History-Deterministic Register Automata

Léo Exibard[1] Karoliina Lehtinen[2]

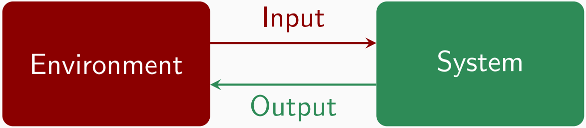
Friday, November 5th, 2021

[1]ICE-TCS
Reykjavik University
Iceland

[2]Laboratoire d'Informatique et des Systèmes
Aix-Marseille Université
France

Motivation: Reactive Synthesis

Reactive systems



Interaction $\rightsquigarrow$ $i_1 o_1 i_2 o_2 i_3 o_3 \ldots$

Goal

Generate a system from a specification

$? \parallel \mathrm{Env} \models \mathrm{Specification}$

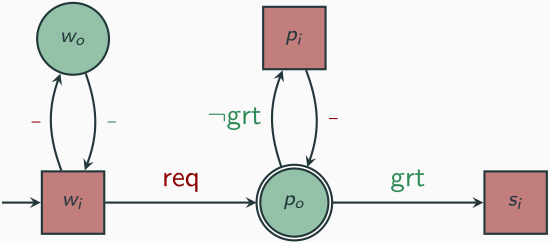
The Finite Alphabet Case

Specification = ω-regular language, as an MSO or LTL formula.

Example

Any request is eventually granted:

$$G(req \Rightarrow F(grt))$$



A non-deterministic Büchi automaton checking that some request is left unsatisfied

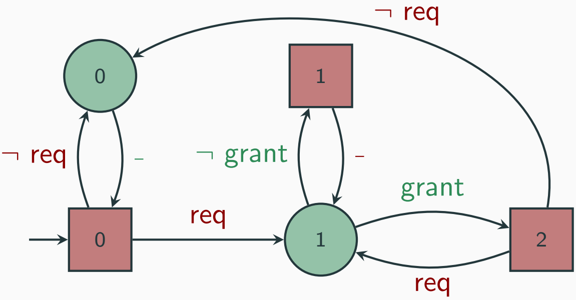
Theorem [J.R. Büchi and L.H. Landweber, 1969]

The reactive synthesis problem is decidable for ω-regular specifications.

Theorem [J.R. Büchi and L.H. Landweber, 1969]

The reactive synthesis problem is decidable for ω-regular specifications.

- Specification $\rightsquigarrow$ equivalent deterministic automaton A
- Resolve an infinite duration game played over A



A parity game corresponding to $G(\text{req} \Rightarrow F(\text{grt}))$.

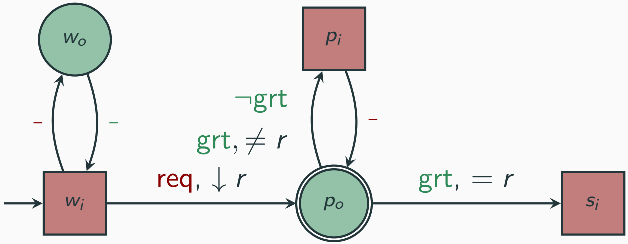
Lifting to Infinite Alphabets: Register Automata [Kaminski and Francez, 1994]

➔ Requests are specifically granted to a client

Register Automata

Finite automata with a finite set R of registers

- Store data
- Test register content



A non-deterministic Büchi *register* automaton checking that some request is left unsatisfied

Lifting to Infinite Alphabets: Register Automata [Kaminski and Francez, 1994]

Theorem [Exibard et al., 2021]

The synthesis problem is decidable for specifications given as deterministic register automata.

Lifting to Infinite Alphabets: Register Automata [Kaminski and Francez, 1994]

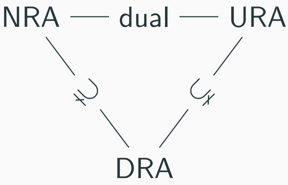
Theorem [Exibard et al., 2021]

The synthesis problem is decidable for specifications given as deterministic register automata.

Problem

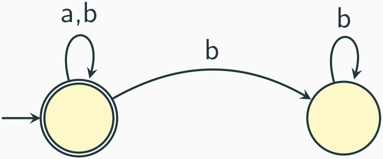
Register automata do not determinise!

