Deciding the value 1 problem
for probabilistic leaktight automata

Séminaire Automates

Nathanaël Fijalkow,
joint work with Hugo Gimbert and Youssouf Oualhadj

LIAFA, CNRS & Université Denis Diderot - Paris 7, France
nath@liafa.jussieu.fr

November 25th, 2011
Outline

1. The value 1 problem for probabilistic automata
   - Definitions
   - Deciding the isolation problem

2. An algebraic solution to the limitedness problem for distance automata
   - Taking a step back: weighted automata
   - Leung’s algorithm

3. Towards an algebraic treatment of probabilistic automata
   - First tries
   - Leaks
   - The good semiring
   - The completeness proof using Simon’s theorem
Outline

1. **The value 1 problem for probabilistic automata**
   - Definitions
   - Deciding the isolation problem

2. **An algebraic solution to the limitedness problem for distance automata**
   - Taking a step back: weighted automata
   - Leung’s algorithm

3. **Towards an algebraic treatment of probabilistic automata**
   - First tries
   - Leaks
   - The good semiring
   - The completeness proof using Simon’s theorem
Outline

1. The value 1 problem for probabilistic automata
   - Definitions
     - Deciding the isolation problem

2. An algebraic solution to the limitedness problem for distance automata
   - Taking a step back: weighted automata
   - Leung’s algorithm

3. Towards an algebraic treatment of probabilistic automata
   - First tries
   - Leaks
   - The good semiring
   - The completeness proof using Simon’s theorem
Probabilistic automata (Rabin, 1963)

\[ \mathbb{P}_A : A^* \rightarrow [0, 1] \]
Cutpoint and value

Fix $0 < \lambda \leq 1$, define:

$$L_\lambda = \{w \mid \mathbb{P}_A(w) \geq \lambda\}.$$
Cutpoint and value

Fix $0 < \lambda \leq 1$, define:

$$L_\lambda = \{w \mid \mathbb{P}_A(w) \geq \lambda\}.$$ 

$\lambda$ is isolated if there exists $\delta > 0$ such that for all $w \in A^*$, we have

$$|\mathbb{P}_A(w) - \lambda| \geq \delta$$
Cutpoint and value

Fix $0 < \lambda \leq 1$, define:

$$L_\lambda = \{ w | P_A(w) \geq \lambda \}.$$

$\lambda$ is isolated if there exists $\delta > 0$ such that for all $w \in A^*$, we have

$$|P_A(w) - \lambda| \geq \delta$$

Theorem (Rabin, 1963)

*If $\lambda$ is isolated, then $L_\lambda$ is a regular language.*
Outline

1. The value 1 problem for probabilistic automata
   - Definitions
   - Deciding the isolation problem

2. An algebraic solution to the limitedness problem for distance automata
   - Taking a step back: weighted automata
   - Leung’s algorithm

3. Towards an algebraic treatment of probabilistic automata
   - First tries
   - Leaks
   - The good semiring
   - The completeness proof using Simon’s theorem
The isolation problem

Fix $0 \leq \lambda \leq 1$, the isolation problem is:

**Instance:** a probabilistic automaton $A$

**Question:** is $\lambda$ isolated in $A$?
The isolation problem

Fix $0 \leq \lambda \leq 1$, the isolation problem is:

**Instance:** a probabilistic automaton $A$

**Question:** is $\lambda$ isolated in $A$?

For $0 < \lambda < 1$, Bertoni showed that this is undecidable (in 1974)!
The value 1 problem

For $\lambda = 1$ the isolation problem can be formulated as: “are there words accepted by $A$ with probability arbitrarily close to 1”.
The value 1 problem

For $\lambda = 1$ the isolation problem can be formulated as: “are there words accepted by $\mathcal{A}$ with probability arbitrarily close to 1”.
Equivalently, define $\text{val}(\mathcal{A}) = \sup_w P_{\mathcal{A}}(w)$, then the problem is:

$\text{val}(\mathcal{A}) \overset{?}{=} 1$.
The value 1 problem

For $\lambda = 1$ the isolation problem can be formulated as: “are there words accepted by $A$ with probability arbitrarily close to 1”.

