PMAF: AN ALGEBRAIC FRAMEWORK FOR STATIC ANALYSIS OF PROBABILISTIC PROGRAMS

Di Wang¹, Jan Hoffmann¹, Thomas Reps²

¹ Carnegie Mellon University
² University of Wisconsin; GrammaTech, Inc.
PROBABILISTIC PROGRAMS

Draw random **data** from distributions

Condition **control-flow** at random
PROBABILISTIC PROGRAMS

- True randomness
- Distributions on executions

```plaintext
b1 ~ Bernoulli(0.5);
b2 ~ Bernoulli(0.7);
while (b1 && b2) do
    if prob(0.6) then
        b1 ~ Bernoulli(0.5)
    else
        b2 ~ Bernoulli(0.7)
    fi;
tick(1.0);
od;
return (b1, b2)
```
BAYESIAN NETWORKS

- Conditional distributions
- Query about the posterior
BAYESIAN NETWORKS

- Conditional distributions
- Query about the posterior

\[ \text{Prob}[\text{Cancer} \mid \text{Smoker} \land \text{Xray Res}] = ? \]
BAYESIAN NETWORKS AS PROB. PROG.
if $\text{prob}(0.6)$ then
    $b_1 \sim \text{Bernoulli}(0.5)$
else
    $b_2 \sim \text{Bernoulli}(0.7)$
fi
b1 ~ Bernoulli(0.5);
b2 ~ Bernoulli(0.7);
while (b1 && b2) do
  if prob(0.6) then
    b1 ~ Bernoulli(0.5)
  else
    b2 ~ Bernoulli(0.7)
  fi;
tick(1.0)
od;
return (b1, b2)
Bayesian Networks as Prob. Prog.

\begin{align*}
b_1 &\sim \text{Bernoulli}(0.5); \\
b_2 &\sim \text{Bernoulli}(0.7); \\
\text{while} &\ (b_1 && b_2) \ \text{do} \\
&\quad \text{if} \ \text{prob}(0.6) \ \text{then} \\
&\qquad b_1 &\sim \text{Bernoulli}(0.5) \\
&\quad \text{else} \\
&\qquad b_2 &\sim \text{Bernoulli}(0.7) \\
&\text{fi;} \\
&\text{tick}(1.0) \\
&\text{od;} \\
&\text{return} \ (b_1, b_2)
\end{align*}

\textbf{Query:} probability that $b_1$ and $b_2$ are both false?
if prob(0.6) then
  b1 ~ Bernoulli(0.5)
else
  b2 ~ Bernoulli(0.7)
fi

while (b1 && b2) do
  if prob(0.6) then
    b1 ~ Bernoulli(0.5)
  else
    b2 ~ Bernoulli(0.7)
  fi;
  tick(1.0)
od;

return (b1, b2)

Query: expected termination time?
SAMPLING-BASED TECHNIQUES

- Simulation & frequency count
- Flexible & universal
- Potentially unsound & inefficient
SAMPLING-BASED TECHNIQUES

- Simulation & frequency count
- Flexible & universal
- Potentially unsound & inefficient

What about static analysis?
ABSTRACT INTERPRETATION

- Cousot et al. proposed **Probabilistic Abstract Interpretation**

- **Sound**, flexible, and universal

---

1 P. Cousot and M. Monerau. Probabilistic Abstract Interpretation. In ESOP’12.
Cousot et al. proposed **Probabilistic Abstract Interpretation**\(^1\)

- **Sound**, flexible, and universal
- Their concrete semantics resolves probabilities *prior to* nondeterminism

---

\(^1\) P. Cousot and M. Monerau. Probabilistic Abstract Interpretation. In *ESOP’12*. 
Cousot et al. proposed **Probabilistic Abstract Interpretation**.

