Deciding the value 1 problem for probabilistic leaktight automata [1]

Nathanaël Fijalkow,
joint work with Hugo Gimbert and Youssouf Oualhadj

LIAFA, Université Paris 7, France,
University of Warsaw, Poland,
École Normale Supérieure de Cachan, France.

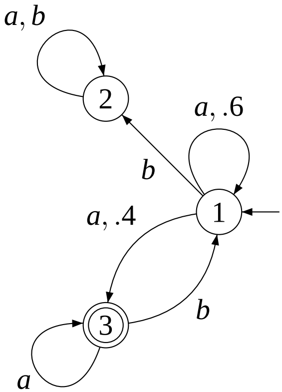
WATA, Dresden, Germany
May 31st 2012

[1] To appear in LICS 2012

Outline

Outline

Probabilistic automata (Rabin, 1963)



$$\mathbb{P}_{\mathcal{A}} : A^* \to [0, 1]$$

The value 1 problem

The value 1 problem is, given a probabilistic automaton $\mathcal{A}$:

"are there words accepted by $\mathcal{A}$ with arbitrarily high probability?"

The value 1 problem

The value 1 problem is, given a probabilistic automaton $\mathcal{A}$:

"are there words accepted by $\mathcal{A}$ with arbitrarily high probability?"

Define $\mathrm{val}(\mathcal{A}) = \sup_w \mathbb{P}_{\mathcal{A}}(w)$. An equivalent formulation of this problem is:

$$\text{“}\mathrm{val}(\mathcal{A}) \stackrel{?}{=} 1\text{”}.$$

The value 1 problem

The value 1 problem is, given a probabilistic automaton $\mathcal{A}$:

"are there words accepted by $\mathcal{A}$ with arbitrarily high probability?"

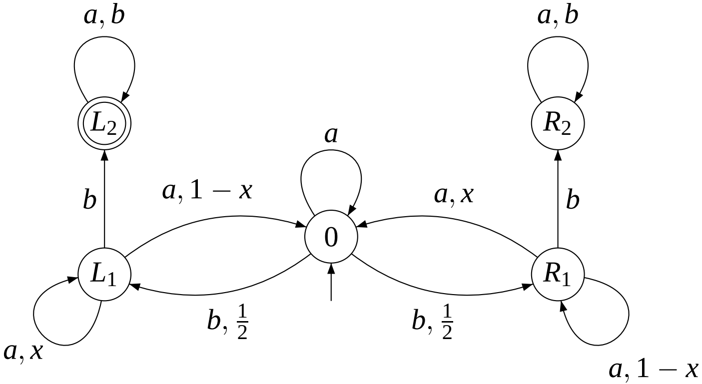
Define $\mathrm{val}(\mathcal{A}) = \sup_w \mathbb{P}_{\mathcal{A}}(w)$. An equivalent formulation of this problem is:

$$\text{“}\mathrm{val}(\mathcal{A}) \stackrel{?}{=} 1\text{”}.$$

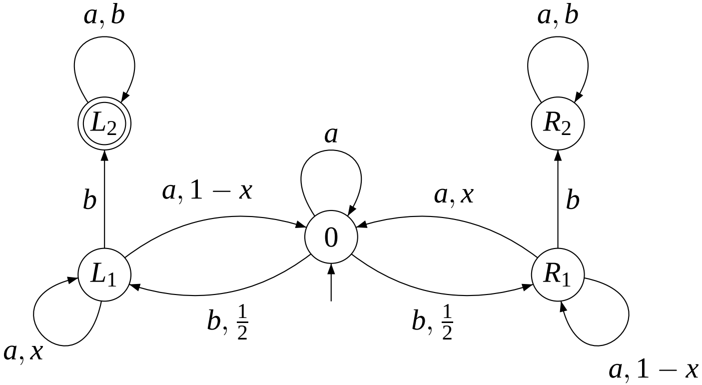
Theorem (Gimbert, Oualhadj, 2010)

The value 1 *problem is undecidable.*

An intuition



An intuition

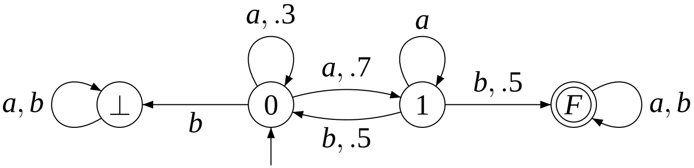


has value 1 if and only if $x > \frac{1}{2}$.

