Analyzing Expected Outcomes and (Positive) Almost-Sure Termination of Probabilistic Programs is Hard

Benjamin Kaminski  Joost-Pieter Katoen

27.2.2015
Motivation

- Probabilistic Programs are like ordinary programs, except:
Motivation

- Probabilistic Programs are like ordinary programs, except:
  - Allow for random choice on how to continue the execution
  - Random choice is done with some specified probability $p$
Motivation

- Probabilistic Programs are like ordinary programs, except:
  - Allow for random choice on how to continue the execution
  - Random choice is done with some specified probability $p$
- **Analysis problems** we consider:
Motivation

- Probabilistic Programs are like ordinary programs, except:
  - Allow for random choice on how to continue the execution
  - Random choice is done with some specified probability $p$

- Analysis problems we consider:
  - Determine the value of a variable after program execution
Motivation

- Probabilistic Programs are like ordinary programs, except:
  - Allow for random choice on how to continue the execution
  - Random choice is done with some specified probability $p$

- **Analysis problems** we consider:
  - Determine the value of a variable after program execution
  - Decide whether the program terminates
Motivation

- Probabilistic Programs are like ordinary programs, except:
  - Allow for random choice on how to continue the execution
  - Random choice is done with some specified probability $p$

- **Analysis problems** we consider:
  - Determine the value of a variable after program execution
    - Determine **expected** values
  - Decide whether the program terminates
Motivation

- Probabilistic Programs are like ordinary programs, except:
  - Allow for random choice on how to continue the execution
  - Random choice is done with some specified probability $p$

- **Analysis problems** we consider:
  - Determine the value of a variable after program execution
  - Determine expected values (expected outcomes)
  - Decide whether the program terminates
Motivation

- Probabilistic Programs are like ordinary programs, except:
  - Allow for random choice on how to continue the execution
  - Random choice is done with some specified probability $p$
- **Analysis problems** we consider:
  - Determine the value of a variable after program execution
    - Determine expected values (expected outcomes)
  - Decide whether the program terminates
    - Decide almost–sure termination
Motivation

- Probabilistic Programs are like ordinary programs, except:
  - Allow for random choice on how to continue the execution
  - Random choice is done with some specified probability $p$

- **Analysis problems** we consider:
  - Determine the value of a variable after program execution
    - Determine expected values (expected outcomes)
  - Decide whether the program terminates (in an expected finite number of steps)
    - Decide almost–sure termination
Motivation

- Probabilistic Programs are like ordinary programs, except:
  - Allow for random choice on how to continue the execution
  - Random choice is done with some specified probability $p$

- **Analysis problems** we consider:
  - Determine the value of a variable after program execution
    - Determine expected values (expected outcomes)
  - Decide whether the program terminates (in an expected finite number of steps)
    - Decide (positive) almost–sure termination
Motivation

- Probabilistic Programs are like ordinary programs, except:
  - Allow for random choice on how to continue the execution
  - Random choice is done with some specified probability $p$

- **Analysis problems** we consider:
  - Determine the value of a variable after program execution
    - Determine expected values (expected outcomes)
  - Decide whether the program terminates (in an expected finite number of steps) [on all inputs]
    - Decide (positive) almost–sure termination
Motivation

- Probabilistic Programs are like ordinary programs, except:
  - Allow for random choice on how to continue the execution
  - Random choice is done with some specified probability $p$

- **Analysis problems** we consider:
  - Determine the value of a variable after program execution
    - Determine expected values (expected outcomes)
  - Decide whether the program terminates (in an expected finite number of steps) [on all inputs]
    - Decide [universal] (positive) almost–sure termination
Motivation

- Probabilistic Programs are like ordinary programs, except:
  - Allow for random choice on how to continue the execution
  - Random choice is done with some specified probability $p$

- **Analysis problems** we consider:
  - Determine the value of a variable after program execution
    - Determine expected values (expected outcomes)
  - Decide whether the program terminates (in an expected finite number of steps) [on all inputs]
    - Decide [universal] (positive) almost–sure termination

How hard is it to solve these analysis problems?
Dissent in the Literature

[Morgan 1996]

“[...] probabilistic reasoning for partial correctness [...] is not much more complex than standard reasoning.”
Dissent in the Literature

[Morgan 1996]

“[...] probabilistic reasoning for partial correctness [...] is not much more complex than standard reasoning.”

