Leaktight and simple probabilistic automata

Edon Kelmendi
Joint work with: Hugo Gimbert

May 23, 2013
Probabilistic automata

- Generalization of DFA by Rabin 1963
- Transition matrices are stochastic
- Formally: $A = (Q, A, i, (M_a)_{a \in A}, F)$
- Word $w = a_1 \ldots a_n$ accepted with probability $p$ if

\[ A(w) = iM_wF = p \]

where $M_w = M_{a_1} \ldots M_{a_n}$
- Stochastic language with respect to a cutpoint $\lambda \in [0, 1]$ (ex: all words $w$ s.t. $A(w) > \lambda$)
Probabilistic automata

- Generalization of DFA by Rabin 1963
- Transition matrices are stochastic
- Formally: $\mathcal{A} = (Q, A, i, (M_a)_{a \in A}, F)$
- Word $w = a_1 \ldots a_n$ accepted with probability $p$ if

$$\mathcal{A}(w) = iM_wF = p$$

where $M_w = M_{a_1} \ldots M_{a_n}$

- Stochastic language with respect to a cutpoint $\lambda \in [0, 1]$ (ex: all words $w$ s.t. $\mathcal{A}(w) > \lambda$)
- Similarly a PA on infinite words
**Value 1 problem**

<table>
<thead>
<tr>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>A cutpoint ( \lambda \in [0, 1] ) is isolated if there is ( \epsilon &gt; 0 ) such that ( \forall w \in A^* ) we have</td>
</tr>
<tr>
<td>[</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem (Isolated cutpoint problem)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given ( A ) and ( \lambda \in [0, 1] ), decide whether ( \lambda ) is isolated.</td>
</tr>
</tbody>
</table>
Value 1 problem

**Definition**

A cutpoint $\lambda \in [0, 1]$ is isolated if there is $\epsilon > 0$ such that $\forall w \in A^*$ we have

$$|A(w) - \lambda| > \epsilon$$

**Problem (Isolated cutpoint problem)**

*Given $A$ and $\lambda \in [0, 1]$, decide whether $\lambda$ is isolated.*

- Isolated cutpoint problem is undecidable [BMT77,GO10]
- Isolated cutpoint problem for the special case of $\lambda = 1$ is called *value 1 problem*
Example of an automaton with value 1

Example

\[
\begin{align*}
\text{Example} & \quad \begin{array}{c}
\text{a} \\
\text{b, } \frac{1}{2}
\end{array} \\
\text{b, } \frac{1}{2} & \quad \begin{array}{c}
b, \frac{1}{2}
\end{array}
\end{align*}
\]
Example of an automaton with value 1

Has value 1 with the sequence of words $((ab)^n)_{n \in \mathbb{N}}$. 
A few classes of PAs with decidable value 1 problem

Goal: Find a robust class of PAs with decidable value 1 problem
Goal: Find a robust class of PAs with decidable value 1 problem

- Hierarchical automata [CSV09]
- #-acyclic automata [GO10]
- Leaktight automata [FGO11]
- Simple automata [CT11]
A few classes of PAs with decidable value 1 problem

- Hierarchical
- #-acyclic
- Leaktight
- Structurally simple
A few classes of PAs with decidable value 1 problem

Hierarchical \( \subseteq \) \#-acyclic \( \subseteq \) Structurally simple

Leaktight

?
Our question is: How are leaktight and structurally simple classes related?
Introduction (Simple automata)

- By Chatterje and Tracol, studied in the context of PAs on infinite words
- Relies on *Decomposition separation theorem* to show that:
  - Limit problem for reachability and parity condition on structurally simple automata is in EXSPACE
  - Implies that value 1 problem is decidable for structurally simple automata
- Structurally simple automata $\subseteq$ simple automata
Definition of a simple automaton

Definition

Given some infinite word $w \in A^\omega$ and initial distribution $\alpha$, $w$ induces a simple process if:

- At every step we can partition $Q$ into two sets: $(A_n)_{n \in \mathbb{N}}$ and $(B_n)_{n \in \mathbb{N}}$ such that:
  - exists $\gamma > 0$, s.t. $\forall q \in A_i$ we have $\mathbb{P}(\alpha \xrightarrow{w \leq i} q) > \gamma$ and,
  - $\lim_n \mathbb{P}(\alpha \xrightarrow{w \leq n} B_n) = 0$

