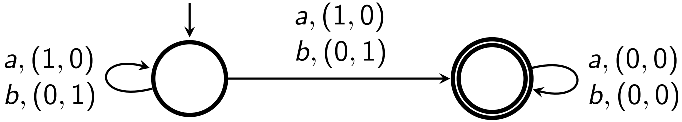
On Parikh Automata: Infinite Words, Games, and History-determinism

Joint work with Shibashis Guha, Ismaël Jecker, and Karoliina Lehtinen

Martin Zimmermann

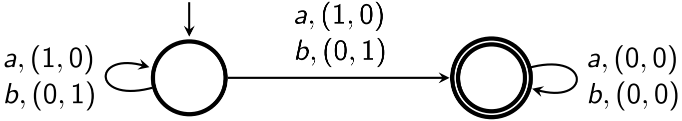
Aalborg University

Parikh Automata



$$C = \{(n, n') \mid n < n'\}$$

Parikh Automata

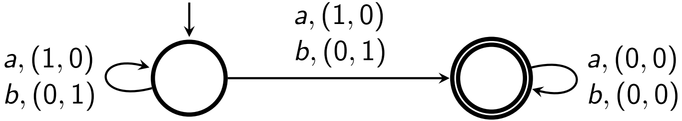


$$C = \{(n, n') \mid n < n'\}$$

Semantics: A run processes a word $w \in \{a, b\}^*$ and yields a vector $\vec{v}$ in $\mathbb{N}^2$ (sum up all vectors on the transitions). It is accepting if

- it ends in an accepting state, and
- $\vec{v} \in C$.

Parikh Automata



$$C = \{(n, n') \mid n < n'\}$$

Semantics: A run processes a word $w \in \{a, b\}^*$ and yields a vector $\vec{v}$ in $\mathbb{N}^2$ (sum up all vectors on the transitions). It is accepting if

- it ends in an accepting state, and
- $\vec{v} \in C$.

$L(\mathcal{A}, C) = \{w \in \{a, b\}^* \mid w$ has a prefix with more b's than a's$\}$

Why Parikh Automata?

- Quantitative
- Equivalent to a quantitative variant of WMSO
- Equivalent to reversal-bounded counter machines
- Emptiness decidable
- "Some" closure properties
- Useful in applications: querying graph databases, model checking transducer properties, etc.

Why Parikh Automata?

- Quantitative
- Equivalent to a quantitative variant of WMSO
- Equivalent to reversal-bounded counter machines
- Emptiness decidable
- "Some" closure properties
- Useful in applications: querying graph databases, model checking transducer properties, etc.

But: So far only considered for finite words!

Here:

- Parikh automata on infinite words
- Games with Parikh automata winning conditions
- History-deterministic Parikh automata

Parikh Automata on Infinite Words

For infinite runs:

- Reachability acceptance: Some run prefix is accepting.
- Safety acceptance: All run prefixes are accepting.
- Büchi acceptance: Infinitely many run prefixes are accepting.
- co-Büchi acceptance: Almost all run prefixes are accepting.

Parikh Automata on Infinite Words

Theorem

Emptiness is decidable for reachability and Büchi Parikh automata.

Parikh Automata on Infinite Words

Theorem

Emptiness is decidable for reachability and Büchi Parikh automata.

But:

Theorem

Emptiness is undecidable for safety and co-Büchi Parikh automata.

Parikh Automata on Infinite Words

Theorem

Emptiness is decidable for reachability and Büchi Parikh automata.

But:

Theorem

Emptiness is undecidable for safety and co-Büchi Parikh automata.

Something is not quite "right"! Solution:

Parikh Automata on Infinite Words

Theorem

Emptiness is decidable for reachability and Büchi Parikh automata.

But:

Theorem

Emptiness is undecidable for safety and co-Büchi Parikh automata.

Something is not quite "right"! Solution:

Theorem

Reachability Parikh automata can be transformed into Büchi and co-Büchi Parikh automata. All other models are pairwise incomparable.

Games with Parikh Automata Winning Conditions

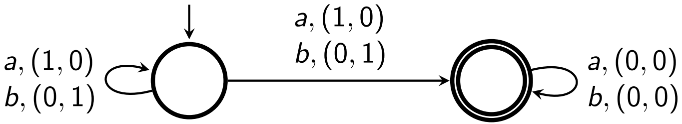
So, model checking finite systems against Büchi Parikh automata is decidable.

What about synthesis?

Theorem

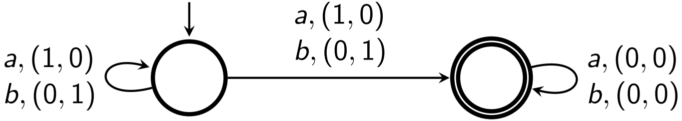
Gale-Stewart games with winning conditions given by deterministic safety Parikh automata are undecidable.

History-determinism



$$C = \{(n, n') \mid n < n'\}$$

History-determinism

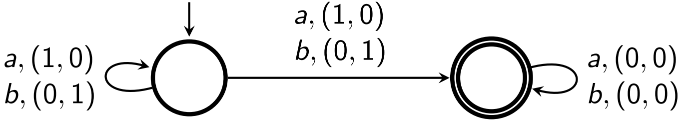


$$C = \{(n, n') \mid n < n'\}$$

The nondeterministic choice can be made based on the prefix processed so far, independently how the word continues.

$\Rightarrow$ History-determinism (a.k.a. good-for-games)

History-determinism



$$C = \{(n, n') \mid n < n'\}$$

Theorem

History-deterministic Parikh automata are more expressive than deterministic ones, but less expressive than nondeterministic ones.

More Results

Preprint will appear on the arXiv soon...

- Closure properties for Parikh automata on infinite words.
- Decision problems for Parikh automata on infinite words.
- Closure properties for history-deterministic Parikh automata on finite words.
- Decision problems for history-deterministic Parikh automata on finite words.
- More expressiveness results.