text{tr}(E_2) )</td>
</tr>
<tr>
<td>( \text{out} \equiv ([E_1]) = \text{out} \equiv ([E_2]) )</td>
<td>( \text{out} \equiv ([E_1]) = \text{out} \equiv ([E_2]) )</td>
</tr>
<tr>
<td>( \text{tr}(E_1) = \text{tr}(E_2) )</td>
<td>( [E_1] = [E_2] )</td>
</tr>
<tr>
<td>( E_1 \sim_{\text{tr}} E_2 )</td>
<td>( E_1 \equiv E_2 )</td>
</tr>
</tbody>
</table>

\((\star)\): existence of \( \text{out} \equiv \), \((\bigtriangleup)\): \( \text{out} \equiv \circ [-] = \text{tr} \), \((\heartsuit)\): \( \text{out} \equiv \) is injective.

The proof of \((\heartsuit)\) is where the difficulties arose.
Conclusions and Future work

Conclusions

- First sound and complete axiomatization of trace semantics of generative systems
- Similarly to Rabinovich

Future Work

- Extend uniformly to other types of systems
- All the proofs are coinductive parametrized by the functor type. The restriction on generalizing lies in the theory of generic coalgebraic trace semantics (Hasuo & Jacobs & Sokolova 2007).
Thank you for your attention!