How to Play Optimally for Regular Objectives?

Patricia Bouyer[1], Nathanaël Fijalkow[2],
Mickael Randour[3], **Pierre Vandenhove**[1,3]

[1]Université Paris-Saclay, CNRS, ENS Paris-Saclay, LMF, France

[2]CNRS, LaBRI and Université de Bordeaux, France & University of Warsaw, Poland

[3]F.R.S.-FNRS & UMONS – Université de Mons, Belgium

July 25, 2023 – Highlights'23

Outline

Synthesis problem

Synthesizing **controllers** for **reactive systems** with an **objective**.
Systems and their environment modeled with **zero-sum games**.

How to play?

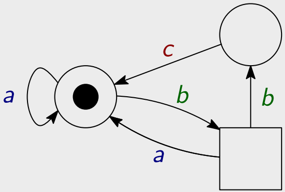
Given an objective, what are the **optimal controllers**?
What are the **smallest** ones?

Results

Characterization of finite-state controllers for *regular objectives*;
computational complexity of finding small ones.

Games

Zero-sum turn-based games on **graphs**



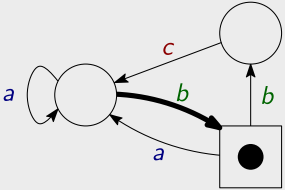
- $C = \{a, b, c\}$, arena $\mathcal{A} = (V, V_1, V_2, E)$.
- Two players $\mathcal{P}_1$ ($\bigcirc$) and $\mathcal{P}_2$ ($\square$)
- **Objective** of $\mathcal{P}_1$ is a set $W \subseteq C^\omega$.
- **Zero-sum**: objective of $\mathcal{P}_2$ is $C^\omega \setminus W$.

How to play?

A **strategy** of a player is a function $\sigma \colon E^* \to E$.
A strategy σ of $\mathcal{P}_1$ is **winning for** W **from** $v \in V$ if all infinite paths from v *consistent with* σ induce an infinite word in W.

Games

Zero-sum turn-based games on graphs



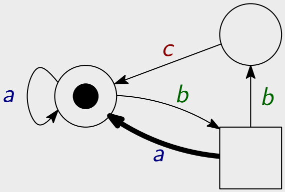
- $C = \{a, b, c\}$, arena $\mathcal{A} = (V, V_1, V_2, E)$.
- Two players $\mathcal{P}_1$ ($\bigcirc$) and $\mathcal{P}_2$ ($\square$) generate an infinite word $w = b$
- **Objective** of $\mathcal{P}_1$ is a set $W \subseteq C^\omega$.
- **Zero-sum**: objective of $\mathcal{P}_2$ is $C^\omega \setminus W$.

How to play?

A **strategy** of a player is a function $\sigma\colon E^* \to E$.
A strategy σ of $\mathcal{P}_1$ is **winning for** W **from** $v \in V$ if all infinite paths from v *consistent with* σ induce an infinite word in W.

Games

Zero-sum turn-based games on graphs



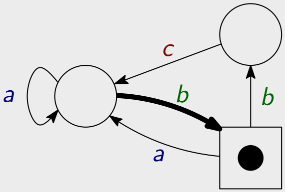
- $C = \{a, b, c\}$, arena $\mathcal{A} = (V, V_1, V_2, E)$.
- Two players $\mathcal{P}_1$ ($\bigcirc$) and $\mathcal{P}_2$ ($\square$) generate an infinite word $w = ba$
- **Objective** of $\mathcal{P}_1$ is a set $W \subseteq C^\omega$.
- **Zero-sum**: objective of $\mathcal{P}_2$ is $C^\omega \setminus W$.

How to play?

A **strategy** of a player is a function $\sigma \colon E^* \to E$.
A strategy σ of $\mathcal{P}_1$ is **winning for** W **from** $v \in V$ if all infinite paths from v *consistent with* σ induce an infinite word in W.