[Esparza et al. 2012]

“[Ordinary] termination is a purely topological property [...], but almost–sure termination is not. [...] proving almost–sure termination requires arithmetic reasoning not offered by termination provers.”
The Arithmetical Hierarchy

Definition

Class $\Sigma_0^n$ is defined as $\Sigma_0^n = \{ A \mid A = \{ \vec{x} \mid \exists y_1 \forall y_2 \exists y_3 \cdots \exists y_n : (\vec{x}, y_1, y_2, y_3, \ldots, y_n) \in R \}$, $R$ is a decidable relation.

Class $\Pi_0^n$ is defined as $\Pi_0^n = \{ A \mid A = \{ \vec{x} \mid \forall y_1 \exists y_2 \forall y_3 \cdots \exists y_n : (\vec{x}, y_1, y_2, y_3, \ldots, y_n) \in R \}$, $R$ is a decidable relation.

Class $\Delta_0^n$ is defined as $\Delta_0^n = \Sigma_0^n \cap \Pi_0^n$. 

Benjamin Kaminski
Analyzing Probabilistic Programs is Hard 27.2.2015
The Arithmetical Hierarchy

- Class $\Sigma^0_n$ is defined as

$$\Sigma^0_n = \left\{ A \mid A = \left\{ \vec{x} \mid \exists y_1 \forall y_2 \exists y_3 \cdots \exists / \forall y_n : (\vec{x}, y_1, y_2, y_3, \ldots, y_n) \in R \right\}, R \text{ is a decidable relation} \right\}$$
The Arithmetical Hierarchy

- Class \( \Sigma^0_n \) is defined as

\[
\Sigma^0_n = \left\{ A \mid A = \{ \overline{x} \mid \exists y_1 \forall y_2 \exists y_3 \cdots \exists /\forall y_n : (\overline{x}, y_1, y_2, y_3, \ldots, y_n) \in R \} \right\},
\]

where \( R \) is a decidable relation.

- Class \( \Pi^0_n \) is defined as

\[
\Pi^0_n = \left\{ A \mid A = \{ \overline{x} \mid \forall y_1 \exists y_2 \forall y_3 \cdots \exists /\forall y_n : (\overline{x}, y_1, y_2, y_3, \ldots, y_n) \in R \} \right\},
\]

where \( R \) is a decidable relation.
The Arithmetical Hierarchy

- Class $\Sigma^0_n$ is defined as
  \[ \Sigma^0_n = \left\{ A \mid A = \{ \vec{x} \mid \exists y_1 \forall y_2 \exists y_3 \cdots \exists /\forall y_n : (\vec{x}, y_1, y_2, y_3, \ldots, y_n) \in R \} \right\}, \]
  $R$ is a decidable relation

- Class $\Pi^0_n$ is defined as
  \[ \Pi^0_n = \left\{ A \mid A = \{ \vec{x} \mid \forall y_1 \exists y_2 \forall y_3 \cdots \exists /\forall y_n : (\vec{x}, y_1, y_2, y_3, \ldots, y_n) \in R \} \right\}, \]
  $R$ is a decidable relation

- Class $\Delta^0_n$ is defined as $\Delta^0_n = \Sigma^0_n \cap \Pi^0_n$
The Arithmetical Hierarchy — The Bigger Picture

The following inclusion diagram holds (all inclusions are strict):

\[
\begin{align*}
\Sigma^0_3 & \subseteq \Delta^0_3 & \subseteq \Pi^0_3 \\
\Sigma^0_2 & \subseteq \Delta^0_2 & \subseteq \Pi^0_2 \\
\Sigma^0_1 & \subseteq \Delta^0_1 & \subseteq \Pi^0_1 \\
\vdots & & \vdots \\
\end{align*}
\]
The Arithmetical Hierarchy — The Bigger Picture