Equivalently, define $\text{val}(A) = \sup_w P_A(w)$, then the problem is:

“$\text{val}(A) \geq 1$”.

Theorem (Gimbert, Oualhadj, 2010)

*The value 1 problem is undecidable.*
An intuition

has value 1 if and only if $x > \frac{1}{2}$. 
A very restricted case

Theorem (Fijalkow, Gimbert, Oualhadj, 2011)

*The isolation problem is (still) undecidable if we randomise only on one transition.*
Sketch of proof (1)
Sketch of proof (2)

Given $A$ reading words from $A^*$, we construct $B$ over a new alphabet $B$, with one probabilistic transition, and a morphism $\hat{\_} : A^* \rightarrow B^*$ such that:

$$\forall w \in A^*, P_A(w) = P_B(\hat{w}).$$
Sketch of proof (2)

Given $A$ reading words from $A^*$, we construct $B$ over a new alphabet $B$, with one probabilistic transition, and a morphism $\hat{\_}: A^* \rightarrow B^*$ such that:

$$\forall w \in A^*, \mathbb{P}_A(w) = \mathbb{P}_B(\hat{w}).$$

$$\hat{a} = \text{check}(a, q_0) \cdot \text{apply}(a, q_0) \ldots \text{check}(a, q_{n-1}) \cdot \text{apply}(a, q_{n-1}) \cdot \text{merge}.$$
Given $A$ reading words from $A^*$, we construct $B$ over a new alphabet $B$, with one probabilistic transition, and a morphism $\hat{\cdot} : A^* \rightarrow B^*$ such that:

$$\forall w \in A^*, \mathbb{P}_A(w) = \mathbb{P}_B(\hat{w}).$$

$$\hat{a} = \text{check}(a, q_0) \cdot \ldots \cdot \text{apply}(a, q_0) \ldots \text{check}(a, q_{n-1}) \cdot \ldots \cdot \text{apply}(a, q_{n-1}) \cdot \text{merge}.$$
Sketch of proof (3)
Sketch of proof (3)

\[ \mathcal{A} \]

\[ a \]

\[ p \rightarrow p \]

\[ q \rightarrow r \]

\[ r \rightarrow q \]

\[ \mathcal{B} \]

\[ \text{check}(a, p) \cdot \ast \cdot \text{apply}(a, p) \]

\[ p \rightarrow \tilde{p} \]

\[ q \]

\[ r \]
Sketch of proof (3)

\[ \mathcal{A} \]

\[ a \]

\[ p \rightarrow p \]

\[ q \rightarrow r \]

\[ r \rightarrow q \]

\[ \mathcal{B} \]

\[ \text{check}(a, q) \cdot * \cdot \text{apply}(a, q) \]

\[ p \rightarrow \tilde{p} \]

\[ q \rightarrow \tilde{r} \]

\[ r \rightarrow q \]
Sketch of proof (3)

\[ \mathcal{A} \]

\[
\begin{align*}
a & \quad \rightarrow \quad p \\
p & \quad \rightarrow \quad p \\
q & \quad \rightarrow \quad r \\
r & \quad \rightarrow \quad q
\end{align*}
\]

\[ \mathcal{B} \]

\[
\begin{align*}
\text{check}(a, r) \cdot \ast \cdot \text{apply}(a, r) \\
p & \quad \rightarrow \quad \tilde{p} \\
q & \quad \rightarrow \quad \tilde{r} \\
r & \quad \rightarrow \quad \tilde{q}
\end{align*}
\]
Sketch of proof (3)

\[ A \]

\[ p \quad a \quad q \]

\[ r \quad p \quad q \]

\[ B \]

\[ p \quad \tilde{p} \quad p \]

\[ q \quad \tilde{r} \quad r \]

\[ r \quad \tilde{q} \quad q \]

merge
Sketch of proof (4)

\[ \mathcal{B} \text{ is unable to check that a letter check}(a, q) \text{ is actually followed by the corresponding apply}(a, q): \text{inbetween, it will go through } s_* \text{ and “forget” the state it was in.} \]
Sketch of proof (4)

$B$ is unable to check that a letter check$(a, q)$ is actually followed by the corresponding apply$(a, q)$: inbetween, it will go through $s_*$ and “forget” the state it was in.

$$\sup_{n} \mathbb{P}_{B}((\hat{w} \cdot \text{finish})^n) = \mathbb{P}_{A}(w)$$
Assume $p \in F$, $q \notin F$ and $i$ is the initial state of a (deterministic) automaton recognizing $(\hat{A}^* \cdot \text{finish})^*$. 

