Strategy Complexity of Zero-Sum Games on Graphs

Pierre Vandenhove[1,2]

Thesis supervised by Patricia Bouyer[2] and Mickael Randour[1]

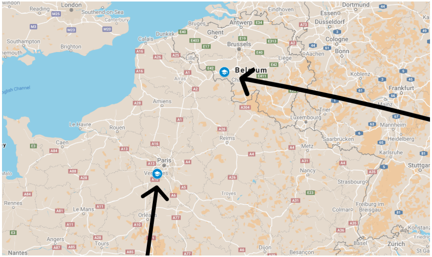
[1]F.R.S.-FNRS & UMONS – Université de Mons, Belgium

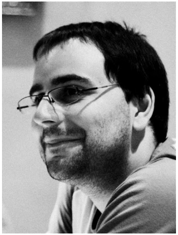
[2]Université Paris-Saclay, CNRS, ENS Paris-Saclay, LMF, France

April 26, 2023 – PhD Public Defense

Context

- Thesis started in **October 2019**.
- Thesis **supervised** by. . .





Mickael Randour,
Université de Mons

Patricia Bouyer,
Laboratoire Méthodes Formelles

- **Public thesis defense**.

Plan

1. Motivate the fields of **verification** and **synthesis**.
2. Explain the focus of my thesis:

   Strategy Complexity of **Zero-Sum Games on Graphs**.
3. Give some intuition about our **results**.

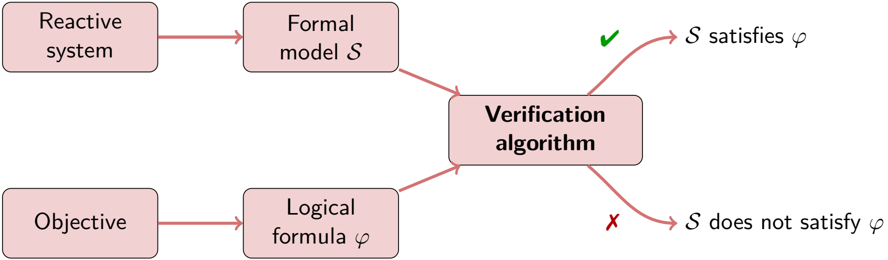
Reactive systems

- **Reactive systems** = systems that continuously interact with their environment (elevator, web server, *robot vacuum cleaner*. . .).

- Must achieve an **objective**
  - using their capabilities (*controllable events*);
  - while **reacting** to events from their environment (*uncontrollable events*).
- Subject to bugs and **errors**, sometimes serious.
- Solution 1: **tests**? Efficient, but not exhaustive.
- Solution 2: **verification** and **synthesis**.

Verification

- **Verification** aims for a formal **proof** that a system achieves its objective, *no matter what happens in the environment*.
- The **objective** describes the desired behaviors.
- Works with abstractions/**models** of systems.

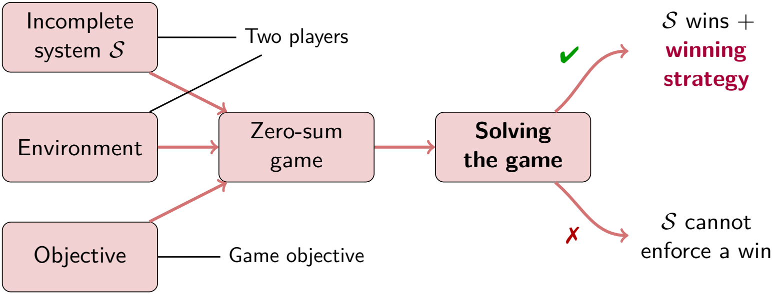


- Downside: requires a "complete" system as an input.

Synthesis

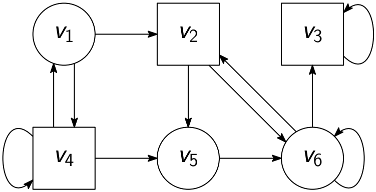
- **Synthesis** seeks to **generate a controller** achieving the objective.
- Accepts an "**incomplete**" description of the system.
- Correct controller **by construction**.
- System and environment are players; the environment is **antagonistic**.

$\rightsquigarrow$ Modeling through a *zero-sum game*.



Zero-sum games on graphs

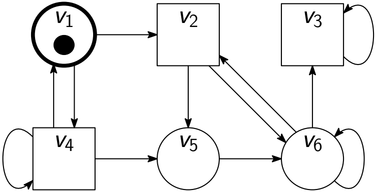
- **Graph** (called **arena**) describing the states of the system.



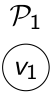
- Two **players**:
  - $\mathcal{P}_1$ (the system) controls the ◯s;
  - $\mathcal{P}_2$ (the environment) controls the □s.
- Interaction of **infinite duration** between the players.

Zero-sum games on graphs

- **Graph** (called **arena**) describing the states of the system.

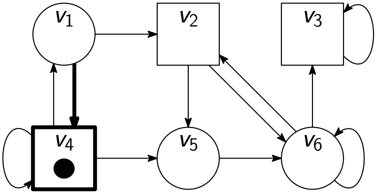


- Two **players**:
  - $\mathcal{P}_1$ (the system) controls the ◯s;
  - $\mathcal{P}_2$ (the environment) controls the □s.
- Interaction of **infinite duration** between the players.