Games

Zero-sum turn-based games on graphs



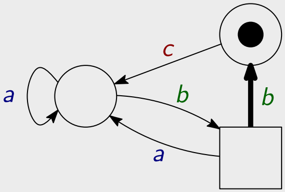
- $C = \{a, b, c\}$, arena $\mathcal{A} = (V, V_1, V_2, E)$.
- Two players $\mathcal{P}_1$ ($\bigcirc$) and $\mathcal{P}_2$ ($\Box$) generate an infinite word $w = bab$
- **Objective** of $\mathcal{P}_1$ is a set $W \subseteq C^\omega$.
- **Zero-sum**: objective of $\mathcal{P}_2$ is $C^\omega \setminus W$.

How to play?

A **strategy** of a player is a function $\sigma \colon E^* \to E$.
A strategy σ of $\mathcal{P}_1$ is **winning for** W **from** $v \in V$ if all infinite paths from v *consistent with* σ induce an infinite word in W.

Games

Zero-sum turn-based games on **graphs**



- $C = \{a, b, c\}$, arena $\mathcal{A} = (V, V_1, V_2, E)$.
- Two players $\mathcal{P}_1$ ($\bigcirc$) and $\mathcal{P}_2$ ($\Box$) generate an infinite word $w = babb$
- **Objective** of $\mathcal{P}_1$ is a set $W \subseteq C^\omega$.
- **Zero-sum**: objective of $\mathcal{P}_2$ is $C^\omega \setminus W$.

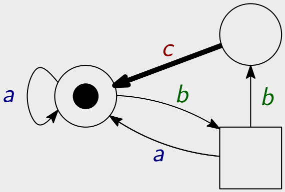
How to play?

A **strategy** of a player is a function $\sigma \colon E^* \to E$.

A strategy σ of $\mathcal{P}_1$ is **winning for** W **from** $v \in V$ if all infinite paths from v *consistent with* σ induce an infinite word in W.

Games

Zero-sum turn-based games on graphs



- $C = \{a, b, c\}$, arena $\mathcal{A} = (V, V_1, V_2, E)$.
- Two players $\mathcal{P}_1$ ($\bigcirc$) and $\mathcal{P}_2$ ($\square$) generate an infinite word $w = babbc \ldots \in C^\omega$.
- **Objective** of $\mathcal{P}_1$ is a set $W \subseteq C^\omega$.
- **Zero-sum**: objective of $\mathcal{P}_2$ is $C^\omega \setminus W$.

How to play?

A **strategy** of a player is a function $\sigma \colon E^* \to E$.
A strategy σ of $\mathcal{P}_1$ is **winning for** W **from** $v \in V$ if all infinite paths from v *consistent with* σ induce an infinite word in W.

Example of a game

Let $C = \{a, b\}$ and

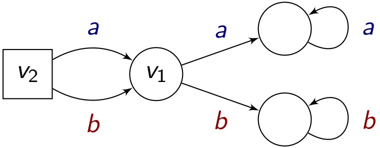
$$W = \{c_1 c_2 \ldots \in C^\omega \mid \exists i \geq 1, c_i = a \;\wedge\; \exists j \geq 1, c_j = b\}.$$

Example of a game

Let $C = \{a, b\}$ and

$$W = \{c_1 c_2 \ldots \in C^\omega \mid \exists i \geq 1, c_i = a \ \wedge \ \exists j \geq 1, c_j = b\}.$$

In the following arena, $\mathcal{P}_1$ has a winning strategy from v_2.

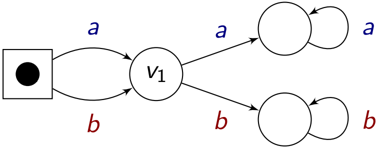


Example of a game

Let $C = \{a, b\}$ and

$$W = \{c_1 c_2 \ldots \in C^\omega \mid \exists i \geq 1, c_i = a \ \land \ \exists j \geq 1, c_j = b\}.$$

In the following arena, $\mathcal{P}_1$ has a winning strategy from v_2.

