Stamina: Stabilisation Monoids in Automata theory

Nathanaël Fijalkow[1] Hugo Gimbert[2] Edon Kelmendi[2]
Denis Kuperberg[3]

[1]Turing Institute, London

[2]LABRI, Bordeaux

[3]ENS Lyon

CIAA, Marne-la-Vallée
28-06-2017

Quantitative Automata

Kinds of quantitative automata:

- Weighted automata [Schützenberger 1961]
- Probabilistic automata [Rabin 1963]
- Timed automata [Alur, Dill 1994]
- Cost automata [Colcombet 2009]
- ...

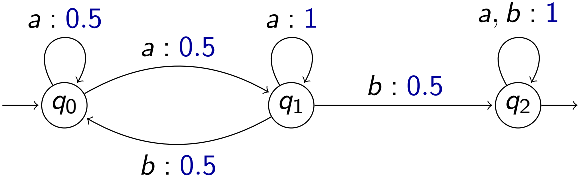
Quantitative Automata

Kinds of quantitative automata:

- Weighted automata [Schützenberger 1961]
- **Probabilistic automata** [Rabin 1963]
- Timed automata [Alur, Dill 1994]
- **Cost automata** [Colcombet 2009]
- ...

This tool: Analyse asymptotic behaviours of quantitative automata.

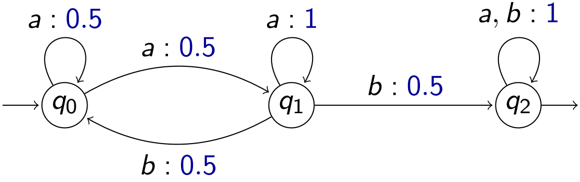
Probabilistic Automata



Probabilistic automaton: computes function $P_{\mathcal{A}} : A^* \to [0, 1]$, mapping words to probability of acceptance.

Example: $P_{\mathcal{A}}(ab) = 0.25$

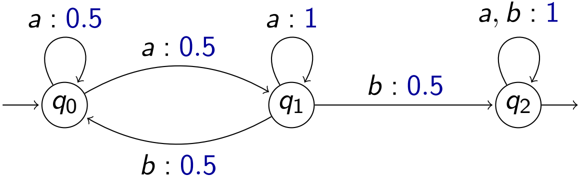
Value 1 Problem



Value 1 problem: is there a sequence of words $(u_n)_{n\in\mathbb{N}}$ such that

$$\lim_{n\to\infty} P_{\mathcal{A}}(u_n) = 1 \ ?$$

Value 1 Problem

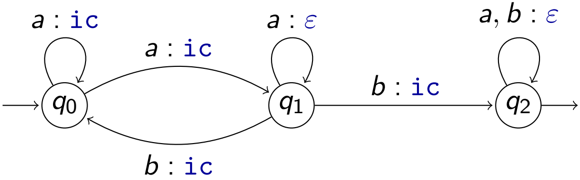


Value 1 problem: is there a sequence of words $(u_n)_{n\in\mathbb{N}}$ such that

$$\lim_{n\to\infty} P_{\mathcal{A}}(u_n) = 1\ ?$$

Yes: $P_{\mathcal{A}}((a^n b)^n) \xrightarrow{n\to\infty} 1.$

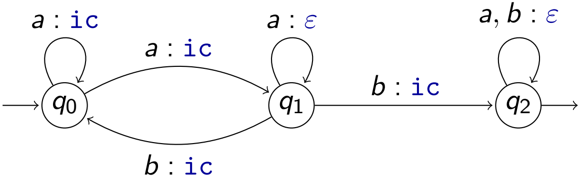
Cost automata



Cost automaton: Non-deterministic automaton equipped with finitely many counters

- Operations on counters: increment (ic), reset (r), wait (ε).

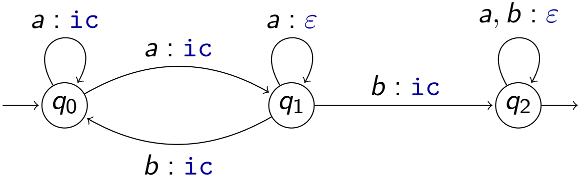
Cost automata



Cost automaton: Non-deterministic automaton equipped with finitely many counters

- Operations on counters: increment (ic), reset (r), wait (ε).
- Value of a run: highest value taken by a counter.