- **Sound**, flexible, and universal
- Their concrete semantics resolves probabilities prior to nondeterminism
- Sometimes desirable to revolve nondeterminism prior to probabilities

---

1 P. Cousot and M. Monerau. Probabilistic Abstract Interpretation. In ESOP’12.
COUSOT ET AL.’S SEMANTICS

* denotes nondeterministic choice

tick(q) increases $T$ by q

if * then
    if prob(0.5) then tick(1.0) else tick(2.0) fi
else
    if prob(0.5) then tick(1.0) else tick(2.0) fi
fi
COUSOT ET AL.'S SEMANTICS

* denotes nondeterministic choice

\[
\text{if } \ast \text{ then }
\begin{align*}
\text{if } \text{prob}(0.5) \text{ then } & \text{tick}(1.0) \text{ else } \text{tick}(2.0) \text{ fi} \\
\text{else} & 
\begin{align*}
\text{if } \text{prob}(0.5) \text{ then } & \text{tick}(1.0) \text{ else } \text{tick}(2.0) \text{ fi} \\
\end{align*}
\end{align*}
\]

\text{with prob. } \frac{1}{4}

tick(q) \text{ increases } T \text{ by } q
COUSOT ET AL.'S SEMANTICS

* denotes nondeterministic choice

tick(q) increases \( T \) by \( q \)

if * then
  if \( \text{prob}(0.5) \) then tick(1.0) else tick(2.0) fi
else
  if \( \text{prob}(0.5) \) then tick(1.0) else tick(2.0) fi
fi
COUSOT ET AL.'S SEMANTICS

* denotes nondeterministic choice

tick(q) increases $T$ by $q$

if * then
  if $prob(0.5)$ then tick(1.0) else tick(2.0) fi
else
  if $prob(0.5)$ then tick(1.0) else tick(2.0) fi
fi

with prob. $\frac{1}{4}$
COUSOT ET AL.'S SEMANTICS

* denotes nondeterministic choice

tick(q) increases $T$ by $q$

if * then
  if prob(0.5) then tick(1.0) else tick(2.0) fi
else
  if prob(0.5) then tick(1.0) else tick(2.0) fi
fi
Cousot et al.'s semantics

* denotes nondeterministic choice

\begin{align*}
\text{if } * \text{ then} \\
& \quad \text{if } \text{prob}(0.5) \text{ then } \text{tick}(1.0) \text{ else } \text{tick}(2.0) \\text{fi} \\
\text{else} \\
& \quad \text{if } \text{prob}(0.5) \text{ then } \text{tick}(1.0) \text{ else } \text{tick}(2.0) \\text{fi} \\
\text{fi}
\end{align*}

tick(q) increases $T$ by $q$

with prob. $\frac{1}{4}$
COUSOT ET AL.’S SEMANTICS

* denotes nondeterministic choice

```
if * then
  if prob(0.5) then tick(1.0) else tick(2.0) fi
else
  if prob(0.5) then tick(1.0) else tick(2.0) fi
fi
```
tick(q) increases $T$ by $q$
COUSOT ET AL.'S SEMANTICS

* denotes nondeterministic choice

tick(q) increases $T$ by q

if * then
  if prob(0.5) then tick(1.0) else tick(2.0) fi
else
  if prob(0.5) then tick(1.0) else tick(2.0) fi
fi

with prob. $\frac{1}{4}$
COUSOT ET AL.’S SEMANTICS

\[ \text{\texttt{if } \ast \text{ then}} \]
\[ \quad \text{if } \text{prob}(0.5) \text{ then } \text{tick}(1.0) \text{ else } \text{tick}(2.0) \text{ fi} \]
\[ \text{else} \]
\[ \quad \text{if } \text{prob}(0.5) \text{ then } \text{tick}(1.0) \text{ else } \text{tick}(2.0) \text{ fi} \]
\[ \text{fi} \]

\( \ast \) denotes nondeterministic choice

\[ \text{tick}(q) \text{ increases } T \text{ by } q \]
COUSOT ET AL.’S SEMANTICS

* denotes nondeterministic choice
tick(q) increases $T$ by q

if * then
  if prob(0.5) then tick(1.0) else tick(2.0) fi
else
  if prob(0.5) then tick(1.0) else tick(2.0) fi
fi