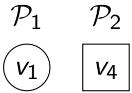


Zero-sum games on graphs

- **Graph** (called **arena**) describing the states of the system.

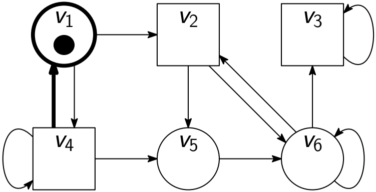


- Two **players**:
  - $\mathcal{P}_1$ (the system) controls the ◯s;
  - $\mathcal{P}_2$ (the environment) controls the □s.
- Interaction of **infinite duration** between the players.

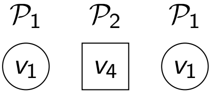


Zero-sum games on graphs

- **Graph** (called **arena**) describing the states of the system.

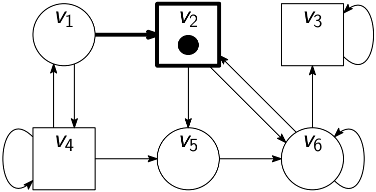


- Two **players**:
  - $\mathcal{P}_1$ (the system) controls the ◯s;
  - $\mathcal{P}_2$ (the environment) controls the □s.
- Interaction of **infinite duration** between the players.



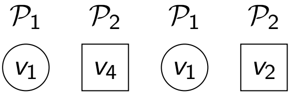
Zero-sum games on graphs

- **Graph** (called **arena**) describing the states of the system.



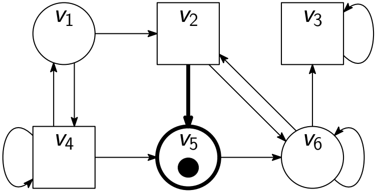
- Two **players**:
  - $\mathcal{P}_1$ (the system) controls the ◯s;
  - $\mathcal{P}_2$ (the environment) controls the □s.

- Interaction of **infinite duration** between the players.

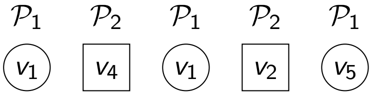


Zero-sum games on graphs

- **Graph** (called **arena**) describing the states of the system.

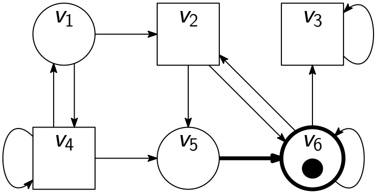


- Two **players**:
  - $\mathcal{P}_1$ (the system) controls the ◯s;
  - $\mathcal{P}_2$ (the environment) controls the □s.
- Interaction of **infinite duration** between the players.

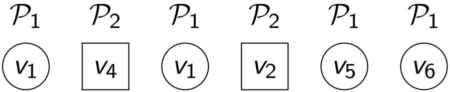


Zero-sum games on graphs

- **Graph** (called **arena**) describing the states of the system.

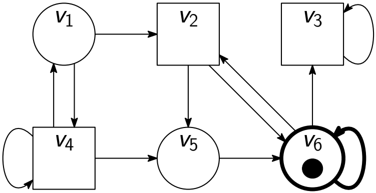


- Two **players**:
  - $\mathcal{P}_1$ (the system) controls the ◯s;
  - $\mathcal{P}_2$ (the environment) controls the □s.
- Interaction of **infinite duration** between the players.

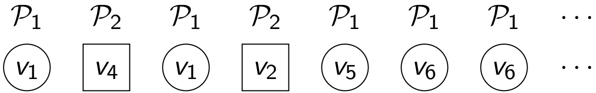


Zero-sum games on graphs

- **Graph** (called **arena**) describing the states of the system.



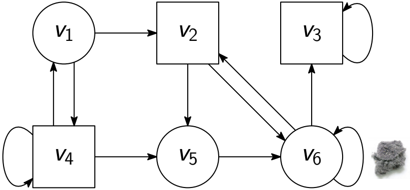
- Two **players**:
  - $\mathcal{P}_1$ (the system) controls the ○s;
  - $\mathcal{P}_2$ (the environment) controls the □s.
- Interaction of **infinite duration** between the players.



Example of objective

Game objective: $\mathcal{P}_1$ should win if and only if the system achieves its objective. We add *events* to the edges.

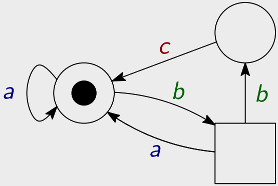
Objective for : *reach some* . (Lazy but a good start!)



- Can $\mathcal{P}_1$ guarantee this **from** v_1 by making decisions only in ◯s?
  Yes, for instance by going to v_2, and then from v_5 to v_6 if necessary.
- Can $\mathcal{P}_1$ guarantee this **from** v_4 by making decisions only in ◯s?
  No, because the opponent may stay in v_4.

Formally

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



- **Colors** (events) C, **arena** $\mathcal{A} = (V_1, V_2, E)$.
- 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 $C^\omega \setminus W$.

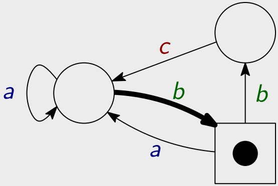
In the previous example:

$C = \{$  ,  $\}$,

$W = \mathrm{Reach}($  $) = \{c_1 c_2 \ldots \in C^\omega \mid \exists i \geq 1, c_i =$  $\}$.

