Bounded-Regret Determinization of Max-Plus Automata

Guillermo A. Pérez

Université libre de Bruxelles
University of Oxford

Highlights'17 @ London

Max-plus automata

(Weighted) automata

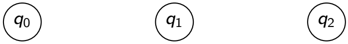
A (non-deterministic weighted finite) automaton $\mathcal{A} = (Q, I, A, \Delta, w, F)$

Max-plus automata

(Weighted) automata

A (non-deterministic weighted finite) automaton $\mathcal{A} = (Q, I, A, \Delta, w, F)$

- states Q,

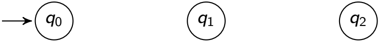


Max-plus automata

(Weighted) automata

A (non-deterministic weighted finite) automaton $\mathcal{A} = (Q, I, A, \Delta, w, F)$

- states Q, initial states I,

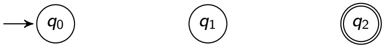


Max-plus automata

(Weighted) automata

A (non-deterministic weighted finite) automaton $\mathcal{A} = (Q, I, A, \Delta, w, F)$

- states Q, initial states I, final states F

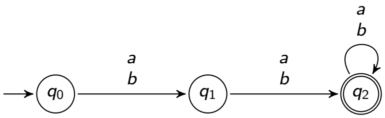


Max-plus automata

(Weighted) automata

A (non-deterministic weighted finite) automaton $\mathcal{A} = (Q, I, A, \Delta, w, F)$

- states Q, initial states I, final states F
- an alphabet $A = \{a, b, \dots\}$, and transition relation $\Delta \subseteq Q \times A \times Q$

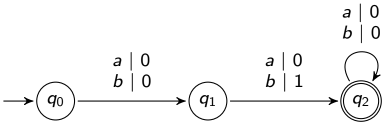


Max-plus automata

(Weighted) automata

A (non-deterministic weighted finite) automaton $\mathcal{A} = (Q, I, A, \Delta, w, F)$

- states Q, initial states I, final states F
- an alphabet $A = \{a, b, \dots\}$, and transition relation $\Delta \subseteq Q \times A \times Q$
- a weight function $w : \Delta \to \mathbb{Z}$

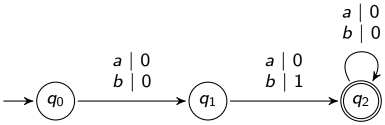


Max-plus automata

(Weighted) automata

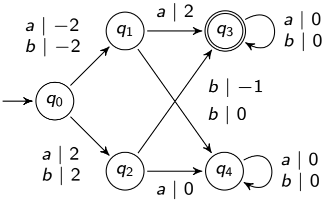
A (non-deterministic weighted finite) automaton $\mathcal{A} = (Q, I, A, \Delta, w, F)$

- states Q, initial states I, final states F
- an alphabet $A = \{a, b, \dots\}$, and transition relation $\Delta \subseteq Q \times A \times Q$
- a weight function $w : \Delta \to \mathbb{Z}$



it is deterministic if 'Δ is functional' and I is a singleton $\{q_I\}$

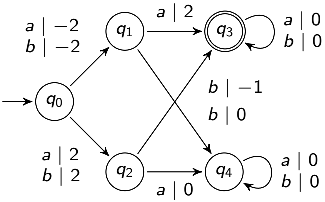
Max-plus automata



Runs, acceptance, and value

A run (on input word) is a state-letter sequence, e.g. $q_0 b q_1 a q_3$; it can be initial, final, or accepting; it has a value e.g.
$w(q_0 b q_1 a q_3) = w(q_0, b, q_1) + w(q_1, a, q_3)$.

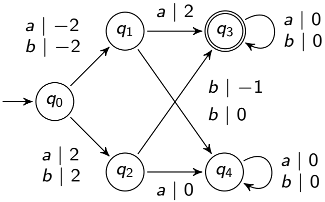
Max-plus automata



Language, function, maximal runs

- A (finite) word α is accepted if the automaton has an accepting run on α; the set $\mathcal{L}_{\mathcal{A}}$ of all such words is the language of the automaton.