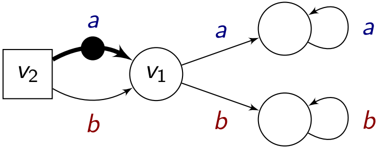


Example of a game

Let $C = \{a, b\}$ and

$$W = \{c_1 c_2 \ldots \in C^\omega \mid \exists i \geq 1, c_i = a \ \wedge \ \exists j \geq 1, c_j = b\}.$$

In the following arena, $\mathcal{P}_1$ has a winning strategy from v_2.

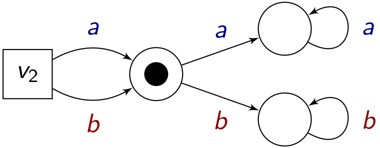


Example of a game

Let $C = \{a, b\}$ and

$$W = \{c_1 c_2 \ldots \in C^\omega \mid \exists i \geq 1, c_i = a \ \wedge \ \exists j \geq 1, c_j = b\}.$$

In the following arena, $\mathcal{P}_1$ has a winning strategy from v_2.

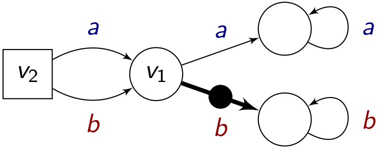


Example of a game

Let $C = \{a, b\}$ and

$$W = \{c_1 c_2 \ldots \in C^\omega \mid \exists i \geq 1, c_i = a \ \wedge \ \exists j \geq 1, c_j = b\}.$$

In the following arena, $\mathcal{P}_1$ has a winning strategy from v_2.

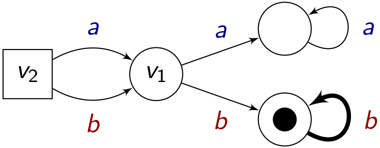


Example of a game

Let $C = \{a, b\}$ and

$$W = \{c_1 c_2 \ldots \in C^\omega \mid \exists i \geq 1, c_i = a \ \wedge \ \exists j \geq 1, c_j = b\}.$$

In the following arena, $\mathcal{P}_1$ has a winning strategy from v_2.

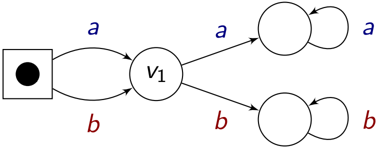


Example of a game

Let $C = \{a, b\}$ and

$$W = \{c_1 c_2 \ldots \in C^\omega \mid \exists i \geq 1, c_i = a \ \wedge \ \exists j \geq 1, c_j = b\}.$$

In the following arena, $\mathcal{P}_1$ has a winning strategy from v_2.

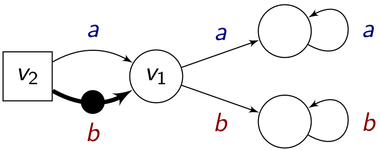


Example of a game

Let $C = \{a, b\}$ and

$$W = \{c_1 c_2 \ldots \in C^\omega \mid \exists i \geq 1, c_i = a \ \wedge \ \exists j \geq 1, c_j = b\}.$$

In the following arena, $\mathcal{P}_1$ has a winning strategy from v_2.

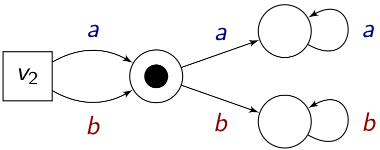


Example of a game

Let $C = \{a, b\}$ and

$$W = \{c_1 c_2 \ldots \in C^\omega \mid \exists i \geq 1, c_i = a \ \wedge \ \exists j \geq 1, c_j = b\}.$$

In the following arena, $\mathcal{P}_1$ has a winning strategy from v_2.

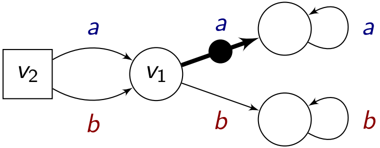


Example of a game

Let $C = \{a, b\}$ and

$$W = \{c_1 c_2 \ldots \in C^\omega \mid \exists i \geq 1, c_i = a \ \wedge \ \exists j \geq 1, c_j = b\}.$$

In the following arena, $\mathcal{P}_1$ has a winning strategy from v_2.