Formally

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



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

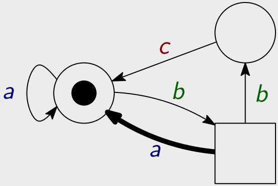
In the previous example:

$C = \{$  ,  $\}$,

$W = \mathrm{Reach}($  $) = \{c_1 c_2 \ldots \in C^\omega \mid \exists i \geq 1, c_i =$  $\}$.

Formally

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



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

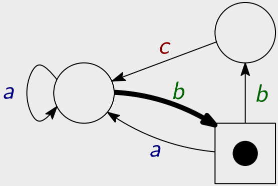
In the previous example:

$C = \{$  ,  $\}$,

$W = \text{Reach}($  $) = \{c_1 c_2 \ldots \in C^\omega \mid \exists i \geq 1, c_i =$  $\}$.

Formally

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



- **Colors** (events) C, **arena** $\mathcal{A} = (V_1, V_2, E)$.
- Two **players** $\mathcal{P}_1$ ($\bigcirc$) and $\mathcal{P}_2$ ($\square$).
  Infinite interaction
  $\rightsquigarrow$ **infinite word** $w = bab$
- **Objective** of $\mathcal{P}_1$ is a set $W \subseteq C^\omega$.
- **Zero-sum**: objective of $\mathcal{P}_2$ is $C^\omega \setminus W$.

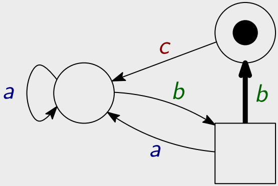
In the previous example:

$C = \{$  ,  $\}$,

$W = \mathrm{Reach}($  $) = \{c_1 c_2 \ldots \in C^\omega \mid \exists i \geq 1, c_i =$  $\}$.

Formally

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



- **Colors** (events) C, **arena** $\mathcal{A} = (V_1, V_2, E)$.
- Two **players** $\mathcal{P}_1$ ($\bigcirc$) and $\mathcal{P}_2$ ($\square$).
  Infinite interaction
  $\rightsquigarrow$ **infinite word** $w = babb$
- **Objective** of $\mathcal{P}_1$ is a set $W \subseteq C^\omega$.
- **Zero-sum**: objective of $\mathcal{P}_2$ is $C^\omega \setminus W$.

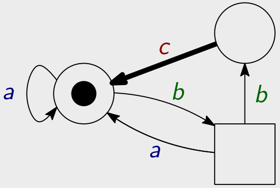
In the previous example:

$C = \{$  ,  $\}$,

$W = \mathrm{Reach}($  $) = \{c_1 c_2 \ldots \in C^\omega \mid \exists i \geq 1, c_i =$  $\}$.

Formally

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



- **Colors** (events) C, **arena** $\mathcal{A} = (V_1, V_2, E)$.
- Two **players** $\mathcal{P}_1$ ($\bigcirc$) and $\mathcal{P}_2$ ($\square$).
  Infinite interaction
  $\rightsquigarrow$ **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 $C^\omega \setminus W$.

Synthesis

Given an arena (with an initial vertex) and an objective, we want to know if $\mathcal{P}_1$ has a *strategy* **winning** against all strategies of the opponent.

Central object: **strategies**

> In general, a strategy is an object that makes decisions using information about the **past interaction**.

A **history** is a sequence $v_0 \xrightarrow{c_1} v_1 \xrightarrow{c_2} \ldots \xrightarrow{c_n} v_n$ of vertices/edges of $\mathcal{A}$.

Definition

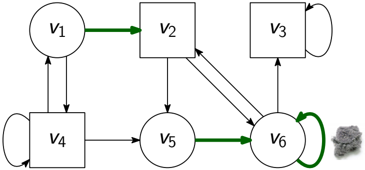
A **strategy** of $\mathcal{P}_1$ is a function

$$\sigma\colon \{\text{histories of } \mathcal{A} \text{ ending in } \bigcirc\} \to E.$$

To solve a game, try to exhibit a winning strategy to show that a player wins. But...

- strategies may be **hard to describe** (set of histories is infinite ☹);
- there are **infinitely many** strategies (cannot try them all ☹).

Describing strategies



In the example, a winning strategy only looks at the current ◯:

$\sigma\colon$ ~~{histories of $\mathcal{A}$ ending in ◯}~~ $\{v_1, v_5, v_6\} \to E.$

Easy to describe. Such a strategy is called **memoryless**.
Not a coincidence!

Memoryless determinacy

For a **reachability** objective, in **all** arenas, when winning is possible for a player, it is always possible to win with a **memoryless strategy**!

Memoryless determinacy

Property

An objective has the property of

memoryless determinacy

if, whenever a player has a **winning strategy**, this player even has a **memoryless winning strategy** (no matter the arena).

This strong property also holds for many other (complex) objectives!

Why is memoryless determinacy nice?

Main advantage: easy **algorithm** to solve the games
$\rightsquigarrow$ solves the synthesis problem for memoryless-determined objectives!