➔ “Some two data values are identical”



History-Determinism [Henzinger and Piterman, 2006] [Colcombet, 2009]

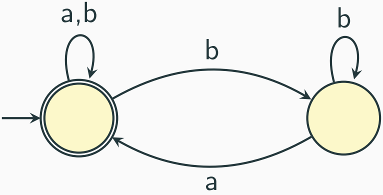
"Non-deterministic choices do not depend on the future"



A non-deterministic co-Büchi automaton which is not history-deterministic

History-Determinism [Henzinger and Piterman, 2006] [Colcombet, 2009]

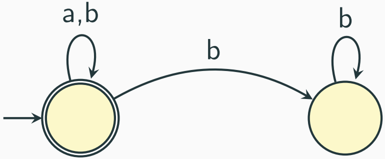
"Non-deterministic choices do not depend on the future"



A non-deterministic co-Büchi automaton which is history-deterministic

History-Determinism [Henzinger and Piterman, 2006] [Colcombet, 2009]

"Non-deterministic choices do not depend on the future"



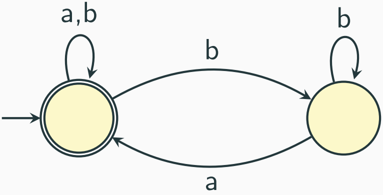
A non-deterministic co-Büchi automaton which is not history-deterministic

The Letter Game

- Two-player game
- Adam chooses a letter
- Eve picks a transition of the automaton
- She wins iff either $\begin{cases} \text{she built an accepting run over } w \\ \text{Adam gave a word } w \notin L(A) \end{cases}$
- Winning strategy: $\lambda : \Sigma^+ \to \Delta$

History-Determinism [Henzinger and Piterman, 2006] [Colcombet, 2009]

"Non-deterministic choices do not depend on the future"



A non-deterministic co-Büchi automaton which is history-deterministic

The Letter Game

- Two-player game
- Adam chooses a letter
- Eve picks a transition of the automaton
- She wins iff either $\begin{cases} \text{she built an accepting run over } w \\ \text{Adam gave a word } w \notin L(A) \end{cases}$
- Winning strategy: $\lambda : \Sigma^+ \to \Delta$

History-Determinism [Henzinger and Piterman, 2006]

Succinctness [Kuperberg and Skrzypczak, 2015]

History-deterministic co-Büchi automata can be *exponentially* more succinct than deterministic ones.

Good-for-gameness [Henzinger and Piterman, 2006]

History-deterministic ω-regular automata "compose well" with games.

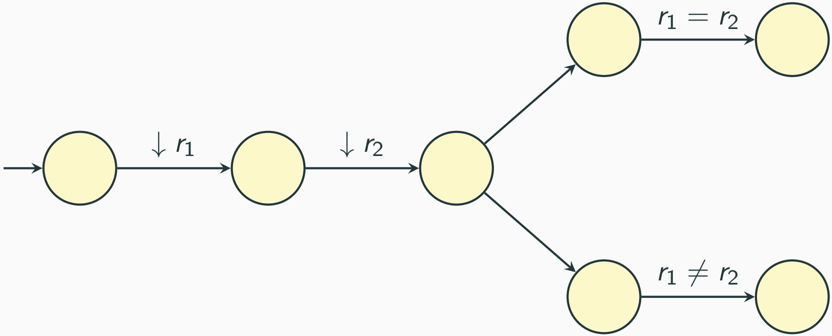
➔ For any game G, the winner of G with winning condition $L(A)$ is the same as the winner of $G \otimes A$.

Lifting to Infinite Alphabets: History-Deterministic Register Automata

Expressivity and Succinctness

Over finite words, history-determinism are as expressive, but exponentially more succinct.

➔ Resolvers depend on equalities between registers.

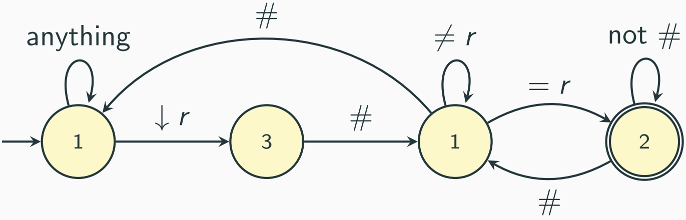


Expressivity and Succinctness

Over *infinite* words, history-determinism is strictly more expressive than determinism.

