Probabilistic Language Inclusion Problems

Tobias Winkler

IMDEA Software Institute, Madrid — 19.09.2023
Probabilistic Language Inclusion Problems
Tobias Winkler
Language Inclusion
Is $L \subseteq M$ true?
Robot on a Grid
Robot on a Grid

- $\Sigma = \{(x, y) \mid 0 \leq x, y < 5\}$
Robot on a Grid

• $\Sigma = \{(x, y) \mid 0 \leq x, y < 5\}$

• $L = \text{all possible walks of } \overline{\text{robot}} \text{ on the grid}$
Robot on a Grid

- $\Sigma = \{(x, y) \mid 0 \leq x, y < 5\}$
- $L = \text{all possible walks of } \text{Robot on the grid}$
- $M = "\text{before } \text{Diamond}"$
Robot on a Grid

- $\Sigma = \{(x, y) \mid 0 \leq x, y < 5\}$
- $L =$ all possible walks of Robot on the grid
- $M =$ "before" 
- $L \subseteq M$ is false
Robot on a Grid

- $\Sigma = \{(x, y) \mid 0 \leq x, y < 5\}$
- $L = \text{all possible walks of } \text{Robot} \text{ on the grid}$
- $M = "\text{before } \text{Diamond}"$
- $L \subseteq M$ is false
Probabilistic Language Inclusion
How true is $L \subseteq M$?
Probabilistic Language Inclusion

How true is $L \subseteq M$ ?

0.95

0.1
Probabilistic Language Inclusion

Problem Definition

\[ P(L \subseteq M) = 0.95 \]

\[ P(L \subseteq M) = 0.1 \]
Probabilistic Language Inclusion
Problem Definition

- Given: probability measure $L$ on $\Sigma^\omega$, $M \subseteq \Sigma^\omega$ measurable
Probabilistic Language Inclusion

Problem Definition

• Given: probability measure $L$ on $\Sigma^\omega$, $M \subseteq \Sigma^\omega$ measurable
• Question: What is $Pr_{w \sim L}(w \in M) = L(M)$
Probabilistic Language Inclusion
Problem Definition

- Given: probability measure $L$ on $\Sigma^\omega$, $M \subseteq \Sigma^\omega$ measurable
- Question: What is $Pr_{w \sim L}(w \in M) = L(M)$
- Variant: Is $Pr_{w \sim L}(w \in M) = 1$? “almost-sure inclusion”
Robot on a Grid (probabilistic)

- $\Sigma = \{(x, y) | 0 \leq x, y < 5\}$
- $L = \text{random walks of } \square \text{ on the grid}$
- $M = "\text{before} \quad \text{"}$
Robot on a Grid (probabilistic)

- $\Sigma = \{(x, y) \mid 0 \leq x, y < 5\}$
- $L =$ random walks of $\bullet$ on the grid
- $M =$ "\text{ before } $"]$

$Pr_{w \sim L}(w \in M) = \frac{7050}{12113} \approx 0.58$
Robot on a Grid (probabilistic)

- $\Sigma = \{(x, y) \mid 0 \leq x, y < 5\}$
- $L = \text{random walks of } \boxed{\text{Robot}} \text{ on the grid}$
- $M = "\boxed{\text{Green}} \text{ before } \boxed{\text{Diamond}}"$

- $Pr_{w \sim L}(w \in M) = \frac{7050}{12113} \approx 0.58$
How to define $L$?
How to define $L$?
How to define $L$?

- $L = \{ \frac{1}{3} : ba^\omega, \frac{2}{3} : ab^\omega \}$ (discrete, finite support)
How to define $L$?

- $L = \left\{ \frac{1}{3} : ba^\omega, \frac{2}{3} : ab^\omega \right\}$ (discrete, finite support)

- $\mathcal{A}_L =$ “Generative” probabilistic automaton with trivial Büchi acceptance
  = finite Markov chain with labeled transitions
How to define $L$?
How to define $L$?

- $L = \text{uniform probability measure on } \{a, b\}^\omega$ (continuous!)
How to define $L$?