Algorithm (for a finite arena $\mathcal{A}$)

- $\mathcal{P}_1$ and $\mathcal{P}_2$ have only **finitely many** *memoryless* strategies.
- Enumerate the *memoryless* strategies of $\mathcal{P}_1$, and check if **there is one** that wins against **all** *memoryless* strategies of $\mathcal{P}_2$.

$\rightsquigarrow$ Not the most efficient for Reach(), but not bad for more complex objectives!

But unfortunately, memoryless strategies **do not** always suffice to win.

Memoryless strategies do not always suffice (1/2)

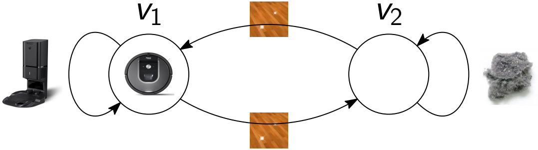
More complex objective for the **vacuum cleaner**:

see both  and  infinitely often. (Still a bit simple but good effort!)

Formally, $C = \{$ , ,  $\}$,

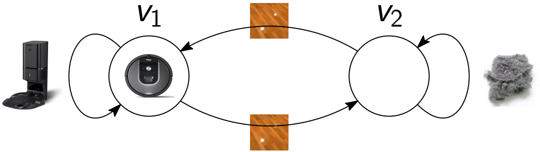
$W = \{c_1 c_2 \ldots \in C^\omega \mid \exists^\infty i, c_i =$  $\wedge \exists^\infty j, c_j =$  $\}$.

In this arena, $\mathcal{P}_1$ **can win** from v_1, but **not** with a memoryless strategy.



Memoryless strategies do not always suffice (2/2)

Objective: see both  and  infinitely often.



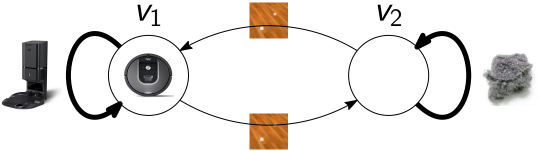
There are 4 **memoryless strategies**, inducing from v_1:

-
-
-
-

Compromise: use memory, but a **finite** amount.

Memoryless strategies do not always suffice (2/2)

Objective: see both  and  infinitely often.



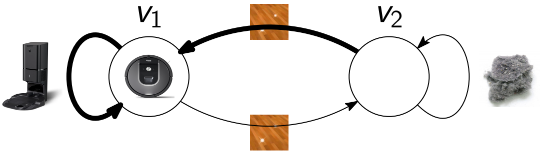
There are 4 **memoryless strategies**, inducing from v_1:

-  $\ldots \notin W$
-
-
-

Compromise: use memory, but a **finite** amount.

Memoryless strategies do not always suffice (2/2)

Objective: see both  and  infinitely often.



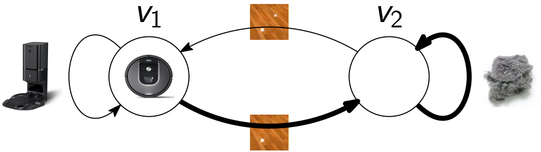
There are 4 **memoryless strategies**, inducing from v_1:

-  $\ldots \notin W$
-  $\ldots \notin W$
-
-

Compromise: use memory, but a **finite** amount.

Memoryless strategies do not always suffice (2/2)

Objective: see both  and  infinitely often.



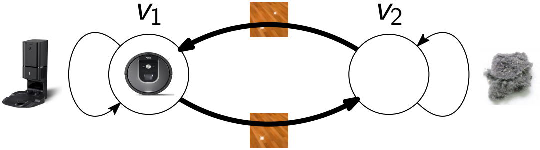
There are 4 **memoryless strategies**, inducing from v_1:

-  $\ldots \notin W$
-  $\ldots \notin W$
-  $\ldots \notin W$
-

Compromise: use memory, but a **finite** amount.

Memoryless strategies do not always suffice (2/2)

Objective: see both  and  infinitely often.



There are 4 **memoryless strategies**, inducing from v_1:

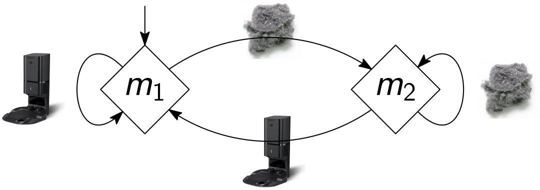
-  $\ldots \notin W$
-  $\ldots \notin W$
-  $\ldots \notin W$
-  $\ldots \notin W$

Compromise: use memory, but a **finite** amount.

Finite-memory strategies

- Even if memoryless strategies do not suffice to win, can we condense the information used by winning strategies **in a finite way**?
- Loss of information (not the full history), but hopefully sufficient!

We store information in finite **memory structures**.

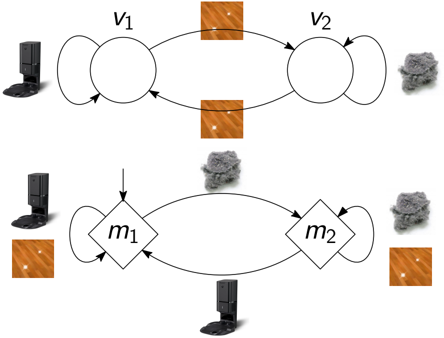


- Their state is **automatically updated** given the events from game.
- The current **state** gives information to help make decisions.