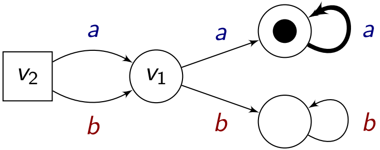


Example of a game

Let $C = \{a, b\}$ and

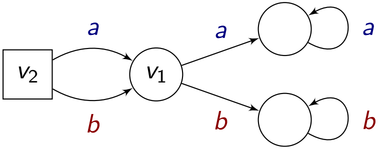
$$W = \{c_1 c_2 \ldots \in C^\omega \mid \exists i \geq 1, c_i = a \ \wedge\ \exists j \geq 1, c_j = b\}.$$

In the following arena, $\mathcal{P}_1$ has a winning strategy from v_2.

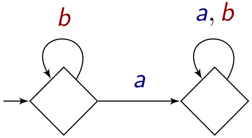


Finite-memory strategies

In this game, $\mathcal{P}_1$ only needs to know whether a or b was seen.



Here, can be tracked with a **finite memory structure** with two states.

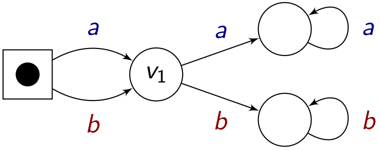


Decisions can only observe the current **arena vertex** and **memory state**.

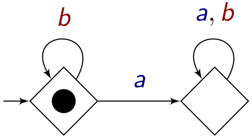
What is a minimal memory structure that suffices in **all** arenas?

Finite-memory strategies

In this game, $\mathcal{P}_1$ only needs to know whether a or b was seen.



Here, can be tracked with a **finite memory structure** with two states.

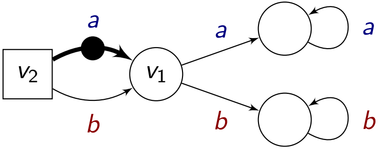


Decisions can only observe the current **arena vertex** and **memory state**.

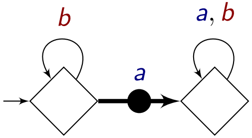
What is a minimal memory structure that suffices in **all** arenas?

Finite-memory strategies

In this game, $\mathcal{P}_1$ only needs to know whether a or b was seen.



Here, can be tracked with a **finite memory structure** with two states.

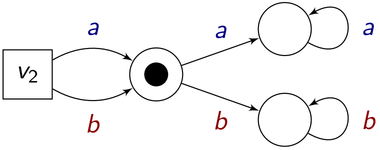


Decisions can only observe the current **arena vertex** and **memory state**.

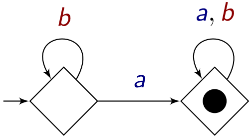
What is a minimal memory structure that suffices in **all** arenas?

Finite-memory strategies

In this game, $\mathcal{P}_1$ only needs to know whether a or b was seen.



Here, can be tracked with a **finite memory structure** with two states.

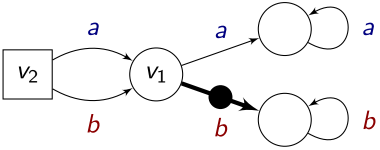


Decisions can only observe the current **arena vertex** and **memory state**.

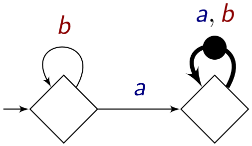
What is a minimal memory structure that suffices in **all** arenas?

Finite-memory strategies

In this game, $\mathcal{P}_1$ only needs to know whether a or b was seen.



Here, can be tracked with a **finite memory structure** with two states.

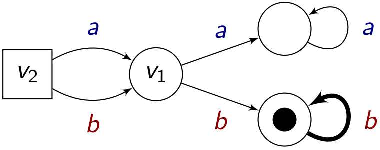


Decisions can only observe the current **arena vertex** and **memory state**.

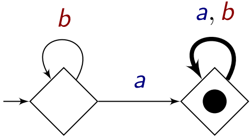
What is a minimal memory structure that suffices in **all** arenas?