The following inclusion diagram holds (all inclusions are strict):

\[
\begin{align*}
\Sigma^0_0 & \subset \Pi^0_0 & \Delta^0_0 \\
\Sigma^0_1 & \subset \Pi^0_1 & \Delta^0_1 \\
\Sigma^0_2 & \subset \Pi^0_2 & \Delta^0_2 \\
\Sigma^0_3 & \subset \Pi^0_3 & \Delta^0_3 \\
\vdots & & \vdots \\
\end{align*}
\]

decidable problems
The Arithmetical Hierarchy — The Bigger Picture

The following inclusion diagram holds (all inclusions are strict):

\[ \ldots \]

\[ \Sigma^0_3 \quad \Delta^0_3 \quad \Pi^0_3 \]

\[ \ldots \]

\[ \Sigma^0_2 \quad \Delta^0_2 \quad \Pi^0_2 \]

\[ \ldots \]

\[ \Sigma^0_1 \quad \Delta^0_1 \quad \Pi^0_1 \]

\[ \ldots \]

decidable problems
The Arithmetical Hierarchy — The Bigger Picture

The following inclusion diagram holds (all inclusions are strict):

\[ \begin{align*}
\Sigma^0_1 & \subseteq \Pi^0_1 \\
\Sigma^0_2 & \subseteq \Pi^0_2 \\
\Sigma^0_3 & \subseteq \Pi^0_3 \\
\ldots & \\
\Sigma & \subseteq \Pi \\
\text{decidable problems} & \\
\end{align*} \]
Some Notation

The expected outcome of variable \( v \) after executing \( P \):
\[ E_{P}(v) \]

The probability that \( P \) terminates on input \( \eta \):
\[ \Pr_{P,\eta}(\downarrow) \]

The expected number of steps until \( P \) terminates on input \( \eta \):
\[ E_{P,\eta}(\downarrow) \]
Some Notation

- The expected outcome of variable $v$ after executing $P$:

$$E_P(v)$$
Some Notation

- The expected outcome of variable $v$ after executing $P$:
  \[ E_P(v) \]

- The probability that $P$ terminates on input $\eta$:
  \[ \Pr_{P,\eta}(\downarrow) \]
Some Notation

- The expected outcome of variable $v$ after executing $P$:
  \[ E_P(v) \]

- The probability that $P$ terminates on input $\eta$:
  \[ \Pr_{P,\eta}(\downarrow) \]

- The expected number of steps until $P$ terminates on input $\eta$:
  \[ E_{P,\eta}(\downarrow) \]
Decision Problems We Analyzed
### Decision Problems We Analyzed

<table>
<thead>
<tr>
<th>Lower and Upper Bounds, and Exact Expected Outcomes</th>
</tr>
</thead>
<tbody>
<tr>
<td>[(P, v, q) \in LEXP :\iff q &lt; E_P(v)]</td>
</tr>
<tr>
<td>[(P, v, q) \in UEXP :\iff q &gt; E_P(v)]</td>
</tr>
<tr>
<td>[(P, v, q) \in EXP :\iff q = E_P(v)]</td>
</tr>
</tbody>
</table>
Decision Problems We Analyzed

Lower and Upper Bounds, and Exact Expected Outcomes

\[(P, v, q) \in \mathcal{L}\mathcal{E}\mathcal{X}\mathcal{P} :\iff q < \mathbb{E}_P(v)\]

\[(P, v, q) \in \mathcal{U}\mathcal{E}\mathcal{X}\mathcal{P} :\iff q > \mathbb{E}_P(v)\]

\[(P, v, q) \in \mathcal{E}\mathcal{X}\mathcal{P} :\iff q = \mathbb{E}_P(v)\]