![Diagram](image)
Our objective

Define a large and interesting subclass of probabilistic automata for which the value 1 problem is decidable.
Outline

1. The value 1 problem for probabilistic automata
   - Definitions
   - Deciding the isolation problem

2. An algebraic solution to the limitedness problem for distance automata
   - Taking a step back: weighted automata
   - Leung’s algorithm

3. Towards an algebraic treatment of probabilistic automata
   - First tries
   - Leaks
   - The good semiring
   - The completeness proof using Simon’s theorem
Outline

1. The value 1 problem for probabilistic automata
   - Definitions
   - Deciding the isolation problem

2. An algebraic solution to the limitedness problem for distance automata
   - Taking a step back: weighted automata
   - Leung’s algorithm

3. Towards an algebraic treatment of probabilistic automata
   - First tries
   - Leaks
   - The good semiring
   - The completeness proof using Simon’s theorem
Probabilistic automata VS distance automata

Consider a semiring \((\mathcal{K}, +, \cdot)\). An automaton computes in the semiring \(\mathcal{K}\) if \(\text{val}(w) = \sum\{\Pi(\rho) \mid \rho \text{ is a run over } w\}\).
Probabilistic automata VS distance automata

Consider a semiring \((\mathcal{K}, +, \cdot)\). An automaton computes in the semiring \(\mathcal{K}\) if \(\text{val}(w) = \sum \{ \Pi(\rho) \mid \rho \text{ is a run over } w \} \).

- Classical automata compute in the boolean semiring.
- Probabilistic automata compute in \((\mathbb{R}, +, \cdot)\) (there is a catch here).
- Distance automata compute in the tropical semiring \((\mathbb{N} \cup \{\infty\}, \min, +)\). Here is an example:
The value 1 problem VS the limitedness problem

The value 1 problem for probabilistic automata is:

“are there words accepted with probability arbitrarily close to 1?”.  

The unlimitedness problem for distance automata is:

“are there words with arbitrarily high value?”.  

The value 1 problem VS the limitedness problem

The value 1 problem for probabilistic automata is:

“are there words accepted with probability arbitrarily close to 1?”.

undecidable

The unlimitedness problem for distance automata is:

“are there words with arbitrarily high value?”.

decidable (Hashiguchi, 1988)
Outline

1. The value 1 problem for probabilistic automata
   - Definitions
   - Deciding the isolation problem

2. An algebraic solution to the limitedness problem for distance automata
   - Taking a step back: weighted automata
   - Leung’s algorithm

3. Towards an algebraic treatment of probabilistic automata
   - First tries
   - Leaks
   - The good semiring
   - The completeness proof using Simon’s theorem
Weighted automata using algebra (Schützenberger)

\[ \langle a \rangle = \begin{pmatrix} 0 & \infty & \infty \\ \infty & 1 & \infty \\ \infty & \infty & 0 \end{pmatrix} \quad \langle b \rangle = \begin{pmatrix} 0 & 0 & \infty \\ \infty & \infty & 0 \\ \infty & \infty & 0 \end{pmatrix} \]

\[ I = \begin{pmatrix} 0 & 0 & \infty \end{pmatrix} \quad F = \begin{pmatrix} \infty \\ 0 \\ 0 \end{pmatrix} \]
Weighted automata using algebra (Schützenberger)

\[
\langle a \rangle = \begin{pmatrix}
0 & \infty & \infty \\
\infty & 1 & \infty \\
\infty & \infty & 0
\end{pmatrix} \quad \quad \langle b \rangle = \begin{pmatrix}
0 & 0 & \infty \\
\infty & \infty & 0 \\
\infty & \infty & 0
\end{pmatrix}
\]

\[
I \cdot \langle aaabaa \rangle \cdot F = \begin{pmatrix}
0 & 0 & \infty \\
\infty & \infty & 3 \\
\infty & \infty & 0
\end{pmatrix} = 2
\]

