The Value 1 Problem
for Probabilistic Automata

Nathanaël Fijalkow

LIAFA, Université Denis Diderot - Paris 7, France
Institute of Informatics, Warsaw University, Poland
nath@liafa.univ-paris-diderot.fr

September 4th, 2014
A Real-life Situation

Drive: 0.4

Drive: 0.6

Drive: 0.45

Drive: 0.55

Exit

Exit

Exit

Exit

Confused, Unsure, Perplexed, Disoriented, Bewildered

Home, Next Exit
A Real-life Situation

- No sequence of actions ensure to reach home *almost surely*.
- For every $\varepsilon > 0$, there exists a sequence of actions ensuring to reach home with probability at least $1 - \varepsilon$!
- This is not true anymore if the probabilities change!
The Value 1 Problem

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

\[ \mathbb{P}_A(w) \] is the probability that a run for \( w \) is successful.

**INPUT:** \( A \) a probabilistic automaton

**OUTPUT:** for all \( \varepsilon > 0 \), there exists \( w \in A^* \), \( \mathbb{P}_A(w) \geq 1 - \varepsilon \).

In other words, define \( \text{val}(A) = \sup_{w \in A^*} \mathbb{P}_A(w) \), is \( \text{val}(A) = 1 \)?
A Research Program

Starting point:

**Theorem (Gimbert and Oualhadj, 2010)**

*The value 1 problem is undecidable.*

**But to what extent?**
A Research Program

Starting point:

Theorem (Gimbert and Oualhadj, 2010)

The value 1 problem is undecidable.

But to what extent?

Construct an algorithm to decide the value 1 problem, which is often correct.
A Research Program

Starting point:

Theorem (Gimbert and Oualhadj, 2010)

The value 1 problem is undecidable.

But to what extent?

Construct an algorithm to decide the value 1 problem, which is often correct.

Quantify how often.
A Research Program

Starting point:

Theorem (Gimbert and Oualhadj, 2010)

*The value 1 problem is undecidable.*

But *to what extent?*

Construct an algorithm to decide the value 1 problem, which is *often* correct.

Quantify *how often.*

Argue that you cannot do *more often* than that.
What was known?

Theorem ([BBG12, CSV13])

The value 1 problem is $\Sigma_2^0$-complete.
Our Contributions

- leaktight
  - [FGO12]
- simple
  - [CT12]
- structurally simple
  - [CT12]
- deterministic
  - [GO10]
In [FGO12], we introduced the Markov Monoid, generalizing the transition monoid.

Theorem ([FGO12])

The value 1 problem is decidable for leaktight automata.

Theorem ([FGKO14])

Leaktight automata strictly contain the simple automata.

Theorem ([Fij14])

The Markov Monoid algorithm is optimal.
The following are equivalent:

- The value 1 problem over finite words,
- The emptiness problem over prostochastic words.
The following are equivalent:

- The value 1 problem over finite words,
- The emptiness problem over prostochastic words.

**Theorem ([Fij14])**

1. *The Markov Monoid Algorithm answers “YES” if and only if there exists a regular \( \omega \)-term accepted by \( A \),*
2. *The following problem is undecidable: determine whether there exists an \( \omega \)-term on the level 2 accepted by \( A \).*
Conclusion

We introduced the Markov Monoid Algorithm to solve the value 1 problem for leaktight automata [FGO12].
Conclusion

We introduced the Markov Monoid Algorithm to solve the value 1 problem for leaktight automata [FGO12].

This algorithm is so far, the most correct algorithm to solve the value 1 problem [FGKO14].
Conclusion

We introduced the Markov Monoid Algorithm to solve the value 1 problem for leaktight automata [FGO12].

This algorithm is so far, the most correct algorithm to solve the value 1 problem [FGKO14].

In some sense, this algorithm is optimal [Fij14].
Conclusion

We introduced the Markov Monoid Algorithm to solve the value 1 problem for leaktight automata [FGO12].

This algorithm is so far, the most correct algorithm to solve the value 1 problem [FGKO14].

In some sense, this algorithm is optimal [Fij14].

Thank you!


Deciding the value 1 problem for probabilistic leaktight automata.

Nathanaël Fijalkow.
On the optimality of the markov monoid algorithm.

Hugo Gimbert and Youssouf Oualhadj.
Probabilistic automata on finite words: Decidable and undecidable problems.