Almost–Sure Termination \(\mathcal{A}\mathcal{S}\mathcal{T}\)

\[(P, \eta) \in \mathcal{A}\mathcal{S}\mathcal{T} :\iff \Pr_{P,\eta}(\downarrow) = 1\]
Variations of $\textit{AST}$
Variations of \( \mathcal{AST} \)

Positive Almost–Sure Termination \( \mathcal{PAST} \)

\[(P, \eta) \in \mathcal{PAST} \iff E_{P,\eta}(\downarrow) < \infty\]
Variations of $\mathcal{AST}$

Positive Almost–Sure Termination $\mathcal{PAST}$

$$(P, \eta) \in \mathcal{PAST} \iff \mathbb{E}_{P,\eta}(\downarrow) < \infty$$

Notice $\mathcal{PAST} \subsetneq \mathcal{AST}$. 
Variations of $AST$

**Positive Almost–Sure Termination $\mathcal{PAST}$**

$$(P, \eta) \in \mathcal{PAST} \iff E_{P,\eta}(\downarrow) < \infty$$

Notice $\mathcal{PAST} \subsetneq AST$.

**Universal Versions of $AST$ and $\mathcal{PAST}$**

$$P \in UAST \iff \forall \eta: (P, \eta) \in AST$$

$$P \in U\mathcal{PAST} \iff \forall \eta: (P, \eta) \in \mathcal{PAST}$$
A (very) Simple Example Program

Consider the program $P_{geo}$:

\[
x := 0;
\{continue := 0\} [0.5] \{continue := 1\};
while (continue \neq 0)\{
    x := x + 1;
    \{continue := 0\} [0.5] \{continue := 1\}
\}\
\]
A (very) Simple Example Program

Consider the program \( P_{geo} \):

\[
\begin{align*}
  x &:= 0; \\
  \{ \text{continue} := 0 \} & [0.5] \{ \text{continue} := 1 \}; \\
  \text{while} (\text{continue} \neq 0) \{ \\
    x &:= x + 1; \\
    \{ \text{continue} := 0 \} & [0.5] \{ \text{continue} := 1 \} \\
  \}
\end{align*}
\]

\[ \mathbb{E}_{P_{geo}}(x) = 2 \]
A (very) Simple Example Program

Consider the program \( P_{geo} \):

\[
x := 0; \\
\{\text{continue} := 0\} \text{[0.5]} \{\text{continue} := 1\}; \\
\text{while (continue \neq 0)}\{
  x := x + 1;
  \{\text{continue} := 0\} \text{[0.5]} \{\text{continue} := 1\}
\}
\]

\( E_{P_{geo}}(x) = 2 \)

\( E_{P_{geo}}(\text{continue}) = 0 \)
A (very) Simple Example Program

Consider the program $P_{geo}$:

\[
\begin{align*}
x &:= 0; \\
\{ \text{continue} := 0 \} &\ [0.5] \ \{ \text{continue} := 1 \}; \\
\text{while} \ (\text{continue} \neq 0) \{} \\
\quad &x := x + 1; \\
\quad &\{ \text{continue} := 0 \} \ [0.5] \ {\text{continue} := 1} \\
\} \\
\end{align*}
\]

- $E_{P_{geo}}(x) = 2$
- $E_{P_{geo}}(\text{continue}) = 0$
- $P_{geo}$ terminates almost–surely on all inputs
A (very) Simple Example Program

Consider the program $P_{geo}$:

\[
x := 0;
\{\text{continue := 0} \ [0.5] \ \{\text{continue := 1}\};
\text{while (continue \neq 0)}\{
    x := x + 1;
    \{\text{continue := 0} \ [0.5] \ \{\text{continue := 1}\}
\}
\]
\]

- $E_{P_{geo}}(x) = 2$
- $E_{P_{geo}}(\text{continue}) = 0$
- $P_{geo}$ terminates almost–surely on all inputs
- Expected runtime of $P_{geo}$ is $\mathcal{O}(E_{P_{geo}}(x))$ on all inputs
Summary of Results