Outline

1. The value 1 problem for probabilistic automata
2. Towards an algebraic treatment of probabilistic automata

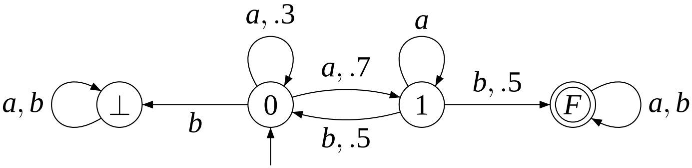
Weighted automata using algebra (Schützenberger)



$$\langle a \rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & .3 & .7 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \langle b \rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & .5 & .5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$I = \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix} \qquad F = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Weighted automata using algebra (Schützenberger)



$$\langle a \rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & .3 & .7 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \langle b \rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & .5 & .5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$I \cdot \langle aaabaa \rangle \cdot F = \mathbb{P}_{\mathcal{A}}(aaabaa)$$

Our objective

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

Our objective

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

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.

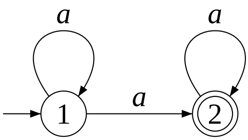
Hence we consider non-deterministic automata: we project $(\mathbb{R}, +, \times)$ into the boolean semiring $(\{0,1\}, +, \times)$.

Our objective

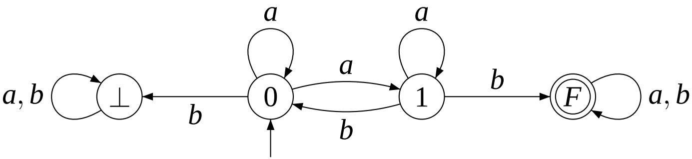
Decide the value 1 problem for a *subclass* of probabilistic automata, by **algebraic** and **non-numerical** means.

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

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



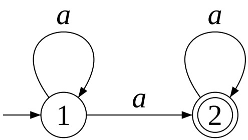
Simplifying



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

$$I \cdot \langle u\rangle \cdot F = 1 \quad \text{if and only if} \quad \mathbb{P}_{\mathcal{A}}(u) > 0$$

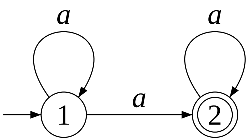
The $\sharp$-expressions from Leung's algorithm



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

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

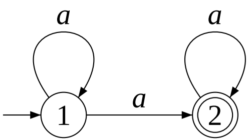
The $\sharp$-expressions from Leung's algorithm



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

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

The $\sharp$-expressions from Leung's algorithm



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

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

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

The algorithm

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

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

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

- Close under product and stabilization.

The algorithm

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

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

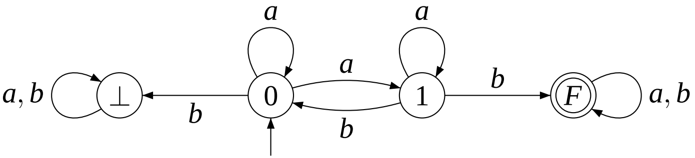
$$\langle a \rangle(s,t) = \begin{cases} 1 & \text{if } \mathbb{P}_{\mathcal{A}}(s \xrightarrow{a} t) > 0, \\ 0 & \text{otherwise.} \end{cases}$$

- Close under product and stabilization.
- If there exists a matrix M such that

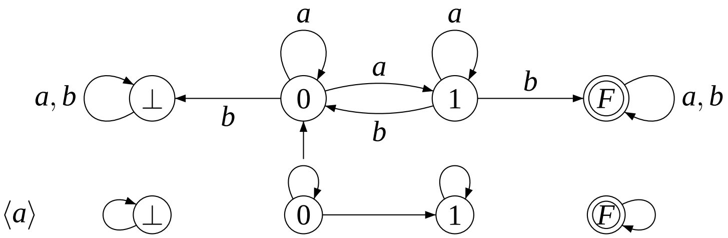
$$\forall t \in Q, \quad M(s_0,t) = 1 \Rightarrow t \in F$$

then "$\mathcal{A}$ has value 1", otherwise "$\mathcal{A}$ does not have value 1".

