Memory complexity for winning games on graphs

Patricia Bouyer

Laboratoire Méthodes Formelles
Université Paris-Saclay, CNRS, ENS Paris-Saclay
France

Based on joined work with Stéphane Le Roux, Youssouf Oualhadj, Michael Randour, Pierre Vandenhove. Thanks to Pierre for his slides
Motivation
—
The setting
My field of research: Formal methods

Give guarantees (+ certificates) on functionalities or performances
Model-checking

System

Properties
Model-checking

System

Properties
Model-checking

System

Properties
Model-checking

System

Properties

Diagram showing a system with a clock and a red X symbol.
Model-checking

System

Properties

\[ \varphi = AG \neg \text{crash} \land \left( \mathbb{P}(F_{\leq 2h_{\text{arr}}} \geq 0.9) \right) \]
Model-checking

\[ \varphi = AG \neg \text{crash} \land \left( \mathbb{P}(F_{\leq 2h_{\text{arr}}}) \geq 0.9 \right) \]
Model-checking

System

Properties

$\varphi = AG \neg \text{crash} \land \left( \mathbb{P}(F_{\leq 2h \text{arr}}) \geq 0.9 \right)$

Yes/No/Why?
Control or synthesis

System

Properties

Control/synthesis algorithm

\[ \varphi = AG \neg \text{crash} \land \left( \mathbb{P}(F_{\leq 2h_{\text{arr}}} \geq 0.9) \right) \]

No/Yes/How?
Strategy synthesis for two-player games

Find good and simple controllers for systems interacting with an antagonistic environment
Strategy synthesis for two-player games

Find good and simple controllers for systems interacting with an antagonistic environment

Good?

Performance w.r.t. objectives / payoffs / preference relations
Strategy synthesis for two-player games

Find good and simple controllers for systems interacting with an antagonistic environment

Good?
Performance w.r.t. objectives / payoffs / preference relations

Simple?
Minimal information for deciding the next steps
The talk in one slide

Strategy synthesis for two-player games

Find good and simple controllers for systems interacting with an antagonistic environment

Good?
Performance w.r.t. objectives / payoffs / preference relations

Simple?
Minimal information for deciding the next steps

When are simple strategies sufficient to play optimally?
Our general approach

[Tho95] On the synthesis of strategies in infinite games (STACS'95).
[Tho02] Thomas. Infinite games and verification (CAV'02).
[BCJ18] Bloem, Chatterjee, Jobstmann. Graph games and reactive synthesis (Handbook of Model-Checking).
Our general approach

- Use **graph-based game models** (state machines) to represent the system and its evolution

[Tho95] On the synthesis of strategies in infinite games (STACS'95).
[Tho02] Thomas. Infinite games and verification (CAV'02).
[BCJ18] Bloem, Chatterjee, Jobstmann. Graph games and reactive synthesis (Handbook of Model-Checking).
Our general approach

- Use **graph-based game models** (state machines) to represent the system and its evolution

- Use **game theory concepts** to express admissible situations
  - Winning strategies
  - (Pareto-)Optimal strategies
  - Nash equilibria
  - Subgame-perfect equilibria
  - ...

[Tho95] On the synthesis of strategies in infinite games (STACS'95).
[Tho02] Thomas. Infinite games and verification (CAV'02).
[BCJ18] Bloem, Chatterjee, Jobstmann. Graph games and reactive synthesis (Handbook of Model-Checking).
Games
What they often are
Games
A broader sense

Goal

- Model and analyze (using math. tools) situations of interactive decision making
Games
A broader sense

Goal
- Model and analyze (using math. tools) situations of interactive decision making

Ingredients
- Several decision makers (players)
- Possibly each with different goals
- The decision of each player impacts the outcome of all
Games
A broader sense

Goal
- Model and analyze (using math. tools) situations of interactive decision making

Ingredients
- Several decision makers (players)
- Possibly each with different goals
- The decision of each player impacts the outcome of all

Wide range of applicability
« [...] it is a context-free mathematical toolbox. »
- Social science: e.g. social choice theory
- Theoretical economics: e.g. models of markets, auctions
- Political science: e.g. fair division
- Biology: e.g. evolutionary biology
- ...