\[
\left\{ \begin{array}{l}
k \in \mathbb{N} \quad \text{best run has value } k \\
\infty \quad \text{no run}
\end{array} \right.
\]
Towards Leung’s algorithm:  \( \dagger \)-expressions

\[
\text{val}( (a^n \cdot b)^n \cdot a^n ) = n
\]
Towards Leung’s algorithm: $\#$-expressions

$$\text{val}((a^n \cdot b)^n \cdot a^n) = n$$

An unlimitedness witness is $(a^\# \cdot b)^\# \cdot a^\#$. 
Towards Leung’s algorithm: stabilization

\[ \langle a \rangle = \begin{pmatrix}
0 & \infty & \infty & \infty & \infty \\
\infty & 1 & \infty & \infty & \infty \\
\infty & \infty & 0 & \infty & \infty \\
\infty & \infty & \infty & 0 & \infty \\
\infty & \infty & \infty & \infty & 0
\end{pmatrix} \]

\[ \langle a^n \rangle = \begin{pmatrix}
0 & \infty & \infty & \infty & \infty \\
\infty & n & \infty & \infty & \infty \\
\infty & \infty & 0 & \infty & \infty \\
\infty & \infty & \infty & 0 & \infty \\
\infty & \infty & \infty & \infty & 0
\end{pmatrix} \]

\[ \langle a^\# \rangle = \begin{pmatrix}
0 & \infty & \infty & \infty & \infty \\
\infty & \infty & \infty & \infty & \infty \\
\infty & \infty & 0 & \infty & \infty \\
\infty & \infty & \infty & 0 & \infty \\
\infty & \infty & \infty & \infty & 0
\end{pmatrix} \]
Towards Leung’s algorithm: stabilization

\[ \langle a \rangle = \begin{pmatrix} 0 & \infty & \infty & \infty \\ \infty & 1 & \infty & \infty \\ \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 \end{pmatrix} \quad \langle a^n \rangle = \begin{pmatrix} 0 & \infty & \infty & \infty \\ \infty & n & \infty & \infty \\ \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 \end{pmatrix} \]

\[ \langle a^\# \rangle = \begin{pmatrix} 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 0 \end{pmatrix} \]

\{ \begin{align*}
  k \in \mathbb{N} & \quad \text{best run has value } k \\
  \infty & \quad \text{arbitrarily high value} \\
  \infty & \quad \text{no run} 
\end{align*} \]
Leung’s algorithm

To ensure termination we project the tropical semiring \((\mathbb{N} \cup \infty, \min, +)\) into the finite semiring \((\{0, 1, \infty\}, \min, +)\).
Leung’s algorithm

To ensure termination we project the tropical semiring \((\mathbb{N} \cup \infty, \min, +)\) into the finite semiring \((\{0, 1, \infty\}, \min, +)\).

Compute a monoid inside the monoid \(\mathcal{M}_{\mathbb{Q} \times \mathbb{Q}}(\{0, 1, \infty\}, \min, +)\).

- Compute \(\langle a \rangle\) for \(a \in A\).
- Close under product and stabilization.
- If there exists a matrix \(M\) such that \(I \cdot M \cdot F = \infty\) then “unlimited”, otherwise “limited”.
Leung’s algorithm: termination and correction

Termination: the monoid $\mathcal{M}_{Q \times Q}(\{0, 1, \infty\}, \text{min}, +)$ is finite.
Leung’s algorithm: termination and correction

Termination: the monoid $\mathcal{M}_{Q \times Q}(\{0, 1, \infty\}, \min, +)$ is finite.

Correction: the proof is complicated, and relies on Simon’s theorem.
Outline

1. The value 1 problem for probabilistic automata
   - Definitions
   - Deciding the isolation problem

2. An algebraic solution to the limitedness problem for distance automata
   - Taking a step back: weighted automata
   - Leung’s algorithm

3. Towards an algebraic treatment of probabilistic automata
   - First tries
   - Leaks
   - The good semiring
   - The completeness proof using Simon’s theorem
Our objective (again)

Decide the value 1 problem for a subclass of probabilistic automata, by **algebraic** and **non-numerical** means.
Our objective (again)

Decide the value 1 problem for a subclass of probabilistic automata, by algebraic and non-numerical means.