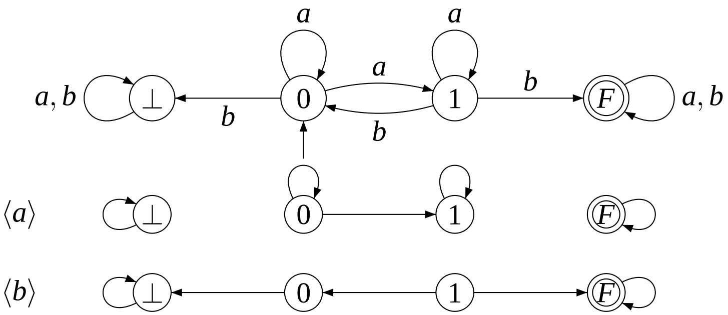
An example



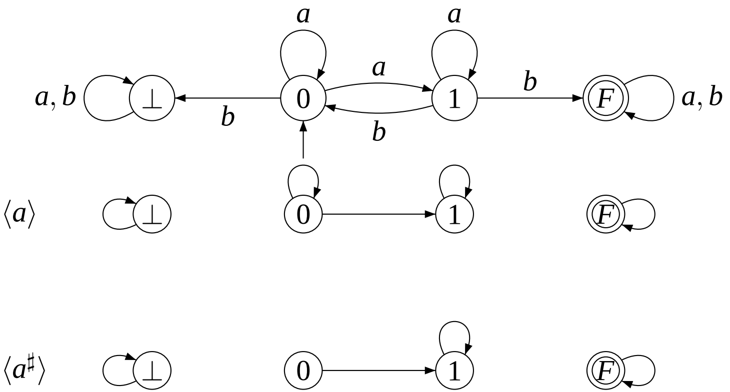
An example



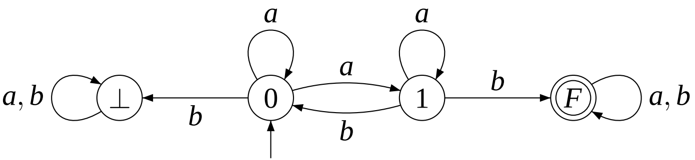
An example



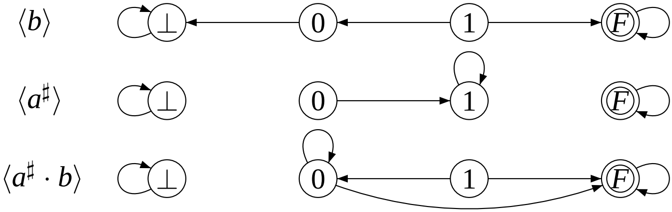
An example



An example

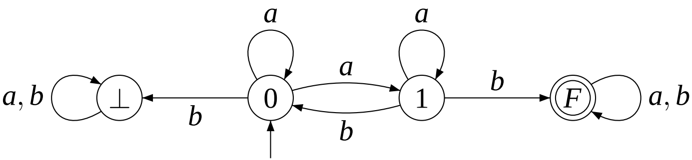


a
a
a
b
a, b
⊥
0
1
F
a, b
b
b

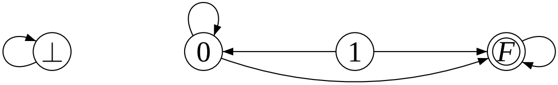


$\langle b \rangle$
$\langle a^\sharp \rangle$
$\langle a^\sharp \cdot b \rangle$
⊥
0
1
F

An example

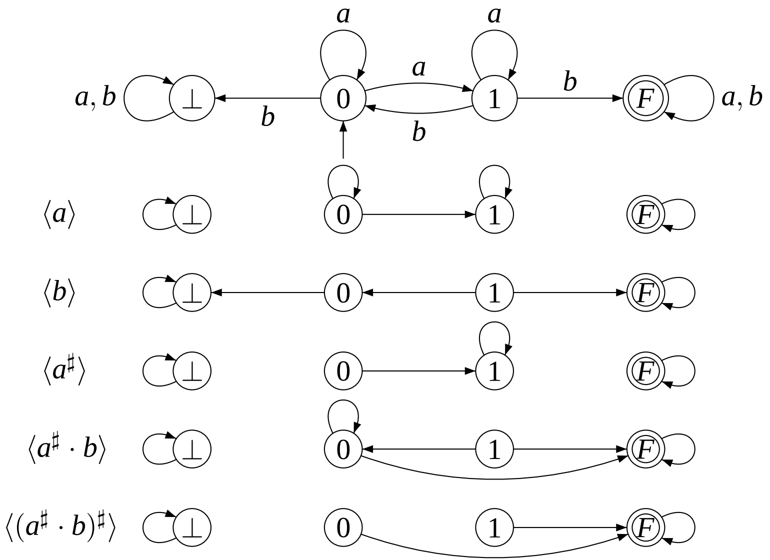


$\langle a^\sharp \cdot b \rangle$



$\langle (a^\sharp \cdot b)^\sharp \rangle$

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 $\mathcal{A}$ has value 1.