Games
A broader sense

Goal
• Model and analyze (using math. tools) situations of interactive decision making

Ingredients
• Several decision makers (players)
• Possibly each with different goals
• The decision of each player impacts the outcome of all

Wide range of applicability
• Social science: e.g. social choice theory
• Theoretical economics: e.g. models of markets, auctions
• Political science: e.g. fair division
• Biology: e.g. evolutionary biology
• …

« [...] it is a context-free mathematical toolbox. »

Games on graphs

\[ \mathcal{G} = (S, s_0, S_1, S_2, E) \]

- States
  - \( s_0 \) to \( s_2 \) : player \( P_1 \)
  - \( s_3 \) to \( s_4 \) : player \( P_2 \)

- Edges
Games on graphs

\[ \mathcal{G} = (S, s_0, S_1, S_2, E) \]

- States
  - \( s_0 \)
  - \( s_1 \)
  - \( s_2 \)
  - \( s_3 \)
  - \( s_4 \)
  - \( \text{smiley face} \)

- Edges

- \( \text{player } P_1 \)
  - \( s_0 \to s_1 \)
  - \( s_1 \to s_0 \)
  - \( s_0 \to s_2 \)

- \( \text{player } P_2 \)
  - \( s_3 \to s_4 \)
  - \( s_4 \to s_3 \)
  - \( s_3 \to \text{smiley face} \)
  - \( \text{smiley face} \to s_3 \)
  - \( s_4 \to s_2 \)
  - \( s_2 \to s_4 \)
Games on graphs

\[ G = (S, s_0, S_1, S_2, E) \]

States

- \( \text{player } P_1 \)

- \( \text{player } P_2 \)

Edges

- \( s_0 \rightarrow s_1 \)

1. \( P_1 \) chooses the edge \((s_0, s_1)\)
Games on graphs

\[ G = (S, s_0, S_1, S_2, E) \]

- \( s_0 \rightarrow s_1 \rightarrow s_4 \)

1. \( P_1 \) chooses the edge \((s_0, s_1)\)
2. \( P_2 \) chooses the edge \((s_1, s_4)\)
Games on graphs

\[ \mathcal{G} = (S, s_0, S_1, S_2, E) \]

- \( s_0 \to s_1 \to s_4 \to s_2 \)

1. \( P_1 \) chooses the edge \((s_0, s_1)\)
2. \( P_2 \) chooses the edge \((s_1, s_4)\)
3. \( P_2 \) chooses the edge \((s_4, s_2)\)

States

- pink circle: player \( P_1 \)
- blue square: player \( P_2 \)

Edges
Games on graphs

\[ \mathcal{G} = (S, s_0, S_1, S_2, E) \]

- \( s_0 \): player \( P_1 \)
- \( s_1 \): player \( P_2 \)

1. \( P_1 \) chooses the edge \((s_0, s_1)\)
2. \( P_2 \) chooses the edge \((s_1, s_4)\)
3. \( P_2 \) chooses the edge \((s_4, s_2)\)
4. \( P_1 \) chooses the edge \((s_2, 😊)\)
Games on graphs

\[ \mathcal{G} = (S, s_0, S_1, S_2, E) \]

- \( \text{States} \)
  - Pink circle: player \( P_1 \)
  - Blue square: player \( P_2 \)

- \( \text{Edges} \)

1. \( P_1 \) chooses the edge \((s_0, s_1)\)
2. \( P_2 \) chooses the edge \((s_1, s_4)\)
3. \( P_2 \) chooses the edge \((s_4, s_2)\)
4. \( P_1 \) chooses the edge \((s_2, \smiley)\)

States: \( s_0 \rightarrow s_1 \rightarrow s_4 \rightarrow s_2 \rightarrow \smiley \)
Games on graphs

\[ \mathcal{G} = (S, s_0, S_1, S_2, E) \]

States

-圆形节点：玩家\( P_1 \)
-方形节点：玩家\( P_2 \)