Their concrete semantics yields

$$\mathbb{E}[T] \in \frac{1}{4} \cdot \{1\} + \frac{1}{4} \cdot \{2\} + \frac{1}{4} \cdot \{1,2\} + \frac{1}{4} \cdot \{1,2\} = \{1.25, 1.5, 1.75\}$$
Their concrete semantics yields

\[ \mathbb{E}[T] \in \frac{1}{4} \cdot \{1\} + \frac{1}{4} \cdot \{2\} + \frac{1}{4} \cdot \{1,2\} + \frac{1}{4} \cdot \{1,2\} = \{1.25,1.5,1.75\} \]
Their concrete semantics yields

\[ \mathbb{E}[T] = \frac{1}{4} \cdot \{1\} + \frac{1}{4} \cdot \{2\} + \frac{1}{4} \cdot \{1,2\} + \frac{1}{4} \cdot \{1,2\} = \{1.25, 1.5, 1.75\} \]

while our semantics yields \( \mathbb{E}[T] = 1.5 \)
CONTRIBUTIONS

- A denotational semantics with nondeterminism resolved first
- An algebraic framework for interprocedural dataflow analysis of first-order probabilistic programs
CONTRIBUTIONS

• A denotational semantics with nondeterminism resolved first
• An algebraic framework for interprocedural dataflow analysis of first-order probabilistic programs

PMAF
CONTRIBUTIONS

- A denotational semantics with nondeterminism resolved first
- An algebraic framework for interprocedural dataflow analysis of first-order probabilistic programs
CONTRIBUTIONS

- A denotational semantics with nondeterminism resolved first
- An algebraic framework for interprocedural dataflow analysis of first-order probabilistic programs

PMAF
CONTRIBUTIONS

- A denotational semantics with nondeterminism resolved first
- An **algebraic framework** for interprocedural dataflow analysis of **first-order probabilistic programs**

- Bayesian Inference
- Markov Decision Problem

Existing → PMAF
CONTRIBUTIONS

- A denotational semantics with nondeterminism resolved first
- An algebraic framework for interprocedural dataflow analysis of first-order probabilistic programs

- Bayesian Inference
- Markov Decision Problem

- Expectation-Invariant Analysis
EXAMPLE ANALYSES

- Our framework can be instantiated to prove:
- the probability that $b_1$ and $b_2$ are both false at the end of the program = 0.15
- the expected termination time (ticks) = $\frac{5}{6}$

```plaintext
b1 ~ Bernoulli(0.5);
b2 ~ Bernoulli(0.7);
while (b1 && b2) do
  if prob(0.6) then
    b1 ~ Bernoulli(0.5)
  else
    b2 ~ Bernoulli(0.7)
  fi;
  tick(1.0)
od;
return (b1, b2)
```
OVERVIEW

- Motivation

- The Algebraic Framework

- Hyper-Graph Analysis

- Evaluation
Any static analysis method performs reasoning in some space of **program properties** and **property operations**.

**Actions**

- `skip`
- `x := x + 5`
- `b ~ Bernoulli(0.4)`
- `tick(1.0)`
- ...

**Semantic Function**

**Program Properties**

**Sequencing**
- Cond.-choice
- Prob.-choice
- Nondet.-choice
THE ALGEBRAIC FRAMEWORK

Actions

- skip
- $x := x + 5$
- $b \sim \text{Bernoulli}(0.4)$
- tick(1.0)
- ...

Concrete Operations

Concrete Semantics

Concrete Semantic Function

Sound Abstraction

Abstract Semantic Function

Abstract Semantics

Concrete Semantics

Abstract Operations
THE ALGEBRAIC FRAMEWORK

- Characterize program properties and property operations by algebraic laws
THE ALGEBRAIC FRAMEWORK

- Characterize program properties and property operations by **algebraic laws**

\[ \langle M, \sqsubseteq, \otimes, \varphi, \oplus, \psi, \bot, 1 \rangle \]
THE ALGEBRAIC FRAMEWORK

- Characterize program properties and property operations by algebraic laws

\[ \langle M, \sqsubseteq, \otimes, \varphi, \oplus, \boxdot, \bot, 1 \rangle \]

Program properties and approximation order
THE ALGEBRAIC FRAMEWORK

- Characterize program properties and property operations by **algebraic laws**

\[ \langle M, \sqsubseteq, \otimes, \phi, p, \oplus, \psi, \bot, 1 \rangle \]

Program properties and approximation order

Sequencing, cond.-choice, prob.-choice, and nondet.-choice
THE ALGEBRAIC FRAMEWORK