Finite-memory strategies

In this game, $\mathcal{P}_1$ only needs to know whether a or b was seen.



Here, can be tracked with a **finite memory structure** with two states.

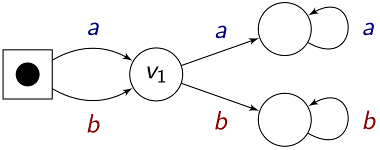


Decisions can only observe the current **arena vertex** and **memory state**.

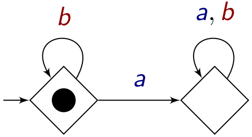
What is a minimal memory structure that suffices in **all** arenas?

Finite-memory strategies

In this game, $\mathcal{P}_1$ only needs to know whether a or b was seen.



Here, can be tracked with a **finite memory structure** with two states.

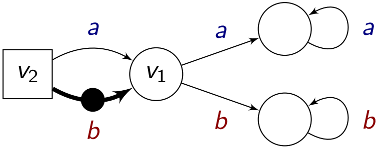


Decisions can only observe the current **arena vertex** and **memory state**.

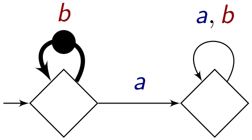
What is a minimal memory structure that suffices in **all** arenas?

Finite-memory strategies

In this game, $\mathcal{P}_1$ only needs to know whether a or b was seen.



Here, can be tracked with a **finite memory structure** with two states.

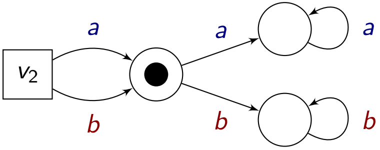


Decisions can only observe the current **arena vertex** and **memory state**.

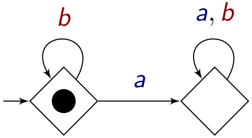
What is a minimal memory structure that suffices in **all** arenas?

Finite-memory strategies

In this game, $\mathcal{P}_1$ only needs to know whether a or b was seen.



Here, can be tracked with a **finite memory structure** with two states.

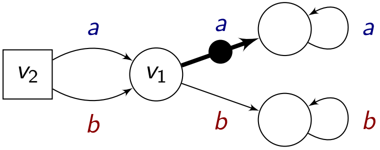


Decisions can only observe the current **arena vertex** and **memory state**.

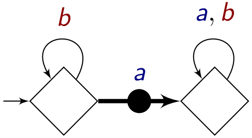
What is a minimal memory structure that suffices in **all** arenas?

Finite-memory strategies

In this game, $\mathcal{P}_1$ only needs to know whether a or b was seen.



Here, can be tracked with a **finite memory structure** with two states.

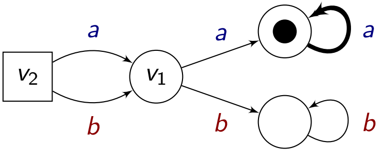


Decisions can only observe the current **arena vertex** and **memory state**.

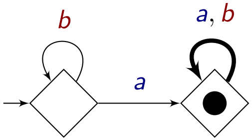
What is a minimal memory structure that suffices in **all** arenas?

Finite-memory strategies

In this game, $\mathcal{P}_1$ only needs to know whether a or b was seen.



Here, can be tracked with a **finite memory structure** with two states.

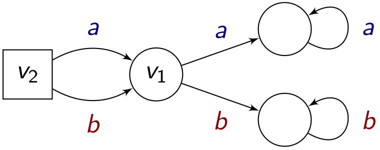


Decisions can only observe the current **arena vertex** and **memory state**.

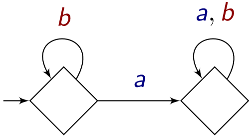
What is a minimal memory structure that suffices in **all** arenas?

Finite-memory strategies

In this game, $\mathcal{P}_1$ only needs to know whether a or b was seen.



Here, can be tracked with a **finite memory structure** with two states.



Decisions can only observe the current **arena vertex** and **memory state**.