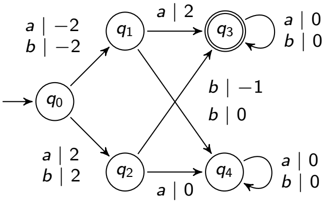
Max-plus automata



Language, function, maximal runs

- A (finite) word α is accepted if the automaton has an accepting run on α; the set $\mathcal{L}_{\mathcal{A}}$ of all such words is the language of the automaton.
- The automaton induces a function $[\![\mathcal{A}]\!] : \mathcal{L}_{\mathcal{A}} \to \mathbb{Z}$ by mapping words α to the maximal value of an accepting run on α.

Max-plus automata



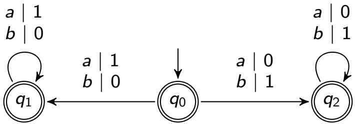
Language, function, maximal runs

- A (finite) word α is accepted if the automaton has an accepting run on α; the set $\mathcal{L}_{\mathcal{A}}$ of all such words is the language of the automaton.
- The automaton induces a function $[\![\mathcal{A}]\!] : \mathcal{L}_{\mathcal{A}} \to \mathbb{Z}$ by mapping words α to the maximal value of an accepting run on α.
- An accepting run ρ on α is maximal if $w(\rho) = [\![\mathcal{A}]\!](\alpha)$.

Motivation: determinizability

Proposition

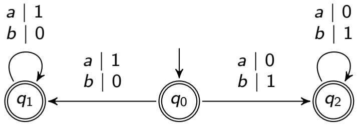
The function $\alpha \mapsto \max(|\alpha|_a, |\alpha|_b)$ *is not realizable by a deterministic weighted automaton.*



Motivation: determinizability

Proposition

The function $\alpha \mapsto \max(|\alpha|_a, |\alpha|_b)$ *is not realizable by a deterministic weighted automaton.*

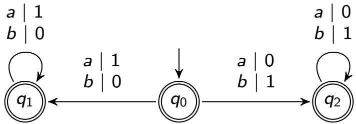


Some interesting problems are not decidable in the presence of non-determinism. . .

Motivation: determinizability

Proposition

The function $\alpha \mapsto \max(|\alpha|_a, |\alpha|_b)$ *is not realizable by a deterministic weighted automaton.*



Some interesting problems are not decidable in the presence of non-determinism...

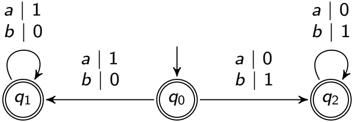
Proposition (Krob 1992; Almagor, Boker, Kupferman 2011)

Determining if $[\![\mathcal{A}]\!] \leq [\![\mathcal{B}]\!]$ *(quantitative language inclusion) is undecidable.*

Motivation: determinizability

Proposition

The function $\alpha \mapsto \max(|\alpha|_a, |\alpha|_b)$ *is not realizable by a deterministic weighted automaton.*



Deciding determinizability is open!

- Sufficient (and decidable) conditions for determinizability are known [Mohri 1997; Aminof, Kupferman, Lampert 2013].
- Determining if a polynomially ambiguous weighted automaton is determinizable is decidable [Kirsten, Lombardy 2009].

Motivation: bounded-regret determinizability

Determinizing on the fly

A weighted automaton $\mathcal{A}$ is determinizable on-the-fly if there is a non-determinism resolving strategy constructing a run of $\mathcal{A}$ as the input word is spelled (letter by letter).

Motivation: bounded-regret determinizability

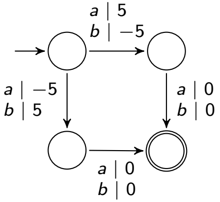
Determinizing on the fly

A weighted automaton $\mathcal{A}$ is determinizable on-the-fly if there is a non-determinism resolving strategy constructing a run of $\mathcal{A}$ as the input word is spelled (letter by letter). (And the same function is obtained.)

Motivation: bounded-regret determinizability

Determinizing on the fly

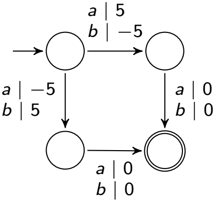
A weighted automaton $\mathcal{A}$ is determinizable on-the-fly if there is a non-determinism resolving strategy constructing a run of $\mathcal{A}$ as the input word is spelled (letter by letter). (And the same function is obtained.)