Edges

-边表示游戏的进行

1. \( P_1 \)选择边\( (s_0, s_1) \)
2. \( P_2 \)选择边\( (s_1, s_4) \)
3. \( P_2 \)选择边\( (s_4, s_2) \)
4. \( P_1 \)选择边\( (s_2, \, \smiley) \)

Players use **strategies** to play. A strategy for \( P_i \) is \( \sigma_i : S^*S_i \to E \)
Objectives for the players

\[ C = \{ a, b \} \]

\[ E \subseteq S \times C \times S \]
Objectives for the players

- Winning objective for \( P_i \): \( W_i \subseteq C^\omega \), e.g. \( W_1 = C^* \cdot b \cdot C^\omega \)

\[
C = \{ a, b \} \\
E \subseteq S \times C \times S
\]
Objectives for the players

- Winning objective for $P_i$: $W_i \subseteq C^\omega$, e.g. $W_1 = C^* \cdot b \cdot C^\omega$

- Payoff function: $p_i: C^\omega \rightarrow \mathbb{R}$, e.g. mean-payoff

$$C = \{a, b\}$$
$$E \subseteq S \times C \times S$$
Objectives for the players

- Winning objective for $P_i: W_i \subseteq C^\omega$, e.g. $W_1 = C^* \cdot b \cdot C^\omega$

- Payoff function: $p_i: C^\omega \rightarrow \mathbb{R}$, e.g. mean-payoff

- Preference relation: $\sqsubseteq_i \subseteq C^\omega \times C^\omega$
  (total preorder)
Objectives for the players

Winning objective for $P_i$: $W_i \subseteq C^\omega$, e.g. $W_1 = C^* \cdot b \cdot C^\omega$

Payoff function: $p_i$: $C^\omega \to \mathbb{R}$, e.g. mean-payoff

Preference relation: $\sqsubseteq_i \subseteq C^\omega \times C^\omega$ (total preorder)

Zero-sum hypothesis

$C = \{a, b\}$

$E \subseteq S \times C \times S$

$W_2 = W_1^c$

$p_1 + p_2 = 0$

$\sqsubseteq_2 = \sqsubseteq_1^{-1}$
What does it mean to win a game?
What does it mean to win a game?

- Play $\rho = s_0s_1s_2\ldots$ is compatible with $\sigma_i$ whenever $s_j \in S_i$ implies $(s_j, s_{j+1}) = \sigma_i(s_0s_1\ldots s_j)$. We write Out$(\sigma_i)$. 
Outcomes of a strategy
Outcomes of a strategy

- Strategy $\sigma$
Outcomes of a strategy

- Strategy $\sigma$
- $\text{Out}(\sigma)$ has two plays, which are both winning
Outcomes of a strategy
Outcomes of a strategy

- Strategy $\sigma$

Diagram:

- States $s_0$, $s_1$, $s_2$, $s_3$, $s_4$, and a happy face.
- Transitions between states.
- Looping transition from $s_0$ to $s_1$.
- Transition from $s_3$ to $s_4$.
- Transition from $s_4$ to $s_2$.
- Transition from $s_2$ to $s_0$.
Outcomes of a strategy

- Strategy $\sigma$
- $\text{Out}(\sigma)$ has infinitely many plays, some of them are not winning
What does it mean to win a game?

- Play $\rho = s_0s_1s_2\ldots$ is compatible with $\sigma_i$ whenever $s_j \in S_i$ implies $(s_j, s_{j+1}) = \sigma_i(s_0s_1\ldots s_j)$. We write $\text{Out}(\sigma_i)$.

- $\sigma_i$ is winning if all plays compatible with $\sigma_i$ belong to $W_i$.

What does it mean to win a game?

- Play $\rho = s_0s_1s_2\ldots$ is compatible with $\sigma_i$ whenever $s_j \in S_i$ implies $(s_j, s_{j+1}) = \sigma_i(s_0s_1\ldots s_j)$. We write $\text{Out}(\sigma_i)$.

- $\sigma_i$ is **winning** if all plays compatible with $\sigma_i$ belong to $W_i$.