➔ Blocks of data such that eventually,
some data appears in all blocks

$$L = \left\{ d_0^0 d_1^0 \dots d_{n_0}^0 \# d_0^1 d_1^1 \dots d_{n_1}^1 \# \dots \;\middle|\; \begin{array}{l} \exists d \in \mathcal{D}, \exists N \geq 0, \\ \forall i \geq N, \exists 0 \leq j \leq n_i, d_j^i = d \end{array} \right\}$$

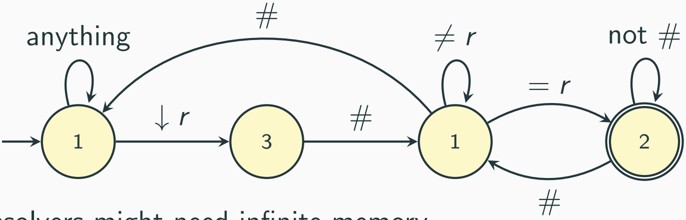


Expressivity and Succinctness

Over *infinite* words, history-determinism is strictly more expressive than determinism.

➔ Blocks of data such that eventually,
some data appears in all blocks

$$L = \left\{ d_0^0 d_1^0 \dots d_{n_0}^0 \# d_0^1 d_1^1 \dots d_{n_1}^1 \# \dots \;\middle|\; \begin{array}{l} \exists d \in \mathcal{D}, \exists N \geq 0, \\ \forall i \geq N, \exists 0 \leq j \leq n_i, d_j^i = d \end{array} \right\}$$



➔ Resolvers might need infinite memory

History-Deterministic Register Automata

Equivalence with Good-for-Gameness

- Recall that over ω-regular languages, history-determinism and good-for-gameness coincide

History-Deterministic Register Automata

Equivalence with Good-for-Gameness

- Recall that over ω-regular languages, history-determinism and good-for-gameness coincide
- The $\Rightarrow$ implication always hold

History-Deterministic Register Automata

Equivalence with Good-for-Gameness

- Recall that over ω-regular languages, history-determinism and good-for-gameness coincide
- The $\Rightarrow$ implication always hold
- The $\Leftarrow$ implication holds if the letter game is determined

Equivalence with Good-for-Gameness

- Recall that over ω-regular languages, history-determinism and good-for-gameness coincide
- The $\Rightarrow$ implication always hold
- The $\Leftarrow$ implication holds if the letter game is determined
- This is the case for register automata over *finite words*, and for co-Büchi ones.

History-Deterministic Register Automata

Equivalence with Good-for-Gameness

- Recall that over ω-regular languages, history-determinism and good-for-gameness coincide
- The $\Rightarrow$ implication always hold
- The $\Leftarrow$ implication holds if the letter game is determined
- This is the case for register automata over *finite words*, and for co-Büchi ones.

Decidability

Over finite words history-determinism is decidable.

Equivalence with Good-for-Gameness

- Recall that over ω-regular languages, history-determinism and good-for-gameness coincide
- The $\Rightarrow$ implication always hold
- The $\Leftarrow$ implication holds if the letter game is determined
- This is the case for register automata over *finite words*, and for co-Büchi ones.

Decidability

Over finite words history-determinism is decidable.

➔ The letter game is then equivalent with the 1-token game, where Adam also constructs a run.

Conclusion

- Relevant class, with applications to synthesis
- Finite word case: runtime verification $\rightsquigarrow$ monitor synthesis for properties with data
- Better understanding of non-determinism

Conclusion

- Relevant class, with applications to synthesis
- Finite word case: runtime verification $\rightsquigarrow$ monitor synthesis for properties with data
- Better understanding of non-determinism

Open problems

- Equivalence with GFG-ness over ω-words
- Decidability of the notion over ω-words
- Are games with co-non-deterministic winning conditions
    - Determined?
    - Decidable?
    - What is their memory structure?

Bibliography i

Colcombet, T. (2009).
The theory of stabilisation monoids and regular cost functions.
In *International Colloquium on Automata, Languages, and Programming*, pages 139–150. Springer.

Exibard, L., Filiot, E., and Reynier, P. (2021).
Synthesis of data word transducers.
Log. Methods Comput. Sci., 17(1).

Henzinger, T. A. and Piterman, N. (2006).
Solving games without determinization.
In *International Workshop on Computer Science Logic*, pages 395–410. Springer.

J.R. Büchi and L.H. Landweber (1969).
Solving sequential conditions finite-state strategies.
Transactions of the American Mathematical Society, 138:295–311.

Kaminski, M. and Francez, N. (1994).
Finite-memory automata.
Theor. Comput. Sci., 134(2):329–363.

Kuperberg, D. and Skrzypczak, M. (2015).
On determinisation of good-for-games automata.
In Halldórsson, M. M., Iwama, K., Kobayashi, N., and Speckmann, B., editors, *Automata, Languages, and*

Programming, pages 299–310, Berlin, Heidelberg. Springer Berlin Heidelberg.