- $L = \text{uniform probability measure on } \{a, b\}^\omega$ (continuous!)
- Each individual word $w \in \{a, b\}^\omega$ occurs with probability 0
Example

\[ \mathcal{A}_L \]

\[ \frac{1}{2}, a \quad \frac{1}{2}, b \]

\[ \mathcal{A}_M \]

\[ b \quad a \quad a, b \]
Example

\[ A_L \]

\[ \frac{1}{2}, a \quad \frac{1}{2}, b \]

\[ A_M \]

\[ b \quad a \quad a, b \]

- \( L = \) uniform probability measure on \( \{a, b\}^\omega \) (continuous)
**Example**

- $L = \text{uniform probability measure on } \{a, b\}^\omega$ (continuous)
- $M = \mathbb{b}^* \mathbb{a}(a + b)^\omega = \text{“eventually a occurs”}$
Example

\( A_L \)

\[ \frac{1}{2}, a \rightarrow \frac{1}{2}, b \]

\( A_M \)

\[ b \rightarrow a \rightarrow a, b \]

- \( L = \) uniform probability measure on \( \{a, b\}^\omega \) (continuous)
- \( M = b^*a(a + b)^\omega = \) “eventually \( a \) occurs”
- \( Pr_{w \sim L}(w \in M) = ? \)
Basic Approach

probabilistic automaton $\mathcal{A}_L$

non-deterministic automaton $\mathcal{A}_M$

determinization

deterministic automaton $\mathcal{A}_M$

product $\mathcal{A}_L \times \mathcal{A}_M$

numerical analysis
Basic Approach

probabilistic automaton $A_L$

non-deterministic automaton $A_M$

determinization

deterministic automaton $A_M$

product $A_L \times A_M$

numerical analysis

often high complexity, loss of expressive power
Example cont’d
The Product Construction

$A_L$

$\frac{1}{2}, a \xrightarrow{} \frac{1}{2}, b$

$A_M$

$b \xrightarrow{a} a, b$
Example cont’d
The Product Construction

\[ A_L \]
\[
\frac{1}{2}, a \quad \rightarrow \quad \frac{1}{2}, b
\]

\[ A_M \]
\[
b \quad \rightarrow \quad a \quad \rightarrow \quad a, b
\]

already deterministic 😊
Example cont’d
The Product Construction

\( \mathcal{A}_L \)

\( \frac{1}{2}, a \)

\( \frac{1}{2}, b \)

\( \mathcal{A}_M \)

\( b \)

\( a \)

\( a, b \)

\( \mathcal{A}_L \times \mathcal{A}_M \)

\( \frac{1}{2}, b \)

\( \frac{1}{2}, a \)

\( \frac{1}{2}, a \)

\( \frac{1}{2}, b \)
Example cont’d
Meaning of the Product

\[ \mathcal{A}_L \times \mathcal{A}_M \]
Example cont’d

Meaning of the Product

\[ \mathcal{A}_L \times \mathcal{A}_M \]

- \( \mathcal{A}_L \times \mathcal{A}_M \) = probabilistic automaton with non-trivial Büchi condition
Example cont’d
Meaning of the Product

\( A_L \times A_M \)

- \( A_L \times A_M = \) probabilistic automaton with non-trivial Büchi condition
- Defines sub-probability measure on words
Example cont’d
Meaning of the Product

$\mathcal{A}_L \times \mathcal{A}_M$

- $\mathcal{A}_L \times \mathcal{A}_M =$ probabilistic automaton with non-trivial Büchi condition
- Defines sub-probability measure on words
- $Pr_{w \sim L}(w \in M) =$ mass of that sub-probability measure
  = probability $\mathcal{A}_L \times \mathcal{A}_M$ generates a run it accepts
Example cont’d
Computing the Acceptance Probability

Dropping $a, b$ from $\mathcal{A}_L \times \mathcal{A}_M$ yields finite Markov chain:
Example cont’d
Computing the Acceptance Probability

Dropping $a, b$ from $\mathcal{A}_L \times \mathcal{A}_M$ yields finite Markov chain:

• Two facts:
Example cont’d
Computing the Acceptance Probability

Dropping $a, b$ from $\mathcal{A}_L \times \mathcal{A}_M$ yields finite Markov chain:

• Two facts:
  ▸ A finite Markov chain reaches a **bottom strongly-connected component (BSCC)** with probability 1
Example cont’d
Computing the Acceptance Probability

Dropping $a, b$ from $\mathcal{A}_L \times \mathcal{A}_M$ yields finite Markov chain:

- Two facts:
  - A finite Markov chain reaches a **bottom strongly-connected component (BSCC)** with probability 1
Example cont’d
Computing the Acceptance Probability

Dropping $a, b$ from $A_L \times A_M$ yields finite Markov chain:

• Two facts:
  ▪ A finite Markov chain reaches a bottom strongly-connected component (BSCC) with probability 1
  ▪ If BSCC $B$ is reached, then all states in $B$ are visited $\infty$-often with probability 1
Two facts:

- A finite Markov chain reaches a bottom strongly-connected component (BSCC) with probability 1.
- If BSCC $B$ is reached, then all states in $B$ are visited infinitely often with probability 1.

$Pr_{w \sim L}(w \in M) = Pr(\text{reach a BSCC containing an accepting state}) = 1$
### Some Known Results

(Büchi Acceptance)

<table>
<thead>
<tr>
<th>$A_L$</th>
<th>$A_M$</th>
<th>$Pr_{w \sim L}(w \in M) = 1$ ?</th>
</tr>
</thead>
<tbody>
<tr>
<td>probabilistic FA (Markov chain)</td>
<td>DFA</td>
<td>PTIME</td>
</tr>
<tr>
<td></td>
<td>unambiguous FA</td>
<td>PTIME [Baier et al.]</td>
</tr>
<tr>
<td></td>
<td>NFA</td>
<td>PSPACE [Courcoubetis &amp; Yannakakis]</td>
</tr>
<tr>
<td>probabilistic PDA</td>
<td>NFA</td>
<td>EXPTIME [Etessami &amp; Yannakakis]</td>
</tr>
<tr>
<td></td>
<td>non-deterministic PDA</td>
<td>undecidable [Dubslaff et al.]</td>
</tr>
<tr>
<td>probabilistic <strong>visibly</strong> PDA</td>
<td>deterministic visibly PDA</td>
<td>in PSPACE [W. et al.]</td>
</tr>
<tr>
<td></td>
<td>unambiguous visibly PDA</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>non-deterministic visibly PDA</td>
<td>EXPTIME [W. et al.]</td>
</tr>
</tbody>
</table>

- Many other versions by varying type of $L$, $M$, acceptance condition, etc.
Part II
Probabilistic Visibly Pushdown Language Inclusion

Based on a FoSSaCS’22 paper with Christina Gehnen and Joost-Pieter Katoen
Visibly Pushdown Automata (VPA)
[Alur & Madhusudan ’04]

Stack alphabet \( \Gamma = \{Z_0, Z\} \)

- \( Z_0 \) initially on stack
- \( Z_0 \) cannot be popped nor pushed
Visibly Pushdown Automata (VPA)
[Alur & Madhusudan ’04]

Stack alphabet \( \Gamma = \{Z_0, Z\} \)

- \( Z_0 \) initially on stack
- \( Z_0 \) cannot be popped nor pushed
Visibly Pushdown Automata (VPA)

[Alur & Madhusudan ’04]

Stack alphabet $\Gamma = \{Z_0, Z\}$

- $Z_0$ initially on stack
- $Z_0$ cannot be popped nor pushed
Visibly Pushdown Automata (VPA)  
[Alur & Madhusudan ’04]

Stack alphabet $\Gamma = \{Z_0, Z\}$

- $Z_0$ initially on stack
- $Z_0$ cannot be popped nor pushed
Visibly Pushdown Automata (VPA)  
[Alur & Madhusudan ’04]