- Characterize program properties and property operations by *algebraic laws*

\[ \langle M, \sqsubseteq, \otimes, \varphi, p \oplus, \psi, \perp, 1 \rangle \]

- Program properties and approximation order
- Sequencing, cond.-choice, prob.-choice, and nondet.-choice
- The bottom element and the identity element
THE ALGEBRAIC FRAMEWORK

- Characterize program properties and property operations by algebraic laws

\[ \langle M, \sqsubseteq, \otimes, \varphi \Diamond, \oplus, \triangledown, \bot, 1 \rangle \]

Program properties and approximation order

Sequencing, cond.-choice, prob.-choice, and nondet.-choice
THE ALGEBRAIC FRAMEWORK

- Characterize program properties and property operations by algebraic laws

\[ \langle M, \sqsubseteq, \otimes, \phi, p \oplus, \uplus, \perp, 1 \rangle \]

- Program properties and approximation order

- Sequencing, cond.-choice, prob.-choice, and nondet.-choice

- Laws:
  - \[ a \oplus b = b \, 1 - p \oplus a \]
  - \[ (a \otimes b) \otimes c = a \otimes (b \otimes c) \]
  - \[ a \otimes 1 = 1 \otimes a = a \]
  - \[ a \uplus a = a \]
  - ...
OVERVIEW

- Motivation
- The Algebraic Framework
- Hyper-Graph Analysis
- Evaluation
PROGRAM SEMANTICS

- Control-flow graphs
- Reason about paths
- Paths are independent

Code examples:
- \( i := i + 1 \)
- \( n := n / 2 \)
- \( n := 3 \times n + 1 \)
PROGRAM SEMANTICS

- Reason about *distributions over paths*

- Paths are *not independent*
PROGRAM SEMANTICS

- Reason about distributions over paths
- Paths are not independent

```
n := n/2
```

Diagram:
```
[ n != 1 ]

[ n % 2 == 0 ]

n := n/2
```
PROGRAM SEMANTICS

- Reason about distributions over paths
- Paths are *not* independent

```
\[ n \neq 1 \]
\[ n \% 2 = 0 \]
\[ n := n/2 \]
```

```
\[ n \neq 1 \]
\[ n \% 2 = 0 \]
\[ n := n/2 \]
```

```
\[ n := n/2 \]
prob(0.6)
```

```
\[ n := n/2 \]
prob(0.6)
\[ n := n+1 \]
\[ n := n/2 \]
```
PROGRAM SEMANTICS

- Reason about distributions over paths
- Paths are not independent

\[ n \neq 1 \]
\[ n \% 2 = 0 \]
\[ n := n / 2 \]

\[ n \neq 1 \]
\[ n \% 2 = 0 \]
\[ n := n / 2 \]
\[ n := n + 1 \]
\[ n := n / 2 \]

\[ \text{prob}(0.6) \]

\[ \text{prob}(0.6) \]

\[ n \] may be a random value
PROGRAM SEMANTICS

- Reason about distributions over paths
- Paths are not independent

Code:

```
if n != 1 and n % 2 == 0:
    n := n / 2
```

```python
n = n // 2
```

n may be a random value

Random control-flow
PROGRAM SEMANTICS

- Reason about **distributions over paths**
- Paths are **not independent**
- **Nondeterminism** is modeled by **collections** of such distributions

![Diagram of program semantics]

- `n:=n/2`
- `n:=n+1`
- `prob(0.6)`
- `prob(0.6)`
- `n!:=1`
- `n may be a random value`
- `random control-flow`
Reason about distributions over paths

Paths are not independent

Nondeterminism is modeled by collections of such distributions

Resolve nondeterminism first!
PROGRAM SEMANTICS

\[ \text{prob}(0.6) \rightarrow \text{true} \]

\[ \text{false} \]
PROGRAM SEMANTICS

- Control-flow hyper-graphs
- Branching are hyper-edges

\[
\begin{align*}
&b_1 \sim \text{Bernoulli}(0.5); \\
&b_2 \sim \text{Bernoulli}(0.7); \\
&\text{while } (b_1 \text{ && } b_2) \text{ do} \\
&\quad \text{if } \text{prob}(0.6) \text{ then} \\
&\quad\quad b_1 \sim \text{Bernoulli}(0.5) \\
&\quad\quad \text{else} \\
&\quad\quad b_2 \sim \text{Bernoulli}(0.7) \\
&\quad \text{fi;} \\
&\quad \text{tick}(1.0) \\
&\text{od;} \\
&\text{return } (b_1, b_2)
\end{align*}
\]
HYPER-GRAHP ANALYSIS

- **Forward assertions**

  - The semantics of a node is a summary of computation that continues from the node.