Back to the previous example

We define a winning strategy

$$\sigma\colon \{v_1, v_2\} \times \{m_1, m_2\} \to E.$$



$$\sigma(v_1, m_1) = \longrightarrow v_2$$

$$\sigma(v_2, m_1) = \longrightarrow v_2$$

$$\sigma(v_2, m_2) = \longrightarrow v_1$$

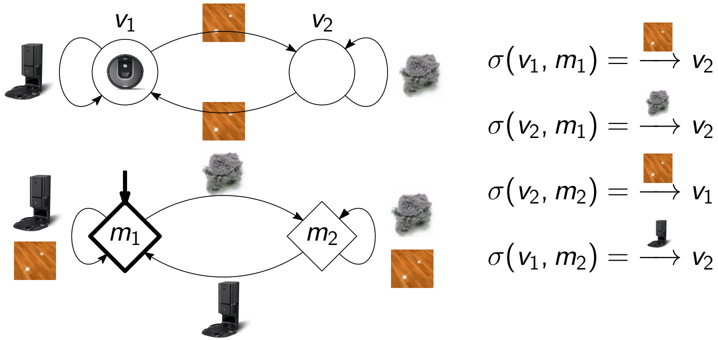
$$\sigma(v_1, m_2) = \longrightarrow v_2$$

- This memory structure **suffices to win** in this arena.
- In **all** arenas, if winning is possible, **finite memory** suffices to win!

Back to the previous example

We define a winning strategy

$$\sigma\colon \{v_1, v_2\} \times \{m_1, m_2\} \to E.$$

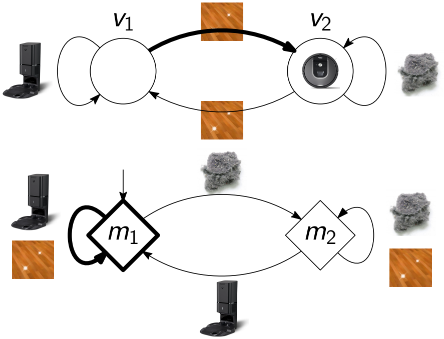


- This memory structure **suffices to win** in this arena.
- In **all** arenas, if winning is possible, **finite memory** suffices to win!

Back to the previous example

We define a winning strategy

$$\sigma\colon \{v_1, v_2\} \times \{m_1, m_2\} \to E.$$



$$\sigma(v_1, m_1) = \longrightarrow v_2$$

$$\sigma(v_2, m_1) = \longrightarrow v_2$$

$$\sigma(v_2, m_2) = \longrightarrow v_1$$

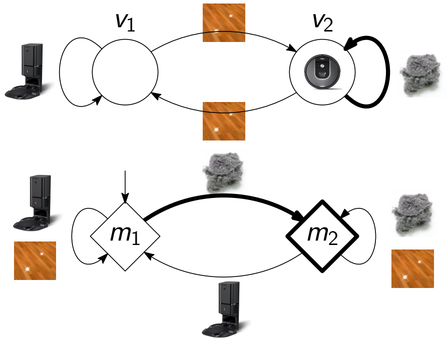
$$\sigma(v_1, m_2) = \longrightarrow v_2$$

- This memory structure **suffices to win** in this arena.
- In **all** arenas, if winning is possible, **finite memory** suffices to win!

Back to the previous example

We define a winning strategy

$$\sigma\colon \{v_1, v_2\} \times \{m_1, m_2\} \to E.$$



$$\sigma(v_1, m_1) = \longrightarrow v_2$$

$$\sigma(v_2, m_1) = \longrightarrow v_2$$

$$\sigma(v_2, m_2) = \longrightarrow v_1$$

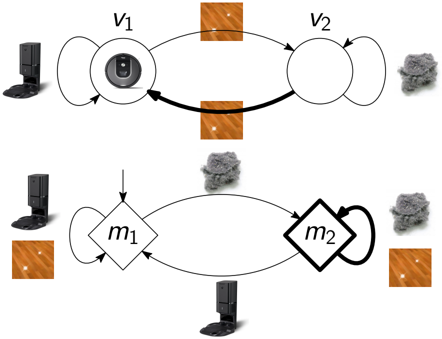
$$\sigma(v_1, m_2) = \longrightarrow v_2$$

- This memory structure **suffices to win** in this arena.
- In **all** arenas, if winning is possible, **finite memory** suffices to win!

Back to the previous example

We define a winning strategy

$$\sigma \colon \{v_1, v_2\} \times \{m_1, m_2\} \to E.$$



$$\sigma(v_1, m_1) = \longrightarrow v_2$$

$$\sigma(v_2, m_1) = \longrightarrow v_2$$

$$\sigma(v_2, m_2) = \longrightarrow v_1$$

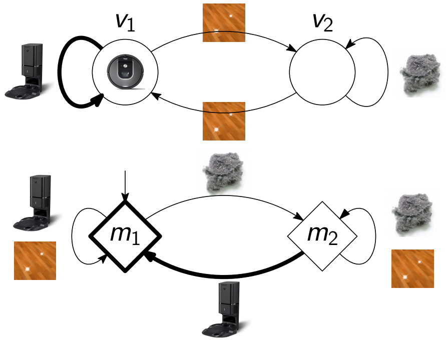
$$\sigma(v_1, m_2) = \longrightarrow v_2$$

- This memory structure **suffices to win** in this arena.
- In **all** arenas, if winning is possible, **finite memory** suffices to win!