- Reading must trigger a push
- Reading must trigger a pop
- (Reading symbol from $\Sigma_{int}$ must not change stack height)

$$\Sigma = \Sigma_{push} \cup \Sigma_{pop} \cup \Sigma_{int} = \{a\} \cup \{b\} \cup \emptyset$$

- $a$, $\star / \star Z$
- $b$, $\star / \varepsilon$

Stack alphabet $\Gamma = \{Z_0, Z\}$
- $Z_0$ initially on stack
- $Z_0$ cannot be popped nor pushed
$\omega$-Visibly Pushdown Languages ($\omega$VPL)

$L$ is an $\omega$-visibly pushdown language if $L = L(\mathcal{A})$ for a Büchi VPA $\mathcal{A}$.
Probabilistic Visibly Pushdown Automata (pVPA)

- “Generative” probabilistic pushdown automaton $\mathcal{A}_L$ with trivial Büchi acceptance

\[
\frac{2}{3}, a, \star / \star Z \quad \quad \quad \frac{1}{3}, b, \star / \varepsilon
\]
Probabilistic $\omega$VPL Inclusion

- pVPA $\mathcal{A}_L$
- non-deterministic Büchi VPA $\mathcal{A}_M$
- determinization
- deterministic automaton for $\mathcal{A}_M$
- product $\mathcal{A}_L \times \mathcal{A}_M$
- numerical analysis
Probabilistic $\omega$VPL Inclusion

- pVPA $A_L$
- non-deterministic Büchi VPA $A_M$
- deterministic automaton for $A_M$
- product $A_L \times A_M$
- determinization
- numerical analysis
Determinizing VPA

[Löding, Madhusudan, Serre ‘04]

Every VPA $\mathcal{A}$ with Büchi acceptance can be transformed into an equivalent deterministic VPA $\mathcal{D}$ with $|\mathcal{D}| \in O(2^{|\mathcal{A}|^2})$. $\mathcal{D}$ has a stair-parity acceptance condition.
Stair-Parity Acceptance
Stair-Parity Acceptance

- Priority function $\Omega : States \rightarrow \mathbb{N}$ (like standard parity)
Stair-Parity Acceptance

• Priority function \( \Omega : States \rightarrow \mathbb{N} \) (like standard parity)

• Position \( i \) is a step of an \( \omega \)-run of a PDA \( \iff \forall j \geq i : stackHeight(j) \geq stackHeight(i) \)
Stair-Parity Acceptance

- Priority function $\Omega : \text{States} \rightarrow \mathbb{N}$ (like standard parity)
- Position $i$ is a step of an $\omega$-run of a PDA $\iff \forall j \geq i : \text{stackHeight}(j) \geq \text{stackHeight}(i)$
Stair-Parity Acceptance

- Priority function $\Omega : States \to \mathbb{N}$ (like standard parity)

- Position $i$ is a step of an $\omega$-run of a PDA $\iff \forall j \geq i : stackHeight(j) \geq stackHeight(i)$

- Stair-parity = standard parity evaluated on sequence of steps
  $(\omega$-run is accepting iff the minimum priority seen $\infty$-often at steps is even)
Example

\[ A_L \]

\[ \frac{2}{3}, a, \star / \star Z \]

\[ \frac{1}{3}, b, \star / \varepsilon \]

\[ A_M \]

\[ b, \star / \varepsilon \]

\[ a, \star / \star Z \]

\[ b, \star / \varepsilon \]

\[ 1 \rightarrow 2 \]

already deterministic 😎
Example

\[ \mathcal{A}_L \]

\[ \frac{2}{3}, a, \star / \star Z \]

\[ \mathcal{A}_M \]

\[ \frac{1}{3}, b, \star / \varepsilon \]

already deterministic 😃

\[ \mathcal{A}_L \times \mathcal{A}_M \]

\[ \frac{2}{3}, a, \star / \star Z \]

\[ \frac{1}{3}, b, \star / \varepsilon \]

\[ \frac{1}{3}, b, \star / \varepsilon \]

\[ \frac{2}{3}, a, \star / \star Z \]
Example cont’d

\( A_L \times A_M \)

\( \frac{2}{3}, \star \rightarrow \star Z \)

\( \frac{1}{3}, \star / \varepsilon \)

1

\( \frac{2}{3}, \star / \star Z \)

2

\( \frac{1}{3}, \star / \varepsilon \)
Example cont’d

- Probability that $\mathcal{A}_L \times \mathcal{A}_M$ generates a run it accepts (with stair-parity)?