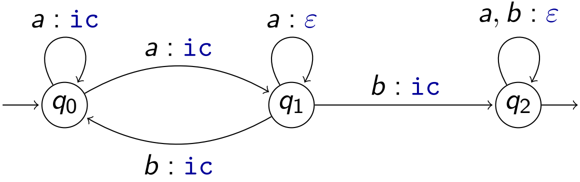
Cost automata



Cost automaton: Non-deterministic automaton equipped with finitely many counters

- Operations on counters: increment (ic), reset (r), wait (ε).
- Value of a run: highest value taken by a counter.
- Semantics $[\![\mathcal{A}]\!] : L(\mathcal{A}) \to \mathbb{N}$, mapping to each accepted word u the minimal value of a run on u. **Example:** $[\![\mathcal{A}]\!](ab) = 2$.

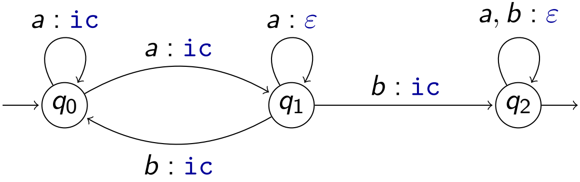
Limitedness Problem



Limitedness problem: is there a sequence of words $(u_n)_{n \in \mathbb{N}}$ such that

$$\lim_{n \to \infty} [\![\mathcal{A}]\!](u_n) = \infty \; ?$$

Limitedness Problem



Limitedness problem: is there a sequence of words $(u_n)_{n\in\mathbb{N}}$ such that

$$\lim_{n\to\infty} [\![\mathcal{A}]\!](u_n) = \infty \ ?$$

No. For all $u \in L(\mathcal{A})$, we have $[\![\mathcal{A}]\!](u) = 2$.

Stabilisation Monoids

Common tool for Value 1 problem and Limitedness problem: Stabilisation monoids.

Monoid with unary "stabilisation" operation ($\sharp$) describing asymptotic behaviour.

Stabilisation Monoids

Common tool for Value 1 problem and Limitedness problem: Stabilisation monoids.

Monoid with unary "stabilisation" operation ($\sharp$) describing asymptotic behaviour.

Used for witness searching among sequences of words.

Stabilisation Monoids

Common tool for Value 1 problem and Limitedness problem: Stabilisation monoids.

Monoid with unary "stabilisation" operation ($\sharp$) describing asymptotic behaviour.

Used for witness searching among sequences of words.

- Complete for Limitedness problem

Stabilisation Monoids

Common tool for Value 1 problem and Limitedness problem: Stabilisation monoids.

Monoid with unary "stabilisation" operation ($\sharp$) describing asymptotic behaviour.

Used for witness searching among sequences of words.

- Complete for Limitedness problem
- Incomplete for Value 1 problem, undecidable in general, complete for leaktight automata [F., G., Oualhadj 2012].

Stabilisation Monoids

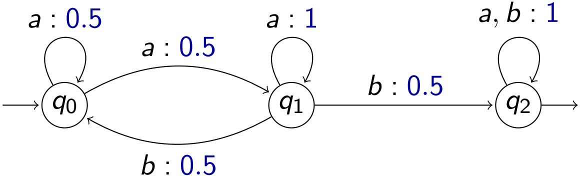
Common tool for Value 1 problem and Limitedness problem: Stabilisation monoids.

Monoid with unary "stabilisation" operation ($\sharp$) describing asymptotic behaviour.

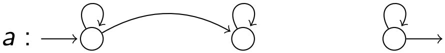
Used for witness searching among sequences of words.

- Complete for Limitedness problem
- Incomplete for Value 1 problem, undecidable in general, complete for leaktight automata [F., G., Oualhadj 2012].
- "Most correct" algorithm for Value 1 problem [Fijalkow 2016]

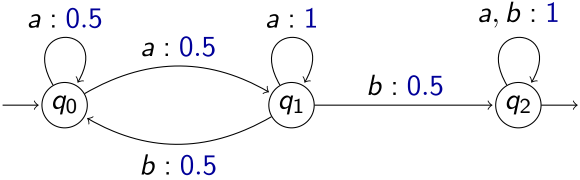
Example



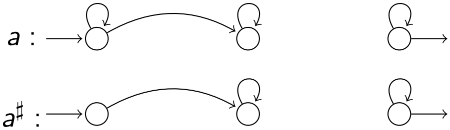
Elements of the stabilisation monoid:



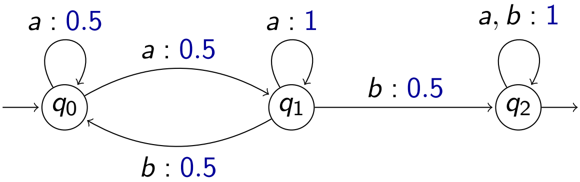
Example



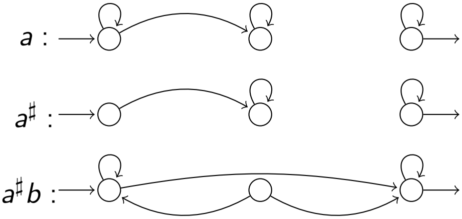
Elements of the stabilisation monoid:



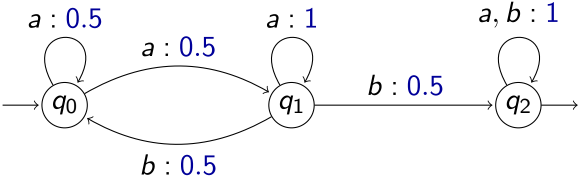
Example



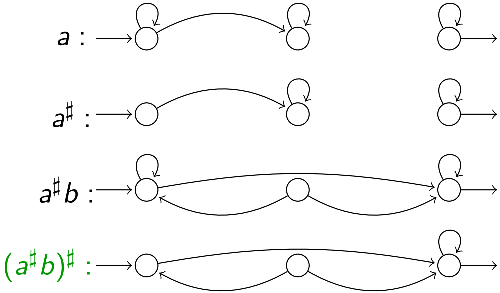
Elements of the stabilisation monoid:



Example



Elements of the stabilisation monoid:

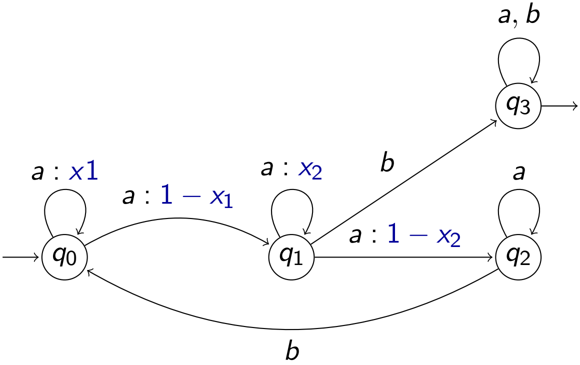


A non-leaktight example

The algorithm does not look at precise values of the labels.
In non-leaktight automata, these values can matter.

A non-leaktight example

The algorithm does not look at precise values of the labels.
In non-leaktight automata, these values can matter.



Idea of the algorithm

- Start with matrices of letters,
- Saturation of the stabilisation monoid by product and stabilisation,
- Stop if witness is found, answer **Value 1/Unlimited**.
- If no witness:
  - **Limited** if cost automaton.
  - **[Probably] not value 1** (depending on leaktightness).

Idea of the algorithm

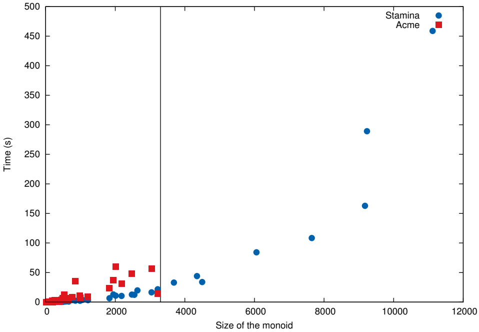
- Start with matrices of letters,
- Saturation of the stabilisation monoid by product and stabilisation,
- Stop if witness is found, answer **Value 1/Unlimited**.
- If no witness:
  - **Limited** if cost automaton.
  - **[Probably] not value 1** (depending on leaktightness).

Optimizations

- basic CPU operations on vectors
- shared vectors between matrices
- rewrite rules like $A^\sharp B \rightarrow AB$ to avoid computations

Performances

ACME [F.,K. 2014]: previous version in OCaml for Value 1 and Limitedness problems.
Comparison on random automata of size 10, according to monoid size.



Star-height problem

L defined by $(a^*b)^*(a^*c)^*$

Question: Can we represent L with fewer nestings of Kleene stars?

Star-height problem

$$L \text{ defined by } (a^* b)^*(a^* c)^*$$

Question: Can we represent L with fewer nestings of Kleene stars?
Yes: $(a+b)^*(a+c)^*$.