- **algebraic**: focus on the automaton structure,
- **non-numerical**: abstract away the values.
Our objective (again)

Decide the value 1 problem for a subclass of probabilistic automata, by \textit{algebraic} and \textit{non-numerical} means.

- \textbf{algebraic}: focus on the automaton structure,
- \textbf{non-numerical}: abstract away the values.

Hence we consider non-deterministic automata: we project \((\mathbb{R}, +, \cdot)\) into the boolean semiring \((\{0, 1\}, +, \cdot)\).
Our objective (again)

Decide the value 1 problem for a subclass of probabilistic automata, by **algebraic** and **non-numerical** means.

- **algebraic**: focus on the automaton structure,
- **non-numerical**: abstract away the values.

Hence we consider non-deterministic automata: we project \((\mathbb{R}, +, \cdot)\) into the boolean semiring \((\{0, 1\}, +, \cdot)\).
Outline

1. The value 1 problem for probabilistic automata
   - Definitions
   - Deciding the isolation problem

2. An algebraic solution to the limitedness problem for distance automata
   - Taking a step back: weighted automata
   - Leung’s algorithm

3. Towards an algebraic treatment of probabilistic automata
   - First tries
   - Leaks
   - The good semiring
   - The completeness proof using Simon’s theorem
Defining stabilization

$\langle a \rangle = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$

In $\langle a \rangle$, the state 1 is transient and the state 2 is recurrent.
Defining stabilization

\[ \langle a \rangle = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \langle a^\# \rangle = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} \]

In \( \langle a \rangle \), the state 1 is transient and the state 2 is recurrent.
Defining stabilization

\[
\langle a \rangle = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \langle a^\# \rangle = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}
\]

In \( \langle a \rangle \), the state 1 is transient and the state 2 is recurrent.

\[
M^\#(s, t) = \begin{cases} 
1 & \text{if } M(s, t) = 1 \text{ and } t \text{ recurrent in } M, \\
0 & \text{otherwise.}
\end{cases}
\]

(This definition gives an asymmetric monoid, this is unusual.)
A first algorithm

Compute a monoid inside the finite monoid \( \mathcal{M}_{\mathbb{Q} \times \mathbb{Q}}(\{0, 1\}, +, \cdot) \).

- Compute \( \langle a \rangle \) for \( a \in A \):

\[
\langle a \rangle(s, t) = \begin{cases} 
1 & \text{if } \mathbb{P}_A(s \xrightarrow{a} t) > 0, \\
0 & \text{otherwise}.
\end{cases}
\]

- Close under product and stabilization.
A first algorithm

Compute a monoid inside the **finite** monoid $\mathcal{M}_{\mathbb{Q} \times \mathbb{Q}}(\{0, 1\}, +, \cdot)$.

- Compute $\langle a \rangle$ for $a \in A$:
  \[
  \langle a \rangle(s, t) = \begin{cases} 
  1 & \text{if } \mathbb{P}_A(s \xrightarrow{a} t) > 0, \\
  0 & \text{otherwise.}
  \end{cases}
  \]

- Close under product and stabilization.
- If there exists a matrix $M$ such that
  \[
  \forall t \in \mathbb{Q}, \quad M(s_0, t) = 1 \Rightarrow t \in F
  \]
  then “$A$ has value 1”, otherwise “$A$ does not have value 1”.

An example
An example
An example
An example
An example
An example
Correctness

Theorem

*If there exists a matrix $M$ such that*

$$\forall t \in Q, \quad M(s_0, t) = 1 \Rightarrow t \in F$$

*then $A$ has value 1.*
Correctness

Theorem

*If there exists a matrix \( M \) such that*

\[
\forall t \in Q, \quad M(s_0, t) = 1 \implies t \in F
\]

*then \( A \) has value 1.*

But the value 1 problem is undecidable, so…
No completeness

Left and right parts are symmetric, so for all $M$:

\[ M(0, L_2) = 1 \iff M(0, R_2) = 1. \]
Outline

1. The value 1 problem for probabilistic automata
   - Definitions
   - Deciding the isolation problem

2. An algebraic solution to the limitedness problem for distance automata
   - Taking a step back: weighted automata
   - Leung’s algorithm