An automaton where every word induces a simple process is called simple
Definition of a simple automaton

Definition

- $\forall q \in A_i, \ P(\alpha \xrightarrow{w \leq i} q) > \gamma$
- $\lim_n P(\alpha \xrightarrow{w \leq n} B_n) = 0$
Definition of a simple automaton

Definition

- \( \forall q \in A_i, \ P(\alpha \xrightarrow{w \leq i} q) > \gamma \)
- \( \lim_n P(\alpha \xrightarrow{w \leq n} B_n) = 0 \)
Definition of a simple automaton

Definition

- \( \forall q \in A_i, \ P(\alpha \xrightarrow{w\leq i} q) > \gamma \)
- \( \lim_n P(\alpha \xrightarrow{w\leq n} B_n) = 0 \)
Definition of a simple automaton

**Definition**

- \( \forall q \in A_i, \quad P(\alpha \xrightarrow{w \leq i} q) > \gamma \)
- \( \lim_n P(\alpha \xrightarrow{w \leq n} B_n) = 0 \)
**Definition of a simple automaton**

<table>
<thead>
<tr>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\forall q \in A_i, \ P(\alpha \xrightarrow{w \leq i} q) &gt; \gamma$</td>
</tr>
<tr>
<td>$\lim_n P(\alpha \xrightarrow{w \leq n} B_n) = 0$</td>
</tr>
</tbody>
</table>

![Diagram showing sets An+2 and Bn+2]
Definition of a simple automaton

Definition

- $\forall q \in A_i, \ P(\alpha \xrightarrow{w\leq i} q) > \gamma$
- $\lim_n P(\alpha \xrightarrow{w\leq n} B_n) = 0$
Definition of a simple automaton

Definition

- $\forall q \in A_i, \ P(\alpha \xrightarrow{w \leq i} q) > \gamma$
- $\lim_n P(\alpha \xrightarrow{w \leq n} B_n) = 0$
When is a process not simple?

Distribution on a state is:
- decreasing progressively to 0
- resetting to some value
When is a process not simple?

Distribution on a state is:

- decreasing progressively to 0
- resetting to some value
When is a process not simple?

Distribution on a state is:
- decreasing progressively to 0
- resetting to some value
When is a process not simple?

Distribution on a state is:

- decreasing progressively to 0
- resetting to some value
When is a process not simple?

Distribution on a state is:
- decreasing progressively to 0
- resetting to some value
When is a process not simple?

Distribution on a state is:
- decreasing progressively to 0
- resetting to some value
When is a process not simple?

Distribution on a state is:

- decreasing progressively to 0
- resetting to some value
When is a process not simple?

Distribution on a state is:
- decreasing progressively to 0
- resetting to some value
Example

An example of a non-simple automaton:

\[
\begin{align*}
q_1 & \xrightarrow{b, (a, \frac{1}{2})} q_2 \\
q_2 & \xrightarrow{a, \frac{1}{2}} q_1
\end{align*}
\]
Example

An example of a non-simple automaton:

The word $w = ab a^2 b a^3 b \ldots a^n b \ldots$ does not induce a simple process
Example

An example of a non-simple automaton:

The word $w = aba^2ba^3b \ldots a^n b \ldots$ does not induce a simple process.

Definition

- $\forall q \in A_i, \ P(\alpha \xrightarrow{w \leq i} q) > \gamma$
- $\lim_n P(\alpha \xrightarrow{w \leq n} B_n) = 0$
A sufficient condition for non-simplicity

For a word $w = a_1 a_2 \cdots \in A^\omega$ if there exist states $q, q' \in Q$ and $\gamma > 0$ such that for i.m. $n$:

- $\mathbb{P}(\alpha \xrightarrow{a_1 a_2 \cdots a_n} q) > \gamma$ and,
- $\mathbb{P}(q \xrightarrow{a_{n+1} \cdots a_{n+k_n}} q') > 0$ for a $k_n \in \mathbb{N}$. 
A sufficient condition for non-simplicity

For a word $w = a_1a_2 \cdots \in A^\omega$ if there exist states $q, q' \in Q$ and $\gamma > 0$ such that for i.m. $n$:

- $\mathbb{P}(\alpha \xrightarrow{a_1a_2\cdots a_n} q) > \gamma$ and,
- $\mathbb{P}(q \xrightarrow{a_{n+1} \cdots a_{n+k_n}} q') > 0$ for a $k_n \in \mathbb{N}$.