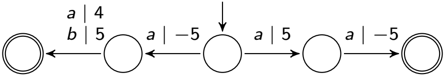


Motivation: bounded-regret determinizability

Determinizing on the fly

A weighted automaton $\mathcal{A}$ is determinizable on-the-fly if there is a non-determinism resolving strategy constructing a run of $\mathcal{A}$ as the input word is spelled (letter by letter). (And the same function is obtained.)





Motivation: bounded-regret determinizability

Determinizing on the fly

A weighted automaton $\mathcal{A}$ is determinizable on-the-fly if there is a non-determinism resolving strategy constructing a run of $\mathcal{A}$ as the input word is spelled (letter by letter). (And the same function is obtained.)

- On-the-fly determinizability is equivalent to the automaton being good for games [Henzinger, Piterman 2006].

Motivation: bounded-regret determinizability

Determinizing on the fly

A weighted automaton $\mathcal{A}$ is determinizable on-the-fly if there is a non-determinism resolving strategy constructing a run of $\mathcal{A}$ as the input word is spelled (letter by letter). (And the same function is obtained.)

- On-the-fly determinizability is equivalent to the automaton being good for games [Henzinger, Piterman 2006].
- If $\mathcal{A}$ is determinizable on the fly, then it is determinizable [Aminof, Kupferman, Lampert 2010].

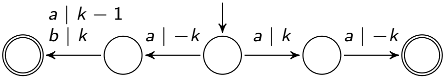
Motivation: bounded-regret determinizability

Determinizing on the fly

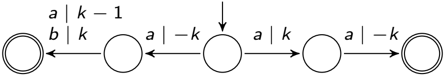
A weighted automaton $\mathcal{A}$ is determinizable on-the-fly if there is a non-determinism resolving strategy constructing a run of $\mathcal{A}$ as the input word is spelled (letter by letter). (And the same function is obtained.)

- On-the-fly determinizability is equivalent to the automaton being good for games [Henzinger, Piterman 2006].
- If $\mathcal{A}$ is determinizable on the fly, then it is determinizable [Aminof, Kupferman, Lampert 2010].
- On-the-fly determinizability is equivalent to the following. There exists a deterministic automaton $\mathcal{D}$ such that $[\![\mathcal{D}]\!] = [\![\mathcal{A}]\!]$ and there is a homomorphism from $\mathcal{D}$ to $\mathcal{A}$ [Colcombet 2012].

Motivation: bounded-regret determinizability



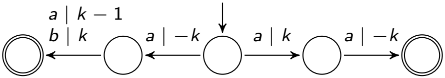
Motivation: bounded-regret determinizability



Approximate determinization on the fly

If a non-zero regret bound r is allowed, the automaton is approximately determinizable on-the-fly.

Motivation: bounded-regret determinizability

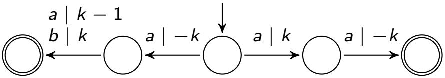


Approximate determinization on the fly

If a non-zero regret bound r is allowed, the automaton is approximately determinizable on-the-fly.

- Can be used to formalize the notion of competitive online algorithms [Aminof, Kupferman, Lampert 2010].

Motivation: bounded-regret determinizability



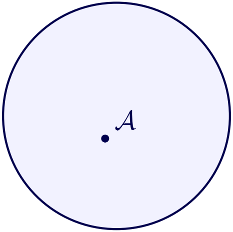
Approximate determinization on the fly

If a non-zero regret bound r is allowed, the automaton is approximately determinizable on-the-fly.

- Can be used to formalize the notion of competitive online algorithms [Aminof, Kupferman, Lampert 2010].
- Quantitative generalization of good-for-games automata [Hunter, P., Raskin 2016].
- (These automata suffice as observers for approximate quantitative games.)

Bounded-regret determinization I

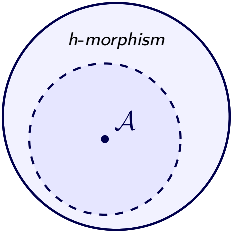
0-regret: stronger notion of determinization



$\{\mathcal{B}$ s.t. $[\![\mathcal{B}]\!] = [\![\mathcal{A}]\!]\}$

Bounded-regret determinization I

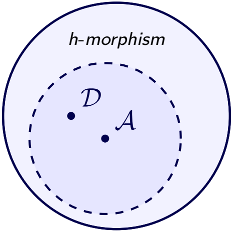
0-regret: stronger notion of determinization