What is a minimal memory structure that suffices in **all** arenas?

Regular objectives

This previous objective was a *regular reachability objective*.

Regular objectives

- A **regular reachability objective** is a set LC^ω with $L \subseteq C^*$ regular.
- A **regular safety objective** is a set $C^\omega \setminus LC^\omega$.

- A player wants to **realize** a word in L, the other wants to **prevent** it.
- Expressible as standard **deterministic finite automata**.

Precise quest

Memory requirements of regular objectives

Characterize the memory structures that suffice to make optimal decisions for **regular objectives** in all arenas. Compute **minimal** ones.

Ideas

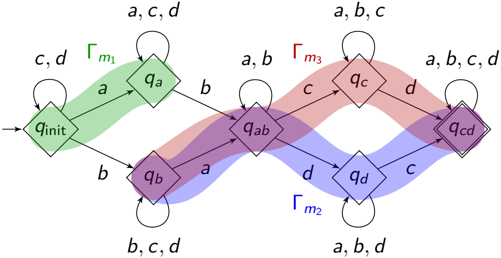
- A **DFA recognizing the regular language** L, seen as a memory structure, always suffices for both players...
- ... but minimal memory structures can be **much smaller**!

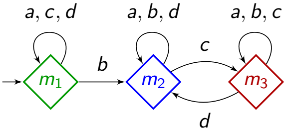
Results (*not detailed here*)

Decidable characterization of **sufficient memory structures** for each kind of objectives.

Automata-theoretic reformulation

$\rightsquigarrow$ Reformulation of "$\mathcal{M}$ suffices for a regular safety objective" into a **"nice" covering of the automaton states**.





Computational complexity: safety

Decision problems

Input: Automaton $\mathcal{D}$ inducing the regular **safety**/**reachability** objective W and $k \in \mathbb{N}$.
Question: $\exists$ memory structure $\mathcal{M}$ with $\leq k$ states that suffices for W?

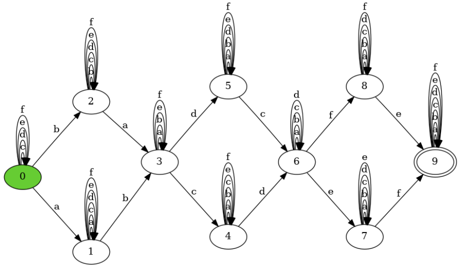
Theorem

These problems are NP-complete.

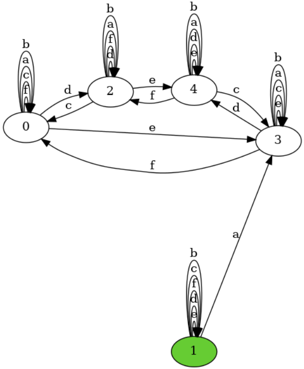
Implementation

Algorithms[1] that find minimal memory structures for regular objectives.

`D =`



`M = memReq.smallest_memory_safety(D)` ⟶



[1] `https://github.com/pvdhove/regularMemoryRequirements`

Conclusion

Summary

- Characterization of the memory structures for **regular objectives**.
- **NP-completeness** of finding small memory structures.

Conclusion

Future work

- Minimal memory structures for all ω-***regular*** *objectives*?
  - ✓ *Muller conditions*,[2,3]
  - ✓ *deterministic Büchi automata* (partially),[4]
  - ✓ **regular objectives**.
- Memory model only observes **colors**... but observing **edges** may need fewer memory states.
  Understood for safety,[5] but not for reachability.

Thanks!

[2] Dziembowski, Jurdziński, and Walukiewicz, "How Much Memory is Needed to Win Infinite Games?", 1997.

[3] Casares, "On the Minimisation of Transition-Based Rabin Automata and the Chromatic Memory Requirements of Muller Conditions", 2022.

[4] Bouyer, Casares, et al., "Half-Positional Objectives Recognized by Deterministic Büchi Automata", 2022.

[5] Colcombet, Fijalkow, and Horn, "Playing Safe", 2014.