...
Summary of Results

\[
\begin{align*}
\Sigma_3^0 & \quad \Delta_3^0 & \quad \Pi_3^0 \\
\Sigma_2^0 & \quad \Delta_2^0 & \quad \Pi_2^0 \\
\mathcal{H} & \quad \Delta_1^0 & \quad \Pi_1^0 \\
\overline{\mathcal{H}} & \quad \mathcal{LExp} & \quad \overline{\mathcal{H}}
\end{align*}
\]
Summary of Results

...:

\[ \sum_3^0 \quad \Delta_3^0 \quad \Pi_3^0 \]

\[ \sum_2^0 \quad \Delta_2^0 \quad \Pi_2^0 \]

\[ \sum_1^0 \quad \mathcal{L}_{EXP} \quad \Pi_1^0 \]

\[ \sum_0^0 \quad \Delta_0^0 \quad \Pi_0^0 \]

\[ \sum \quad \Pi \quad \Delta \]

...:

semi-decidable

Thank you for your kind attention :-)

Benjamin Kaminski

Analyzing Probabilistic Programs is Hard 27.2.2015
Summary of Results

\[ \Sigma_3^0 \quad \Delta_3^0 \quad \Pi_3^0 \]

\[ \Sigma_2^0 \quad \mathcal{UEXP} \quad \Pi_2^0 \]

\[ \mathcal{UH} \quad \Delta_2^0 \quad \mathcal{UH} \]

\[ \Sigma_1^0 \quad \mathcal{LEXP} \quad \Pi_1^0 \]

\[ \mathcal{H} \quad \Delta_1^0 \quad \overline{\mathcal{H}} \]

semi-decidable

Thank you for your kind attention :-)

Benjamin Kaminski

Analyzing Probabilistic Programs is Hard 27.2.2015
Summary of Results

- $\Sigma_0$ semi-decidable
- $\Pi_0$ semi-decidable
- $\Delta_0$ semi-decidable

- $\Sigma_2$ $\mathcal{UEXP}$
- $\Pi_2$ $\mathcal{UEXP}$
- $\Delta_2$ $\mathcal{UEXP}$

- $\Sigma_3$ $\mathcal{EXP}$
- $\Pi_3$ $\mathcal{EXP}$
- $\Delta_3$ $\mathcal{EXP}$

with access to $\mathcal{H}$-oracle:
- semi-decidable
Summary of Results

with access to $\mathcal{H}$–oracle: semi–decidable

semi–decidable

Thank you for your kind attention :-)

Benjamin Kaminski
Summary of Results

with access to $\mathcal{H}$–oracle: semi–decidable

semi–decidable
Summary of Results

with access to $\mathcal{H}$–oracle: semi–decidable

semi–decidable

Thank you for your kind attention :-)

Benjamin Kaminski
Summary of Results

with access to \( H \)-oracle: semi-decidable

semi-decidable

not semi-decidable; even with access to \( H \)-oracle

Thank you for your kind attention :-)

Benjamin Kaminski

Analyzing Probabilistic Programs is Hard 27.2.2015
with access to $\mathcal{H}$–oracle: semi–decidable

semi–decidable

not semi–decidable; even with access to $\mathcal{H}$–oracle
Summary of Results

with access to $\mathcal{H}$–oracle: semi–decidable

semi–decidable

not semi–decidable; even with access to $\mathcal{H}$–oracle

Thank you for your kind attention :-)

Benjamin Kaminski
Summary of Results

with access to $\mathcal{H}$–oracle: semi–decidable

semi–decidable

not semi–decidable; even with access to $\mathcal{H}$–oracle

not semi–decidable; even with access to $\mathcal{U}\mathcal{H}$–oracle

Thank you for your kind attention :-)

Benjamin Kaminski
Summary of Results

Thank you for your kind attention :-)

with access to $\mathcal{H}$–oracle: semi-decidable

semi-decidable

not semi-decidable; even with access to $\mathcal{UH}$–oracle

not semi-decidable; even with access to $\mathcal{H}$–oracle