$\{\mathcal{B}$ s.t. $[\![\mathcal{B}]\!] = [\![\mathcal{A}]\!]\}$

Bounded-regret determinization I

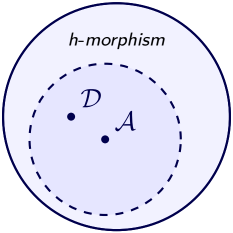
0-regret: stronger notion of determinization



$\{\mathcal{B}$ s.t. $[\![\mathcal{B}]\!] = [\![\mathcal{A}]\!]\}$

Bounded-regret determinization I

0-regret: stronger notion of determinization



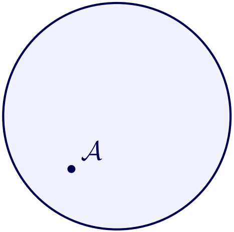
$\{\mathcal{B}$ s.t. $[\![\mathcal{B}]\!] = [\![\mathcal{A}]\!]\}$

Proposition (Aminof, Kupferman, Lampert 2010)

Determining if a given weighted automaton is 0*-regret determinizable is decidable in polynomial time.*

Bounded-regret determinization II

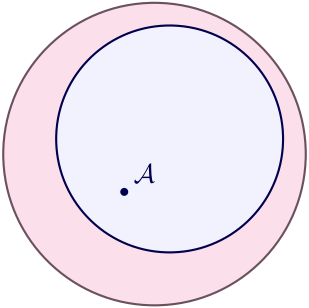
r-regret: approximate determinization under a structural restriction



$\{\mathcal{B} \text{ s.t. } [\![\mathcal{B}]\!] = [\![\mathcal{A}]\!]\}$

Bounded-regret determinization II

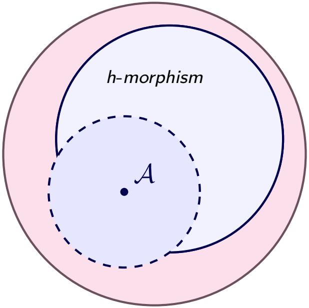
r-regret: approximate determinization under a structural restriction



$\{\mathcal{B} \text{ s.t. } |[\![\mathcal{B}]\!] - [\![\mathcal{A}]\!]| \leq r\}$

Bounded-regret determinization II

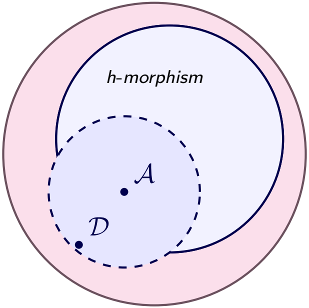
r-regret: approximate determinization under a structural restriction



$\{\mathcal{B} \text{ s.t. } |[\![\mathcal{B}]\!] - [\![\mathcal{A}]\!]| \leq r\}$

Bounded-regret determinization II

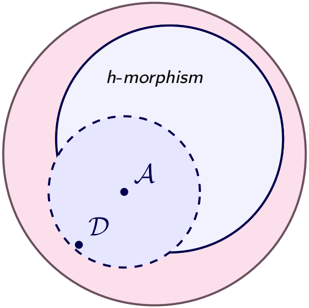
r-regret: approximate determinization under a structural restriction



$\{\mathcal{B}$ s.t. $|[\![\mathcal{B}]\!] - [\![\mathcal{A}]\!]| \leq r\}$

Bounded-regret determinization II

r-regret: approximate determinization under a structural restriction



$\{\mathcal{B} \text{ s.t. } |[\![\mathcal{B}]\!] - [\![\mathcal{A}]\!]| \leq r\}$

Theorem

The r-regret determinization problem is decidable in exponential time (using a quantitative generalization of Joker games [Kuperberg, Skrzypczak 2015]).

Conclusion

	the 0 case	general r
Bounded regret	polynomial time	EXP-complete

- Are the regret and quantitative Joker games equivalent?
- What is the complexity of deciding if $\exists r$ s.t. an automaton is r-regret determinizable?
- What if we define regret w.r.t. ratio instead of difference?

For more details. . .

- Emmanuel Filiot, Ismael Jecker, Nathan Lhote, Guillermo A. Pérez, Jean-François Raskin. **On delay and regret determinization of max-plus automata.** LICS 2017: 1-12.