$\mathcal{A}_L \times \mathcal{A}_M$

\[
\begin{align*}
0.7, \star / \star Z & \quad 0.3, \star / \varepsilon \\
1 & \quad 2
\end{align*}
\]
Remember: Stair-Parity Acceptance

- Stair-parity = standard parity evaluated on sequence of steps
  
  $(\omega)$-run is accepting iff the minimum priority seen $\infty$-often at steps is even
Remember: Stair-Parity Acceptance

- Stair-parity = standard parity evaluated on sequence of steps
  
  (ω-run is accepting iff the minimum priority seen infinitely often at steps is even)
The Markov Chain of Steps: Easy Example

[Esparza, Kucera, Mayr ’04]
The Markov Chain of Steps: Easy Example

[Esparza, Kucera, Mayr ’04]
The Markov Chain of Steps: Easy Example

[Esparza, Kucera, Mayr ’04]

Transition probs of step Markov chain may be irrational, but are expressible in $\mathbb{R}$.
### General Formulas for Step Markov Chain

**Don’t read this**

<table>
<thead>
<tr>
<th>$q \rightarrow r$</th>
<th>$q \perp \rightarrow r$</th>
<th>$q \perp \rightarrow r \perp$</th>
<th>$q \rightarrow r \perp$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q \in Q_{\text{call}}$</td>
<td>[ \left[ \begin{array}{c} [r^\uparrow] \ [q^\uparrow] \end{array} \right] \left( \sum_{r', Z} P_{\text{call}}(q, r'Z) [r'Z \downarrow r] + \sum_{Z} P_{\text{call}}(q, rZ) \right) \sum_{Z} P_{\text{call}}(q, rZ) [r^\uparrow] + \sum_{r', Z} P_{\text{call}}(q, r'Z) [r'Z \downarrow r] \right) = 0 ]</td>
<td>[ \left[ \begin{array}{c} [r^\uparrow] \ [q^\uparrow] \end{array} \right] P_{\text{call}}(q, r) = 0 ]</td>
<td>[ P_{\text{int}}(q, r) = 0 ]</td>
</tr>
<tr>
<td>$q \in Q_{\text{int}}$</td>
<td>[ \left[ \begin{array}{c} [r^\uparrow] \ [q^\uparrow] \end{array} \right] P_{\text{int}}(q, r) = 0 ]</td>
<td>[ P_{\text{int}}(q, r) = 0 ]</td>
<td>[ P_{\text{ret}}(q \perp, r) = 0 ]</td>
</tr>
<tr>
<td>$q \in Q_{\text{ret}}$</td>
<td>undef.</td>
<td>0</td>
<td>undef.</td>
</tr>
</tbody>
</table>
Example cont’d

\[ \mathcal{A}_L \times \mathcal{A}_M \]

\[ \frac{2}{3}, \star \leftrightarrow \star Z \]

\[ \frac{1}{3}, \star \leftrightarrow \epsilon \]

\[ \frac{2}{3}, \star \leftrightarrow \star Z \]

\[ \frac{1}{3}, \star \leftrightarrow \epsilon \]
Example cont’d

\[ A_L \times A_M \]

\[ \frac{2}{3}, \star / \star Z \]

\[ \frac{1}{3}, \star / \varepsilon \]

Markov chain of steps:
Example cont’d

$$\mathcal{A}_L \times \mathcal{A}_M$$

Markov chain of steps:

the only BSCC violates standard parity

$$\implies Pr_{w \sim L}(w \in M) = 0$$
Two Birds, One Stone

With the step Markov chain construction we

• … got rid of the stack

• … reduced stair-parity to standard parity
Two Birds, One Stone

With the step Markov chain construction we

• ... got rid of the stack
• ... reduced stair-parity to standard parity

It follows: For a pVPA $\mathcal{A}_L$ and a Büchi VPA $\mathcal{A}_M$
Two Birds, One Stone