---

**Martin’s determinacy theorem**

Turn-based zero-sum games are determined for Borel winning objectives: in every game, either $P_1$ or $P_2$ has a winning strategy.

---

Optimality of strategies
Optimality of strategies

Out(σ₁)
Optimality of strategies
Optimality of strategies

\[ \sigma_1 \text{ is better than } \sigma_1' \text{ whenever } \text{Out}(\sigma_1)\uparrow \subseteq \text{Out}(\sigma_1')\uparrow \]
Optimality of strategies

- $\sigma_1$ is better than $\sigma_1'$ whenever $\text{Out}(\sigma_1) \uparrow \subseteq \text{Out}(\sigma_1') \uparrow$

- $\sigma_1$ is optimal whenever it is better than any other $\sigma_1'$
Optimality of strategies

- \( \sigma_1 \) is better than \( \sigma'_1 \) whenever \( \text{Out}(\sigma_1) \uparrow \subseteq \text{Out}(\sigma'_1) \uparrow \)

- \( \sigma_1 \) is optimal whenever it is better than any other \( \sigma'_1 \)

Remark

- Optimal strategies might not exist
- If \( \sqsubseteq \) given by a payoff function, notion of \( \epsilon \)-optimal strategies
- Optimality vs subgame-optimality
Relevant questions

\[ \varphi = \text{Reach}(\text{😊}) \]
Relevant questions

- Can $P_1$ win the game, i.e. does $P_1$ have a winning strategy?
- Can $P_1$ play optimally?

$\varphi = \text{Reach}(\ 🙂)$
Relevant questions

- Can $P_1$ win the game, i.e. does $P_1$ have a winning strategy?
  - Can $P_1$ play optimally?
- Is there an effective (efficient) way of winning?

$\varphi = \text{Reach}(\text{😊})$
Relevant questions

- Can $P_1$ win the game, i.e. does $P_1$ have a winning strategy?
  - Can $P_1$ play optimally?
- Is there an effective (efficient) way of winning?
- How complex is it to win?

$$\varphi = \text{Reach} (\smiley)$$
Example: the Nim game

- Players alternate
- Each player can take one or two sticks
- The player who takes the last one wins
- $P_1$ starts
Example: the Nim game

- Players alternate
- Each player can take one or two sticks
- The player who takes the last one wins
- $P_1$ starts

![Diagram showing the Nim game with starting positions and winning conditions for $P_1$ and $P_2$.]
Example: the Nim game

- Players alternate
- Each player can take one or two sticks
- The player who takes the last one wins
- $P_1$ starts

Diagram:

Players start with 8 sticks. $P_1$ can choose 7 or 6 sticks. If $P_1$ chooses 7, $P_2$ wins. If $P_1$ chooses 6, $P_2$ also wins. If $P_1$ chooses 5 or 4, $P_2$ loses and $P_1$ wins.

$P_2$ wins if $P_1$ chooses 5 or 4, and $P_1$ wins if $P_2$ chooses 6 or 7.

18
Example: the Nim game

- Players alternate
- Each player can take one or two sticks
- The player who takes the last one wins
- \( P_1 \) starts

\[
\begin{align*}
\equiv 1 \text{ or } 2 \mod 3 \\
\equiv 0 \mod 3
\end{align*}
\]
Example: the Nim game

- Players alternate
- Each player can take one or two sticks
- The player who takes the last one wins
- $P_1$ starts

From all
- $\equiv 1 \text{ or } 2 \mod 3$
- $\equiv 0 \mod 3$
Computation of winning states in the running example
Computation of winning states in the running example
Computation of winning states in the running example
Computation of winning states in the running example
Computation of winning states in the running example
Computation of winning states in the running example

All states are winning for $P_1$
Computation of winning states in the running example

One state is not winning for $P_1$
It is winning for $P_2$
Chess game

Chess game

**Zermelo’s Theorem**

From every position, either White can force a win, or Black can force a win, or both sides can force at least a draw.

Chess game

Zermelo’s Theorem

From every position, either White can force a win, or Black can force a win, or both sides can force at least a draw.