3. Towards an algebraic treatment of probabilistic automata
   - First tries
   - Leaks
   - The good semiring
   - The completeness proof using Simon’s theorem
An example
An example
An example

\[ \langle a^\# \cdot b \rangle \]
Outline

1. The value 1 problem for probabilistic automata
   - Definitions
   - Deciding the isolation problem

2. An algebraic solution to the limitedness problem for distance automata
   - Taking a step back: weighted automata
   - Leung’s algorithm

3. Towards an algebraic treatment of probabilistic automata
   - First tries
   - Leaks
   - The good semiring
   - The completeness proof using Simon’s theorem
A three-valued semiring

Instead of \((\{0, 1\}, +, \cdot)\) we compute in \((\{0, \varepsilon, 1\}, +, \cdot)\), where \(0 < \varepsilon < 1\).
A three-valued semiring

Instead of \((\{0, 1\}, +, \cdot)\) we compute in \((\{0, \varepsilon, 1\}, +, \cdot)\), where \(0 < \varepsilon < 1\).

\[
\begin{array}{c|c|c|c}
+ & 0 & \varepsilon & 1 \\
0 & 0 & \varepsilon & 1 \\
\varepsilon & \varepsilon & \varepsilon & 1 \\
1 & 1 & 1 & 1 \\
\end{array}
\quad
\begin{array}{c|c|c|c}
\cdot & 0 & \varepsilon & 1 \\
0 & 0 & 0 & 0 \\
\varepsilon & 0 & \varepsilon & \varepsilon \\
1 & 0 & \varepsilon & 1 \\
\end{array}
\]
The algorithm

- Compute $\langle a \rangle$ for $a \in A$:
  
  \[
  \langle a \rangle(s, t) = \begin{cases} 
  1 & \text{if } \mathbb{P}_A(s \xrightarrow{a} t) > 0, \\
  0 & \text{otherwise.}
  \end{cases}
  \]

- Close under product and stabilization:
  
  \[
  M^\#(s, t) = \begin{cases} 
  1 & \text{if } M(s, t) = 1 \text{ and } t \text{ recurrent in } M, \\
  \varepsilon & \text{if } M(s, t) = 1 \text{ and } t \text{ transient in } M, \\
  \varepsilon & \text{if } M(s, t) = \varepsilon, \\
  0 & \text{otherwise.}
  \end{cases}
  \]
The algorithm

- Compute $\langle a \rangle$ for $a \in A$:

$$\langle a \rangle(s, t) = \begin{cases} 
1 & \text{if } \mathbb{P}_A(s \xrightarrow{a} t) > 0, \\
0 & \text{otherwise}. 
\end{cases}$$

- Close under product and stabilization:

$$M^\#(s, t) = \begin{cases} 
1 & \text{if } M(s, t) = 1 \text{ and } t \text{ recurrent in } M, \\
\varepsilon & \text{if } M(s, t) = 1 \text{ and } t \text{ transient in } M, \\
\varepsilon & \text{if } M(s, t) = \varepsilon, \\
0 & \text{otherwise}. 
\end{cases}$$

- If there exists a matrix $M$ such that

$$\forall t \in Q, \quad M(s_0, t) = 1 \Rightarrow t \in F$$

then “$A$ has value 1”, otherwise “$A$ does not have value 1”.
The control lemma

We say that a word $w$ reify $M$ in $\mathcal{M}_A$ if:

- $M = \langle a \rangle$ and $w = a$;
- $M = M_1 \cdot M_2$ and there exists $w_1$ and $w_2$ reifying $M_1$ and $M_2$, respectively, such that $w = w_1 \cdot w_2$;
- $M = N^\#$ and there exists $x_1, \ldots, x_n$ each reifying $N$, such that $w = x_1 \ldots x_n$ for some $n \geq 1$. 
The control lemma

We say that a word \( w \) reify \( M \) in \( \mathcal{M}_A \) if:

- \( M = \langle a \rangle \) and \( w = a \);
- \( M = M_1 \cdot M_2 \) and there exists \( w_1 \) and \( w_2 \) reifying \( M_1 \) and \( M_2 \), respectively, such that \( w = w_1 \cdot w_2 \);
- \( M = N^\# \) and there exists \( x_1, \ldots, x_n \) each reifying \( N \), such that \( w = x_1 \ldots x_n \) for some \( n \geq 1 \).