  ![Graph Diagram]

  - \( b_1, b_2 \sim B(0.5), B(0.7) \)
  - \([b_1 \& \& b_2]\)
  - \( \text{false} \)
  - \( \text{true} \)
  - \( \text{ret} \)
  - \( \text{true} \)
  - \( \text{prob}(0.6) \)
  - \( \text{false} \)
  - \( b_1 \sim B(0.5) \)
  - \( b_2 \sim B(0.7) \)
  - \( \text{tick}(1.0) \)
**HYPER-GRAPH ANALYSIS**

- **Forward assertions**
- The semantics of a node is a summary of computation that continues from the node

E.g. the semantics of the node is

\[
\lambda(b_1, b_2). \text{if } b_2 \text{ then } \frac{1}{7}[b_1' = T, b_2' = F] + \frac{6}{7}[b_1' = F, b_2' = T]
\]

\[
\text{else } \frac{1}{2}[b_1' = T, b_2' = F] + \frac{1}{2}[b_1' = F, b_2' = F]
\]
The hyper-graph analysis is formulated by an equation system
The hyper-graph analysis is formulated by an equation system

\[ S[0] = \text{seq}[b1, b2 \sim B(0.5), B(0.7)](S[1]) \]
The hyper-graph analysis is formulated by an equation system

\[ S[0] = \text{seq}[b1, b2 \sim B(0.5), B(0.7)](S[1]) \]
\[ S[1] = \text{cond}[b1 \& b2](S[2], S[3]) \]
The hyper-graph analysis is formulated by an equation system

\[ S[0] = \text{seq}[b1, b2 \sim \text{B}(0.5), \text{B}(0.7)](S[1]) \]
\[ S[1] = \text{cond}[b1 \& \& b2](S[2], S[3]) \]
\[ S[2] = \text{prob}[0.6](S[4], S[5]) \]
The hyper-graph analysis is formulated by an equation system

\[
S[0] = \text{seq}[b1, b2 \sim \text{B}(0.5), \text{B}(0.7)](S[1])
\]

\[
S[1] = \text{cond}[b1 \& \& b2](S[2], S[3])
\]

\[
S[2] = \text{prob}[0.6](S[4], S[5])
\]
The hyper-graph analysis is formulated by an equation system

\[
\begin{align*}
S[0] &= \text{seq}[b_1, b_2 \sim B(0.5), B(0.7)](S[1]) \\
S[1] &= \text{cond}[b_1 \& \& b_2](S[2], S[3]) \\
S[2] &= \text{prob}[0.6](S[4], S[5])
\end{align*}
\]

Use the semantic algebra to interpret \textit{seq, cond, prob}.
The hyper-graph analysis is formulated by an equation system

\[
S[0] = [b1, b2 \sim B(0.5), B(0.7)] \otimes S[1]
\]

\[
S[1] = S[2]_{b1 \& \& b2} \otimes S[3]
\]

\[
S[2] = S[4]_{0.6} \oplus S[5]
\]

Use the semantic algebra to interpret \texttt{seq, cond, prob}
HYPER-GRAH ANALYSIS

The hyper-graph analysis is formulated by an equation system

\[ S[0] = [b1, b2 \sim B(0.5), B(0.7)] \otimes S[1] \]

\[ S[1] = S[2]_{b1 \& b2} \otimes S[3] \]

\[ S[2] = S[4]_{0.6} \oplus S[5] \]

If using **abstract** semantics, we obtain an equation system for **static analysis**
OVERVIEW

- Motivation
- The Algebraic Framework
- Hyper-Graph Analysis
- Evaluation
INSTANTIATIONS