- We don’t know what is the case for the initial position, and no winning strategy (for either of the players) is known.
Chess game

Zermelo’s Theorem

From every position, either White can force a win, or Black can force a win, or both sides can force at least a draw.

- We don’t know what is the case for the initial position, and no winning strategy (for either of the players) is known
- According to Claude Shannon, there are $10^{43}$ legit positions in chess

Hex game
Hex game

Solving the Hex game

First player has always a winning strategy.
Hex game

Solving the Hex game

First player has always a winning strategy.

- Determinacy results (no tie is possible) + strategy stealing argument
Hex game

Solving the Hex game

First player has always a winning strategy.

- Determinacy results (no tie is possible) + strategy stealing argument
- A winning strategy is not known yet.
What we do not consider

- Concurrent games
- Stochastic games and strategies
- Partial information
- Values
- Determinacy of Blackwell games
Families of strategies
Families of strategies
General strategies

\[ \sigma_i : S^i S_i \rightarrow E \]

- May use any information of the past execution
- Information used is therefore potentially infinite
- Not adequate if one targets implementation
On the simplest side: positional strategies

From $\sigma_i : S^*S_i \rightarrow E$ to $\sigma_i : S_i \rightarrow E$
On the simplest side: positional strategies

From $\sigma_i : S^* S_i \rightarrow E$ to $\sigma_i : S_i \rightarrow E$

- Positional = memoryless
On the simplest side: positional strategies

From $\sigma_i : S^*S_i \rightarrow E$ to $\sigma_i : S_i \rightarrow E$

- Positional = memoryless
- Reachability, parity, mean-payoff, positive energy, ... → positional strategies are sufficient to win
On the simplest side: positional strategies

From $\sigma_i : S^*S_i \rightarrow E$ to $\sigma_i : S_i \rightarrow E$

- Positional = memoryless
- Reachability, parity, mean-payoff, positive energy, ... 
  $\rightarrow$ positional strategies are sufficient to win
Example: mean-payoff

Example: mean-payoff

- $P_1$ maximizes, $P_2$ minimizes

\[ \overline{MP} = \limsup_n \sum_{i \neq n} \frac{c_i}{n} \]

Example: mean-payoff

- $P_1$ maximizes, $P_2$ minimizes
- Positional strategies are sufficient to win

\[
\text{MP} = \lim_{n \to \infty} \sup_n \frac{\sum_{i \neq n} c_i}{n}
\]

Do we need more?
« See infinitely often both $a$ and $b$ »

$\text{Büchi}(a) \land \text{Büchi}(b)$
Examples

« See infinitely often both $a$ and $b$ »

Büchi($a$) ∧ Büchi($b$)

Winning strategy

- At each visit to $s_1$, loop once in $s_1$ and then go to $s_2$
- At each visit to $s_2$, loop once in $s_2$ and then go to $s_1$
- Generates the sequence $(acbc)^\omega$
Examples

« See infinitely often both $a$ and $b$ »
Büchi$(a) \land$ Büchi$(b)$

Winning strategy

- At each visit to $s_1$, loop once in $s_1$ and then go to $s_2$
- At each visit to $s_2$, loop once in $s_2$ and then go to $s_1$
- Generates the sequence $(acbc)\omega$

« Reach the target with energy level 0 »
$\textbf{FG} \ (EL = 0)$
Examples

« See infinitely often both $a$ and $b$ »
$\text{Büchi}(a) \land \text{Büchi}(b)$

Winning strategy

- At each visit to $s_1$, loop once in $s_1$ and then go to $s_2$
- At each visit to $s_2$, loop once in $s_2$ and then go to $s_1$
- Generates the sequence $(acbc)^\omega$

« Reach the target with energy level 0 »
$\text{FG (EL} = 0)$

Winning strategy

- Loop five times in $s_0$
- Then go to the target
- Generates the sequence of colors $111111 - 500000...$
Examples

« See infinitely often both $a$ and $b$ »
Büchi$(a) \land$ Büchi$(b)$

Winning strategy

- At each visit to $s_1$, loop once in $s_1$ and then go to $s_2$
- At each visit to $s_2$, loop once in $s_2$ and then go to $s_1$
- Generates the sequence $(acbc)\omega$