Then:

- we can not give a bound for $\mathbb{P}(\alpha \xrightarrow{a_1\cdots a_{n+k_n}} q')$ implies non-simple process
Introduction (Leaktight automata)

- By Fijalkow, Gimbert and Oualhadj, studied in the context of PAs on finite words
- Proof of correctness relies on Simon’s factorization forest theorem
- Value 1 problem is decidable for this class
- Membership in this class is decidable
- Both in PSPACE
Markov monoid

Abstract asymptotic behaviour of sequences of finite words \((u_n)_{n \in \mathbb{N}}\) with *limit words*, which are maps from \(Q \times Q\) to \(\{0, 1\}\). One can
Markov monoid

Abstract asymptotic behaviour of sequences of finite words \((u_n)_{n \in \mathbb{N}}\) with *limit words*, which are maps from \(Q \times Q\) to \(\{0, 1\}\). One can

- concatenate limit words \((u_1u_2)\),
Markov monoid

Abstract asymptotic behaviour of sequences of finite words \((u_n)_{n \in \mathbb{N}}\) with *limit words*, which are maps from \(Q \times Q\) to \(\{0, 1\}\). One can

- concatenate limit words \((u_1u_2)\),
- iterate them \((u^\#)\), where we remove incoming edges to transient states
Markov monoid

Abstract asymptotic behaviour of sequences of finite words \((u_n)_{n \in \mathbb{N}}\) with *limit words*, which are maps from \(Q \times Q\) to \(\{0, 1\}\). One can

- concatenate limit words \((u_1u_2)\),
- iterate them \((u^\#)\), where we remove incoming edges to transient states.

**Definition**

Markov monoid is the smallest set containing \(a\) for \(a \in A\) and closed under the operations above.

In the extended Markov monoid, we keep track of edges deleted by iteration in second component: \((u, u^+)\).
Value 1 witness

Deciding value 1 is done by looking for a value 1 witness in the Markov monoid:

**Definition**

A limit word $\mathbf{u}$ is called a value 1 witness if for all $q \in Q$:

$$\mathbf{u}(i, q) = 1 \implies q \in F$$
Value 1 witness

Definition
A limit word $u$ is called a value 1 witness if for all $q \in Q$:

$$u(i, q) = 1 \implies q \in F$$

Example

![Automaton diagram]

- $q_1$ is the start state.
- $q_2$ and $q_3$ are accepting states.
- The transitions are labeled as follows:
  - $q_1 \rightarrow q_4$
  - $q_4 \rightarrow q_5$
  - $q_3 \rightarrow q_5$
Value 1 witness

**Definition**

A limit word $u$ is called a value 1 witness if for all $q \in Q$:

$$u(i, q) = 1 \implies q \in F$$

**Example**
Algorithm

Require: A probabilistic automaton.

1. $\mathcal{G} \leftarrow \{a \mid a \in A\} \cup \{1\}$.
2. repeat
3. 
4. if there is $u, v \in \mathcal{G}$ such that $u \cdot v \notin \mathcal{G}$ then
5. 
6. add $u \cdot v$ to $\mathcal{G}$
7. 
8. if there is $u \in \mathcal{G}$ such that $u$ is idempotent and $u^\# \notin \mathcal{G}$ then
9. 
10. add $u^\#$ to $\mathcal{G}$
11. until there is nothing to add
12. if there is a value 1 witness in $\mathcal{G}$ then
13. 
14. return true
15. else
16. return false
Leak witness

Similarly membership is decided by looking for a **leak-witness** in the extended Markov monoid:

**Definition**

An extended limit word \((u, u_+)\) is a leak-witness if there are states \(r, t \in Q\) such that:

- \(r\) is \(u\)-recurrent
- \(u_+(r, t) = 1\)
- \(u(t, r) = 0\)
Leak witness

**Definition**

- $r$ is $u$-recurrent
- $u_+(r, t) = 1$
- $u(t, r) = 0$

**Example**

![Diagram showing a leak witness](image-url)
Leak witness

**Definition**
- $r$ is $u$-recurrent
- $u_+(r, t) = 1$
- $u(t, r) = 0$

**Example**

![Diagram showing leak witness](attachment:image.png)
<table>
<thead>
<tr>
<th>Probabilistic automata and value 1 problem</th>
<th>Simple automata</th>
<th>Leaktight automata</th>
<th>Comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leaktight and simple automata</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Simple automata</th>
<th>Leaktight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extended support graph</td>
<td>Markov monoid</td>
</tr>
<tr>
<td></td>
<td>Leak witness</td>
</tr>
</tbody>
</table>
Leaktight and simple automata

Simple automata

Leaktight

Extended support graph

Markov monoid

Non-simplicity witness

Leak witness
Non-simplicity witness

Relation between leaktight and simple automata is in the non-simplicity witness:

**Definition**

The triple \((u_1, u_2, u_3)\) is called a non-simplicity witness if there are states \(r, t \in Q\) such that:

- \(r\) is \(u_1 u_2^\# u_3\)-recurrent
- \(u_1 u_2(r, t) = 1\)
- \(t\) is \(u_2\)-transient
Non-simplicity witness

Definition

- $r$ is $u_1 u_2 \# u_3$-recurrent
- $u_1 u_2(r, t) = 1$
- $t$ is $u_2$-transient
Simplicity witness implies non-simple process

Proof sketch:
- Concatenate a particular sequence of words that reifies $u_1u_2\#u_3$, show that it induces a non-simple process
- Give a bound for distribution on $r$ in i.m. steps,
- Show that we reach $t$ from these points
- Show that $t$ cannot be bounded
\[ \exists \text{ leak implies } \exists \text{ non-simplicity witness} \]

Proof sketch:
- Look for elements of the extended Markov monoid \((u, u_+)\) for which:
  - \(r\) is \(u\)-recurrent,
  - \(u(r, t) = 0\),
  - \(u_+(r, t) = 1\)

Finally decompose \((u, u_+)\) into a triple that is a non-simplicity witness.
Strict inclusion

Inclusion is strict from the example given earlier:

Example

The word $w = aba^2ba^3b \ldots a^nb \ldots$ does not induce a simple process
But the extended Markov monoid of this automaton contains no leak witness:
But the extended Markov monoid of this automaton contains no leak witness:

\[ q_1 \xrightarrow{(b, b)} q_2 \]
But the extended Markov monoid of this automaton contains no leak witness:
But the extended Markov monoid of this automaton contains no leak witness:

\[
\begin{array}{c}
q_1 \\
\uparrow \\
+ \\
\downarrow \\
q_2
\end{array}
\]

\((a\neq, a)\)
But the extended Markov monoid of this automaton contains no leak witness:

$$\begin{align*}
q_1 \xrightarrow{+} & q_2 \\
q_2 \xrightarrow{+} & q_1
\end{align*}$$

$$(ba\#, ba)$$
Strict inclusion

But the extended Markov monoid of this automaton contains no leak witness:

\[\begin{align*}
q_1 & \xrightarrow{+} q_2 \\
q_2 & \xrightarrow{+} (ba\#, ba)
\end{align*}\]

And it is the monoid defined by these relations:

\[
\begin{align*}
a^2 &= a, & b^2 &= b, & ab &= b, & a\#a &= a\#, & a\#b &= b, \\
a\#^2 &= a\#, & (ba\#)^2 &= ba\#, & a\#a &= a\#, & a\#b &= b,
\end{align*}\]
A conjecture

In the case of leaktight automata, we do qualitative analysis on the PA. Previous work in quantum automata do quantitave analysis also.
A conjecture

In the case of leaktight automata, we do qualitative analysis on the PA.
Previous work in quantum automata do quantitative analysis also.

**Corollary:** the value 1 problem is decidable for the class of automata with orthogonal matrices.

**Conjecture:** remains decidable with invertible matrices.
Leaktight and simple automata were developed with dissimilar methods but simple $\subset$ leaktight.
Leaktight and simple automata were developed with dissimilar methods but simple $\subset$ leaktight.

Problem (open)

*Given an isolated cutpoint* $0 < \lambda < 1$ *decide whether*

$$\mathcal{L}_>(\mathcal{A}) = \{ w \in A^* \mid \mathcal{A}(w) > \lambda \},$$

*is empty or not.*