Back to the previous example

We define a winning strategy

$$\sigma\colon \{v_1, v_2\} \times \{m_1, m_2\} \to E.$$



$$\sigma(v_1, m_1) = \longrightarrow v_2$$

$$\sigma(v_2, m_1) = \longrightarrow v_2$$

$$\sigma(v_2, m_2) = \longrightarrow v_1$$

$$\sigma(v_1, m_2) = \longrightarrow v_2$$

- This memory structure **suffices to win** in this arena.
- In **all** arenas, if winning is possible, **finite memory** suffices to win!

Finite-memory determinacy

An objective has the property of

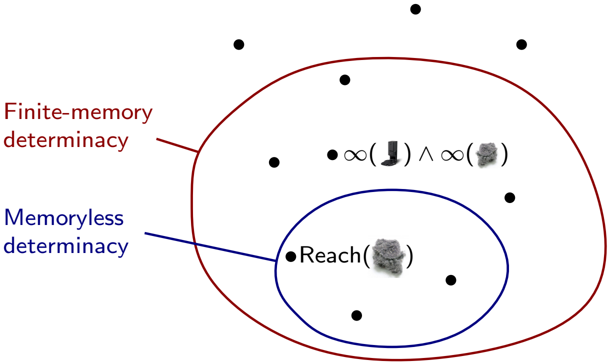
finite-memory determinacy

if, whenever a player has a winning strategy, this player also has a **finite-memory** winning strategy.

Why is it nice?

When the memory structure is known, **finite-memory determinacy** also makes the **synthesis problem** solvable!

Classifying **objectives**



Strategy complexity

Given an **objective**, understand if **simple** strategies suffice to win, or if **complex** strategies are required to win *when possible*.

Memoryless determinacy is well-understood.[1,2,3,4,5,6]
$\rightsquigarrow$ Easy to prove that an objective is memoryless-determined or not.

Finite-memory determinacy is less well-understood.

[1] Aminof and Rubin, "First-cycle games", 2017.

[2] Kopczyński, "Half-Positional Determinacy of Infinite Games", 2006.

[3] Bianco et al., "Exploring the boundary of half-positionality", 2011.

[4] Gimbert and Zielonka, "Games Where You Can Play Optimally Without Any Memory", 2005.

[5] Colcombet and Niwiński, "On the positional determinacy of edge-labeled games", 2006.

[6] Ohlmann, "Characterizing Positionality in Games of Infinite Duration over Infinite Graphs", 2023.

Contributions

In the thesis: focus on **finite-memory strategies**.

Research agenda

1. Understand for which **objectives** finite-memory strategies suffice.
2. When they suffice, find **small** sufficient **memory structures** (i.e., the minimal amount of information to make optimal decisions).

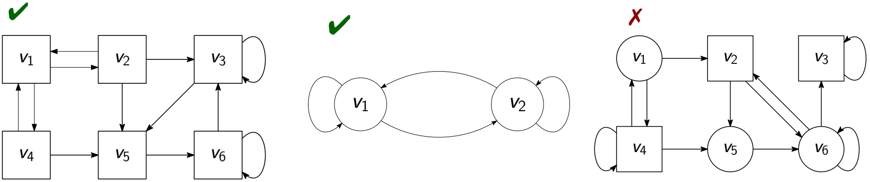
Part I
Theoretical results $\rightsquigarrow$
characterizations, boundaries;
as few hypotheses as possible.

Part II
Practical results $\rightsquigarrow$
automatically compute
small memory structures
for concrete classes of objectives.

Part I: General conditions for finite-memory determinacy

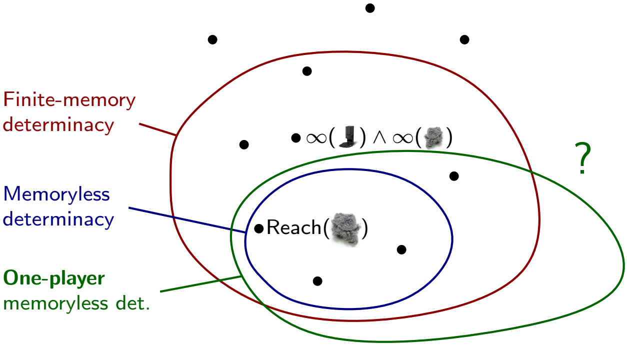
One-player games

- A simpler kind of game is a **one-player game**, in which a **single player** controls **all the vertices** (roughly, a graph).



One-player games

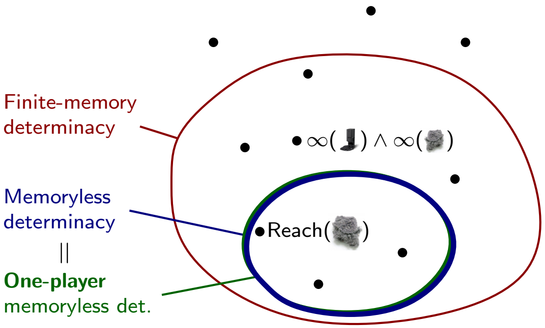
- A simpler kind of game is a **one-player game**, in which a **single player** controls **all the vertices** (roughly, a graph).
- Easier to prove memoryless determinacy in one-player games, but seemingly weaker than in two-player games:



Yet. . .