Correctness

Theorem

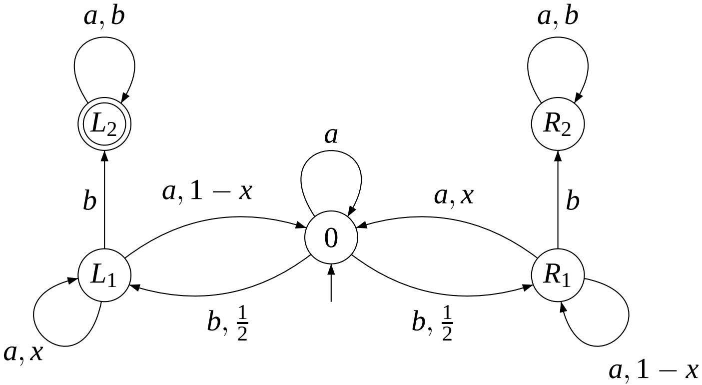
If there exists a matrix M such that

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

then $\mathcal{A}$ has value 1.

But the value 1 problem is undecidable, so...

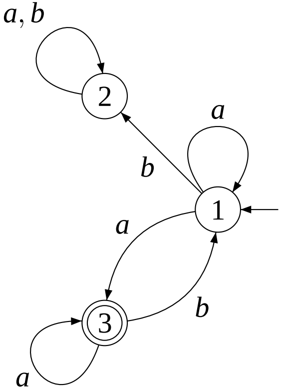
No completeness



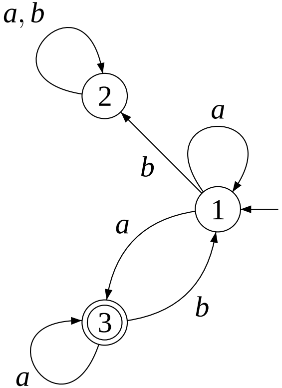
Left and right parts are symmetric, so for all M:

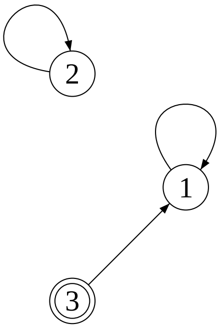
$$M(0, L_2) = 1 \iff M(0, R_2) = 1.$$

A leak



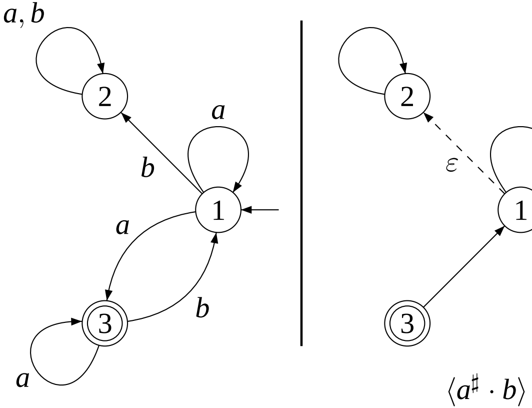
A leak





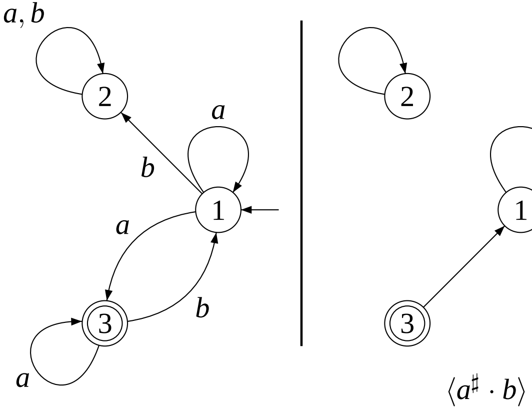
$\langle a^\sharp \cdot b \rangle$

A leak



There is a leak from 1 to 2.

A leak



There is a leak from 1 to 2.

Definition

An automaton $\mathcal{A}$ is leaktight if it has no leak.

Leaktight automata

Theorem (Fijalkow,Gimbert,Oualhadj)

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

The proof relies on Simon's factorization forest theorem.

Conclusion and perspectives

- We defined a subclass of probabilistic automata which subsumes all subclasses of probabilistic automata whose value 1 problem is known to be decidable,

Conclusion and perspectives

- We defined a subclass of probabilistic automata which subsumes all subclasses of probabilistic automata whose value 1 problem is known to be decidable,
- We defined an algebraic algorithm (similar to Leung's algorithm) for the value 1 problem and proved its completeness for the class of leaktight automata.

Conclusion and perspectives

- We defined a subclass of probabilistic automata which subsumes all subclasses of probabilistic automata whose value 1 problem is known to be decidable,

- We defined an algebraic algorithm (similar to Leung's algorithm) for the value 1 problem and proved its completeness for the class of leaktight automata.

- What does this algorithm actually compute?

Conclusion and perspectives

- We defined a subclass of probabilistic automata which subsumes all subclasses of probabilistic automata whose value 1 problem is known to be decidable,

- We defined an algebraic algorithm (similar to Leung's algorithm) for the value 1 problem and proved its completeness for the class of leaktight automata.

- What does this algorithm actually compute?

- Can we use similar algorithms for other semirings?