Lemma (The control lemma)

For all \( M \) in \( \mathcal{M}_A \), for all words \( w \) reifying \( M \), for all states \( s, t \) in \( Q \), we have:

\[
M(s, t) \neq 0 \iff \mathbb{P}_A(s \xrightarrow{w} t) > 0.
\]
Leaktight automata

Definition
An automaton $A$ is leaktight if for all $M$, we have

$$M(s, t) = \varepsilon \implies (s \text{ is transient}) \text{ or } (M(t, s) = 1).$$
Leaktight automata

Definition
An automaton $A$ is leaktight if for all $M$, we have

$$M(s, t) = \varepsilon \implies (s \text{ is transient}) \text{ or } (M(t, s) = 1).$$

Theorem (Fijalkow, Gimbert, Oualhadj)

*The value 1 problem is decidable for leaktight automata.*
Outline

1. The value 1 problem for probabilistic automata
   - Definitions
   - Deciding the isolation problem

2. An algebraic solution to the limitedness problem for distance automata
   - Taking a step back: weighted automata
   - Leung’s algorithm

3. Towards an algebraic treatment of probabilistic automata
   - First tries
   - Leaks
   - The good semiring
   - The completeness proof using Simon’s theorem
Decomposition trees

Fact
The set $\mathcal{M}_A$ computed by the algorithm is a stabilization monoid.

Definition
A decomposition tree of a word $w \in A^+$ is a finite unranked ordered tree, whose nodes have labels in $(A^+, \mathcal{M}_A)$ and such that:

- the root is labeled by $(w, u)$, for some $u \in \mathcal{M}_A$,
- every leaf is labeled by $(a, \langle a \rangle)$ where $a$ is a letter,
- every internal node with two children labeled by $(w_1, u_1)$ and $(w_2, u_2)$ is labeled by $(w_1 \cdot w_2, u_1 \cdot u_2)$,
- for every internal node with three or more children, there exists $e \in E(M)$ such that the node is labeled by $(w_1 \ldots w_n, e^\#)$ and its children are labeled by $(w_1, e), \ldots, (w_n, e)$. 
Bounding the height of a decomposition tree

In a decomposition tree, an iteration node is said discontinuous if $M^\# \neq M$. The span of a decomposition tree is the maximal length of a path that contains no discontinuous path.
Bounding the height of a decomposition tree

In a decomposition tree, an iteration node is said discontinuous if 
\( M^\# \neq M \). The span of a decomposition tree is the maximal length of a path that contains no discontinuous path.

Theorem (Simon, 1990)

*Every word* \( w \in A^+ \) *has a decomposition tree whose span is less than* 
\( 3 \cdot |\mathcal{M}_A| \).
Bounding the height of a decomposition tree

In a decomposition tree, an iteration node is said discontinuous if $M^# \neq M$. The span of a decomposition tree is the maximal length of a path that contains no discontinuous path.

Theorem (Simon, 1990)

*Every word $w \in A^+$ has a decomposition tree whose span is less than $3 \cdot |M_A|$.*

Lemma (Simon, 1990)

*Let $M \in E(M_A)$, if $M^# \neq M$, then $M^# < \mathcal{J} M$.*
Bounding the height of a decomposition tree

In a decomposition tree, an iteration node is said discontinuous if $M^\# \neq M$. The span of a decomposition tree is the maximal length of a path that contains no discontinuous path.