Nice reduction for memoryless determinacy

... they coincide [GZ05][7]!



⇝ **Reduces** a problem about strategy complexity in **two-player** games to a problem in **one-player** games! Very useful.

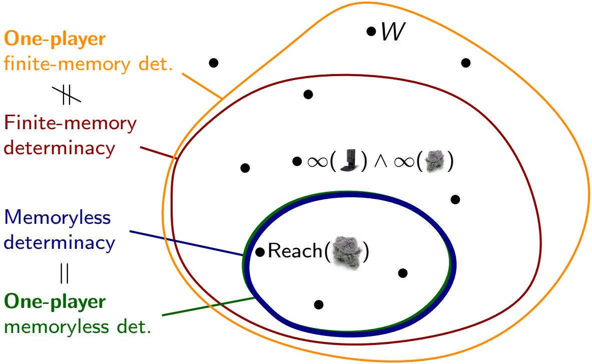
What about **finite-memory** determinacy?

[7] Gimbert and Zielonka, "Games Where You Can Play Optimally Without Any Memory", 2005.

Not as nice ☹

We found an objective W such that:

- finite-memory strategies suffice in all **one-player** games,
- but infinite memory is required in a **two-player** game.



For W, the **size of the memory** depends on the **size of the arena**...

Restriction of finite-memory determinacy

Let W be an objective.

Reminder: finite-memory determinacy

Objective W is **finite-memory determined** if

for all arenas $\mathcal{A}$, **there exists** a finite memory structure $\mathcal{M}$ such that $\mathcal{M}$ suffices to win in $\mathcal{A}$.

Arena-independence

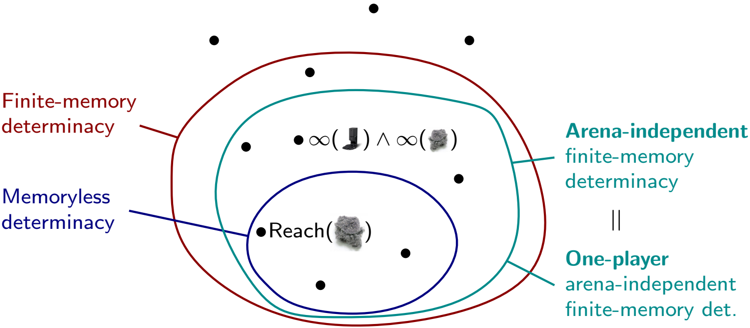
Objective W is **arena-independent** **finite-memory determined** if

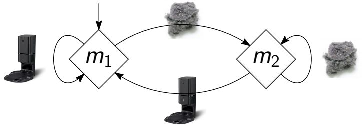
there exists a finite memory structure $\mathcal{M}$ such that **for all** arenas $\mathcal{A}$, $\mathcal{M}$ suffices to win in $\mathcal{A}$.

Stronger property ($\mathcal{M}$ cannot depend on $\mathcal{A}$).

Arena-independent finite-memory determinacy

Between memoryless and finite-memory determinacy:



It contains $\infty($  $) \wedge \infty($  $)$ (with $\mathcal{M} =$ ).

Also reducible to the same property, but over one-player games!

Nice property

One-to-two-player arena-independent finite-memory lift

Let W be an objective and $\mathcal{M}_1$, $\mathcal{M}_2$ be memory structures. If

- in **one-player** arenas of $\mathcal{P}_1$, $\mathcal{P}_1$ has winning strategies using $\mathcal{M}_1$,
- in **one-player** arenas of $\mathcal{P}_2$, $\mathcal{P}_2$ has winning strategies using $\mathcal{M}_2$,

then both players have winning strategies using $\mathcal{M}_1 \otimes \mathcal{M}_2$ in **two-player** arenas.

Robust property: holds over the classes of **finite** and **infinite** arenas.

Applicability?

Even if stronger than finite-memory determinacy, still encompasses many objectives. Not the least being. . .

ω-regular objectives

Important class of objectives

The **ω-regular languages** are a natural generalization of regular languages to languages of **infinite** words.

Theorem[8]

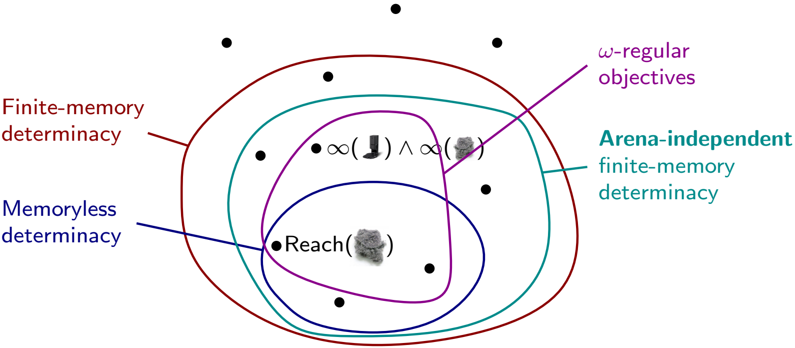
The ω-regular objectives are *arena-independent* finite-memory determined.