With the step Markov chain construction we

- ... got rid of the stack
- ... reduced stair-parity to standard parity

It follows: For a pVPA $\mathcal{A}_L$ and a Büchi VPA $\mathcal{A}_M$

- ... deciding $Pr_{w \sim L}(w \in M) = 1$ is \textsc{EXPTIME}-complete
Two Birds, One Stone

With the step Markov chain construction we

• … got rid of the stack
• … reduced stair-parity to standard parity

It follows: For a pVPA $\mathcal{A}_L$ and a Büchi VPA $\mathcal{A}_M$

• … deciding $\Pr_{w \sim L}(w \in M) = 1$ is $\text{EXPTIME}$-complete
• … deciding $\Pr_{w \sim L}(w \in M) \geq \lambda$ is in $\text{EXPSPACE}$
Summary & Outlook

- probabilistic automaton $A_L$
- non-deterministic automaton $A_M$
- deterministic automaton for $A_M$
- product $A_L \times A_M$
- numerical analysis
- determinization
Summary & Outlook

• **In this talk:** General approach for probabilistic language inclusion + concrete case of $\omega$VPL
Summary & Outlook

• In this talk: General approach for probabilistic language inclusion + concrete case of $\omega$VPL

• Main technique: Reduce problems to limiting behaviour of finite Markov chains
Summary & Outlook

• **In this talk:** General approach for probabilistic language inclusion + concrete case of $\omega$VPL

• **Main technique:** Reduce problems to limiting behaviour of finite Markov chains

• **Complexity bottleneck:** Determinization
Summary & Outlook

• **In this talk:** General approach for probabilistic language inclusion + concrete case of $\omega$VPL

• **Main technique:** Reduce problems to limiting behaviour of finite Markov chains

• **Complexity bottleneck:** Determinization

• **What’s next?** Unambiguous instead of deterministic, probabilistic automaton for $\mathcal{A}_M$
Summary & Outlook

• **In this talk:** General approach for probabilistic language inclusion + concrete case of $\omega$VPL

• **Main technique:** Reduce problems to limiting behaviour of **finite** Markov chains

• **Complexity bottleneck:** Determinization

• **What’s next?** Unambiguous instead of deterministic, probabilistic automaton for $A_M$

Thank you for listening!
Code for probabilistic robot

dtmc

const int N = 4;

module probot
    x : [0..N] init 2;
    y : [0..N] init 2;

    [] x=0 & y=0 -> 0.5 : (x'=x+1) + 0.5 : (y'=y+1);
    [] x=0 & y=N -> 0.5 : (x'=x+1) + 0.5 : (y'=y-1);
    [] x=N & y=0 -> 0.5 : (x'=x-1) + 0.5 : (y'=y+1);
    [] x=N & y=N -> 0.5 : (x'=x-1) + 0.5 : (y'=y-1);

    [] x=0 & y>0 & y<N -> 1/3: (x'=x+1) + 1/3 : (y'=y-1) + 1/3 : (y'=y+1);
    [] x=N & y>0 & y<N -> 1/3: (x'=x-1) + 1/3 : (y'=y-1) + 1/3 : (y'=y+1);
    [] y=0 & x>0 & x<N -> 1/3: (y'=y+1) + 1/3 : (x'=x-1) + 1/3 : (x'=x+1);
    [] y=N & x>0 & x<N -> 1/3: (y'=y-1) + 1/3 : (x'=x-1) + 1/3 : (x'=x+1);

    [] x>0 & x<N & y>0 & y<N -> 0.25 : (x'=x+1) + 0.25 : (x'=x-1) + 0.25 : (y'=y+1) + 0.25 : (y'=y-1);

endmodule

label "treasure" = x=4 & y=1;
label "safe" = !(x=0 & y=N);