Theorem (Simon, 1990)

*Every word $w \in A^+$ has a decomposition tree whose span is less than $3 \cdot |\mathcal{M}_A|$.*

Lemma (Simon, 1990)

*Let $M \in E(\mathcal{M}_A)$, if $M^\# \neq M$, then $M^\# < \mathcal{J} M$.*

Corollary

*Every word $w \in A^+$ has a decomposition tree whose height is less than $3 \cdot |\mathcal{M}_A| \cdot J(\mathcal{A})$.***
Bounding the acceptance probability from below

Lemma

There exists a positive rational number \( \eta \) which depends only on \( A \) such that: for all words \( w \in A^+ \), there exists \( M \) in \( \mathcal{M}_A \) satisfying for all states \( s, t \in Q \),

\[
M(s, t) = 1 \Rightarrow \mathbb{P}_A(s \xrightarrow{w} t) \geq \eta.
\]
Bounding the acceptance probability from below

Lemma

*There exists a positive rational number* \( \eta \) *which depends only on* \( A \) *such that: for all words* \( w \in A^+ \), *there exists* \( M \) *in* \( M_A \) *satisfying for all states* \( s, t \in Q \),

\[
M(s, t) = 1 \Rightarrow \mathbb{P}_A(s \xrightarrow{w} t) \geq \eta.
\]

*Proof idea*: given \( w \), consider a decomposition tree of bounded height, and prove by induction that the lower bound \( 2^{-h+1} \) holds at depth \( h \), going from leaves to the root.
The case of an iteration node (1)

The node is labelled by \((w_1 \ldots w_n, \langle u^\# \rangle)\) and its children are labelled by \((w_1, \langle u \rangle), \ldots, (w_n, \langle u \rangle)\), where \(\langle u \rangle\) is idempotent, and \(\eta\) a lower bound shared by the \(n \geq 3\) children.
The case of an iteration node (1)

The node is labelled by \((w_1 \ldots w_n, \langle u^\# \rangle)\) and its children are labelled by \((w_1, \langle u \rangle), \ldots, (w_n, \langle u \rangle)\), where \(\langle u \rangle\) is idempotent, and \(\eta\) a lower bound shared by the \(n \geq 3\) children.

Let \(s, t\) such that \(\langle u^\# \rangle(s, t) = 1\), then:

\[
P_A(s \xrightarrow{w_1 \ldots w_n} t) \geq P_A(s \xrightarrow{w_1} t) \cdot P_A(t \xrightarrow{w_2 \ldots w_n} t) \geq \eta^2.
\]
The case of an iteration node (1)

The node is labelled by $(w_1 \ldots w_n, \langle u \rangle)$ and its children are labelled by $(w_1, \langle u \rangle), \ldots, (w_n, \langle u \rangle)$, where $\langle u \rangle$ is idempotent, and $\eta$ a lower bound shared by the $n \geq 3$ children.

Let $s, t$ such that $\langle u \rangle(s, t) = 1$, then:

$$\mathbb{P}_A(s \xrightarrow{w_1 \ldots w_n} t) \geq \mathbb{P}_A(s \xrightarrow{w_1} t) \cdot \mathbb{P}_A(t \xrightarrow{w_2 \ldots w_n} t) \geq \eta^2.$$

The left inequality follows from induction hypothesis, since $\langle u \rangle(s, t) = 1$. 

The case of an iteration node (2)

Consider the right inequality: $\mathbb{P}_A(t \xrightarrow{w_2 \ldots w_n} t) \geq \eta$

Let $C = \{q \mid \langle u \rangle(t, q) \neq 0\}$, we have:

$$\mathbb{P}_A(t \xrightarrow{w_2 \ldots w_n} t) = \sum_{q \in C} \mathbb{P}_A(t \xrightarrow{w_2 \ldots w_{n-1}} q) \cdot \mathbb{P}_A(q \xrightarrow{w_n} t) \geq \eta \geq \eta \cdot \sum_{q \in C} \mathbb{P}_A(t \xrightarrow{w_2 \ldots w_{n-1}} q) = \eta$$

Indeed, since $t$ is recurrent and thanks to the leaktight assumption, we have $C \subseteq \{q \mid \langle u \rangle(q, t) = 1\}$, so the inequality follows from induction hypothesis, and the equality from the “control lemma”.
What I didn’t (and won’t) say

- One can decide whether an automaton is leaktight in PSPACE,
- The value 1 problem for probabilistic leaktight automata is PSPACE-complete,
- The class of leaktight automata subsumes all subclasses of probabilistic automata whose value 1 problem is known to be decidable,
- The class of leaktight automata is closed under parallel composition and synchronized product.
The end.

Thanks for your attention!