$\rightsquigarrow$ Synthesis with such objectives can be done!

[8] Büchi and Landweber, "Definability in the Monadic Second-Order Theory of Successor", 1969; Rabin, "Decidability of Second-Order Theories and Automata on Infinite Trees", 1969; Gurevich and Harrington, "Trees, Automata, and Games", 1982.

ω-regular objectives

Using this theorem, ω-regular objectives are somewhere there:



Strategic characterization

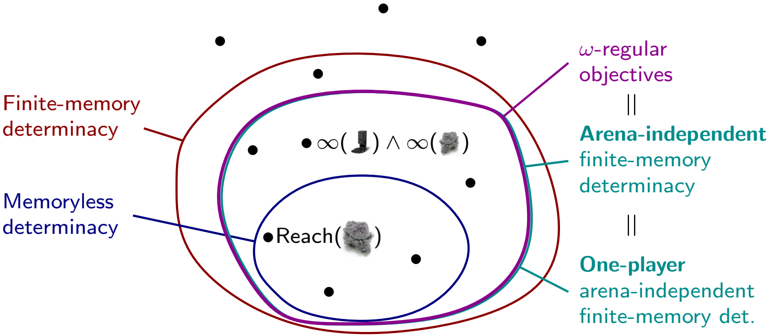
Over games played on *infinite arenas*, we have:

Contribution

An objective is ω-**regular**

$$\iff$$

it is **arena-independent finite-memory determined**.



Summary of Part I

Contributions

- **Characterizations** of kinds of finite-memory determinacy in various contexts.
- *Strengthens the links between memory structures and representations of the objectives.*
- Generalizes [GZ05],[9] [CN06][10] (about memoryless strategies).

Related publications

- Bouyer, Le Roux, Oualhadj, Randour, V. (CONCUR'20 & LMCS) "Games Where You Can Play Optimally with Arena-Independent Finite Memory"
- Bouyer, Randour, V. (STACS'22 & TheoretiCS) "Characterizing Omega-Regularity through Finite-Memory Determinacy of Games on Infinite Graphs"

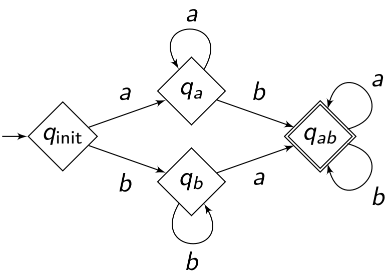
[9] Gimbert and Zielonka, "Games Where You Can Play Optimally Without Any Memory", 2005.

[10] Colcombet and Niwiński, "On the positional determinacy of edge-labeled games", 2006.

Part II: How many memory states for precise objectives?

Regular languages (1/2)

Automata are used to define sets of finite words. They accept the finite words that can be read from the initial state →◇ to the final state ◇.



This automaton

- accepts *aab* ✔
- rejects *aa* ✗
- accepts *baab* ✔
- ...

This automaton accepts exactly finite words that see both *a* and *b*.

Regular languages (2/2)

Sets of words that can be defined by an automaton are called **regular**.

Regular objectives

Assume the objective of $\mathcal{P}_1$ is to achieve a word from a regular language L (i.e., $W = LC^\omega$).
What is a **minimal** memory structure that suffices in all arenas?

The whole automaton suffices as a memory structure, but not necessary!

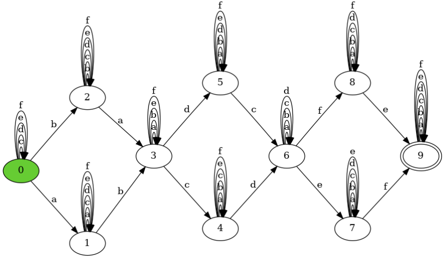
Contributions

- **Characterization** of the memory structures through properties of the language.
- This problem **can be solved** with an algorithm, but **not in an efficient way** (*the related decision problem is NP-complete*).

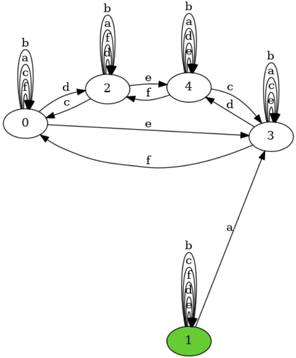
Implementation

Algorithms that find **minimal memory structures** for regular objectives for both players, starting from an automaton, *using a SAT solver*.

`D =`



`M = memReq.smallest_memory_safety(D)` →



Summary of Part II

Contributions

- Ways to **automatically compute** the smallest memory structures for classes of ω-*regular* objectives.
- Work on regular objectives and on *deterministic Büchi automata*.

Related publications

- Bouyer, Fijalkow, Randour, V. (Accepted to ICALP'23) "How to Play Optimally for Regular Objectives?"
- Bouyer, Casares, Randour, V. (CONCUR'22) "Half-Positional Objectives Recognized by Deterministic Büchi Automata"

Conclusion

Future works

- More expressive **game models** (e.g., what if both players can make decisions *at the same time*?).
- More expressive **strategy models** (beyond *finite-state machines*).
- Compute minimal memory structures of **all** ω-regular objectives.

Thank you
for your attention!