- Decidability open from 1963 [Eggan] to 1988 [Hashiguchi]

Star-height problem

$$L \text{ defined by } (a^*b)^*(a^*c)^*$$

Question: Can we represent L with fewer nestings of Kleene stars?
Yes: $(a+b)^*(a+c)^*$.

- Decidability open from 1963 [Eggan] to 1988 [Hashiguchi]
- Impractical algorithm: on a 4-state automaton, more than c^{c^c} operations with $c = 10^{10^{10}}$ [Lombardy, Sakarovitch 2002].

Star-height problem

$$L \text{ defined by } (a^*b)^*(a^*c)^*$$

Question: Can we represent L with fewer nestings of Kleene stars?
Yes: $(a+b)^*(a+c)^*$.

- Decidability open from 1963 [Eggan] to 1988 [Hashiguchi]
- Impractical algorithm: on a 4-state automaton, more than c^{c^c} operations with $c = 10^{10^{10}}$ [Lombardy, Sakarovitch 2002].
- Improved to **2EXPSPACE** in [Kirsten 2005], by reduction to boundedness of automata with counters

Star-height problem

L defined by $(a^* b)^* (a^* c)^*$

Question: Can we represent L with fewer nestings of Kleene stars?
Yes: $(a + b)^* (a + c)^*$.

- Decidability open from 1963 [Eggan] to 1988 [Hashiguchi]
- Impractical algorithm: on a 4-state automaton, more than c^{c^c} operations with $c = 10^{10^{10}}$ [Lombardy, Sakarovitch 2002].
- Improved to **2EXPSPACE** in [Kirsten 2005], by reduction to boundedness of automata with counters
- Embedded in cost functions theory [Colcombet 2009]

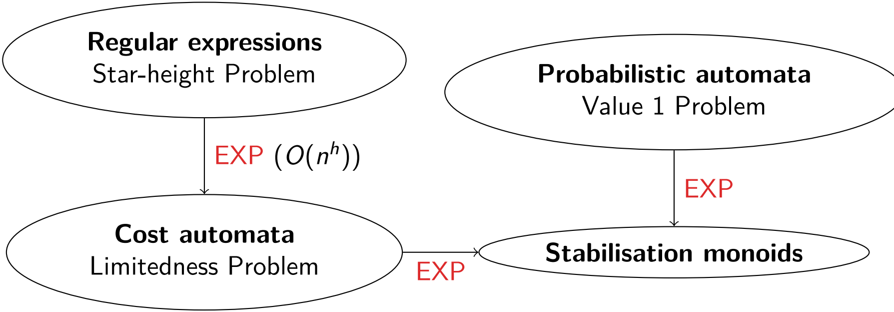
Star-height problem

L defined by $(a^*b)^*(a^*c)^*$

Question: Can we represent L with fewer nestings of Kleene stars?
Yes: $(a+b)^*(a+c)^*$.

- Decidability open from 1963 [Eggan] to 1988 [Hashiguchi]
- Impractical algorithm: on a 4-state automaton, more than c^{c^c} operations with $c = 10^{10^{10}}$ [Lombardy, Sakarovitch 2002].
- Improved to **2EXPSPACE** in [Kirsten 2005], by reduction to boundedness of automata with counters
- Embedded in cost functions theory [Colcombet 2009]
- Implemented here for the first time.

Big picture


Regular expressions
Star-height Problem
EXP ($O(n^h)$)
Probabilistic automata
Value 1 Problem
EXP
Cost automata
Limitedness Problem
EXP
Stabilisation monoids

More on the star-height algorithm

Loop Complexity (LC) of a NFA: minimal star-height of an expression derived from this NFA.
From [Eggan 1963]:

$$\text{Star-Height}(L) = \min\{LC(\mathcal{A}) \mid \mathcal{A} \text{ NFA for } L\}$$

More on the star-height algorithm

Loop Complexity (LC) of a NFA: minimal star-height of an expression derived from this NFA.
From [Eggan 1963]:

$$\text{Star-Height}(L) = \min\{LC(\mathcal{A}) \mid \mathcal{A} \text{ NFA for } L\}$$

Heuristics for Star-height problem:

- Loop complexity of input automaton
  - computable in **PTIME**
  - gives a regular expression e of low star-height
  - optimal in many cases [Cohen 1970]
- Test probable unlimited witnesses based on e
- Minimisation of an intermediary automaton [Colcombet, Löding 2008].