- Bayesian Inference
- Markov Decision Problem

Existing → PMAF
INSTANTIATIONS

Existing

Bayesian Inference
Markov Decision Problem

PMAF

New

Interprocedural analyzers!
INSTANTIATIONS

- Bayesian Inference
- Markov Decision Problem
INSTANTIATIONS

- Bayesian Inference
- Markov Decision Problem

- Expectation-Invariant Analysis

Existing -> PMAF -> New
INSTANTIATIONS

- Bayesian Inference
- Markov Decision Problem

Prove invariants among initial values and expected final values

- Expectation-Invariant Analysis
## PROBABILISTIC MODEL ANALYSES

- Benchmark collected from PReMo\(^1\)
- Achieve the same precision

**Bayesian Inference (Table 2)**

<table>
<thead>
<tr>
<th>Program</th>
<th>#loc</th>
<th>time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>compare</td>
<td>17</td>
<td>2.22</td>
</tr>
<tr>
<td>dice</td>
<td>12</td>
<td>0.02</td>
</tr>
<tr>
<td>eg1</td>
<td>10</td>
<td>0.02</td>
</tr>
<tr>
<td>eg2</td>
<td>16</td>
<td>0.01</td>
</tr>
<tr>
<td>recursive</td>
<td>14</td>
<td>0.01</td>
</tr>
</tbody>
</table>

**Markov Decision Problem (Table 2)**

<table>
<thead>
<tr>
<th>Program</th>
<th>#loc</th>
<th>time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>binary10</td>
<td>184</td>
<td>0.03</td>
</tr>
<tr>
<td>loop</td>
<td>10</td>
<td>0.03</td>
</tr>
<tr>
<td>quicksort7</td>
<td>109</td>
<td>0.03</td>
</tr>
<tr>
<td>recursive</td>
<td>13</td>
<td>0.03</td>
</tr>
<tr>
<td>student</td>
<td>43</td>
<td>0.03</td>
</tr>
</tbody>
</table>

\(^1\) D. Wojtczak and K. Etessami. PReMo - Probabilistic Recursive Models analyzer. Available at [groups.inf.ed.ac.uk/premo/](groups.inf.ed.ac.uk/premo/).
**EXPECTATION-INVARIANT ANALYSIS**

- Benchmark collected from the literature\(^1,2\) and also handcrafted by us
- Derive expectation invariants as least as precise as them in most case

### Expectation-Invariant Analysis (Table 1)

<table>
<thead>
<tr>
<th>Program</th>
<th>#loc</th>
<th>time (sec)</th>
<th>Expectation Invariants</th>
</tr>
</thead>
<tbody>
<tr>
<td>binom-update</td>
<td>14</td>
<td>0.06</td>
<td>$E[4x'-n']=4x-n$, $E[x']\leq x+1/4$</td>
</tr>
<tr>
<td>eg</td>
<td>8</td>
<td>0.89</td>
<td>$E[x'+y']=x+y+4$, $E[z']=1/4z+3/4$</td>
</tr>
<tr>
<td>recursive</td>
<td>13</td>
<td>0.37</td>
<td>$E[x']=x+9$</td>
</tr>
<tr>
<td>mot-ex</td>
<td>16</td>
<td>0.06</td>
<td>$E[2x'-y']=2x-y$, $E[4x'-3c']=4x-3c$, $E[x']\leq x+3/4$</td>
</tr>
</tbody>
</table>

---

SUMMARY

Design → Prove → Implement
SUMMARY

PMAF

Design → Prove → Implement

Hyper-Graph Semantics
SUMMARY

- Hyper-Graph Semantics
- Bayesian Inference
- Markov Decision Problem
- Expectation-Invariant Analysis

Design → Prove → Implement

PMAF

Instantiations
LIMITATIONS:
- Only first-order programs
- No function pointers
- Not Galois connections

SUMMARY

- Bayesian Inference
- Markov Decision Problem
- Expectation-Invariant Analysis

INSTANTIATIONS

- Hyper-Graph Semantics

DESIGN -> PROVE -> IMPLEMENT
Limitations:
- Only first-order programs
- No function pointers
- Not Galois connections

Future work:
- Higher-order programs
- More efficient algorithm
- New instantiations

Design

Prove

Implement

Bayesian Inference
Markov Decision Problem
Expectation-Invariant Analysis

PMAF

Hyper-Graph Semantics

Instantiations