Winning strategy

- Loop five times in $s_0$
- Then go to the target
- Generates the sequence of colors $1 1 1 1 1 1 − 5 0 0 0 0...$

« Reach the target with energy level 0 »
$FG \ (EL = 0)$

These two strategies require only finite memory
Example: multi-dimensional mean-payoff

« Have a (limsup) mean-payoff ≥ 0 on both dimensions »
So-called multi-dimensional mean-payoff
Example: multi-dimensional mean-payoff

« Have a (limsup) mean-payoff $\geq 0$ on both dimensions »
So-called multi-dimensional mean-payoff

Winning strategy

- After $k$-th switch between $s_1$ and $s_2$, loop $2k - 1$ times and then switch back
- Generates the sequence

$$
(-1, -1) (-1, +1) (-1, -1) (+1, -1) (+1, -1) (+1, -1) (-1, -1) \\
(-1, +1) (-1, +1) (-1, +1) (-1, +1) (-1, +1) (-1, -1) \\
(+1, -1) (+1, -1) (+1, -1) (+1, -1) (+1, -1) (+1, -1) (-1, -1)...
$$
Example: multi-dimensional mean-payoff

- « Have a \( (\limsup) \) mean-payoff \( \geq 0 \) on both dimensions »
- So-called multi-dimensional mean-payoff

Winning strategy

- After \( k \)-th switch between \( s_1 \) and \( s_2 \), loop \( 2k - 1 \) times and then switch back
- Generates the sequence:
  \[
  (-1, -1) (-1, +1) (-1, -1) (+1, -1) (+1, -1) (+1, -1) (-1, -1)
  (-1, +1) (-1, +1) (-1, +1) (+1, +1) (+1, +1) (-1, -1)
  (+1, -1) (+1, -1) (+1, -1) (+1, -1) (+1, -1) (+1, -1) (-1, -1)...
  \]

This strategy requires \textbf{infinite} memory, and this is unavoidable
We focus on finite memory!
Memory skeleton

\[ \mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}}) \text{ with } m_{\text{init}} \in M \text{ and } \alpha_{\text{upd}} : M \times C \to M \]
\[ \mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}}) \text{ with } m_{\text{init}} \in M \text{ and } \alpha_{\text{upd}} : M \times C \to M \]

Memory skeleton

Not yet a strategy!

\[ \sigma_i : S^* S_i \to E \]
\( \mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}}) \) with \( m_{\text{init}} \in M \) and \( \alpha_{\text{upd}} : M \times C \to M \)

**Memory skeleton**

Not yet a strategy!

\( \sigma_i : S^*_i \to E \)

**Strategy with memory \( \mathcal{M} \)**

Additional next-move function \( \alpha_{\text{next}} : M \times S_i \to E \)

\( (\mathcal{M}, \alpha_{\text{next}}) \) defines a strategy!

---

* Terminology by Kopczyński
Chromatic* memory

**Memory skeleton**

\[ \mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}}) \text{ with } m_{\text{init}} \in M \text{ and } \alpha_{\text{upd}} : M \times C \to M \]

Not yet a strategy!

\[ \sigma_i : S^* S_i \to E \]

**Strategy with memory \( \mathcal{M} \)**

Additional next-move function \( \alpha_{\text{next}} : M \times S_i \to E \)

\( (\mathcal{M}, \alpha_{\text{next}}) \) defines a strategy!

Remark: positional strategies are \( \mathcal{M}_{\text{triv}} \)-strategies, where \( \mathcal{M}_{\text{triv}} \) is

* Terminology by Kopczyński
**Chromatic* memory**

**Memory skeleton**

\[ \mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}}) \text{ with } m_{\text{init}} \in M \text{ and } \alpha_{\text{upd}} : M \times S \rightarrow M \]

**Strategy with memory \( \mathcal{M} \)**

Additional next-move function \( \alpha_{\text{next}} : M \times S_i \rightarrow E \)

\( (\mathcal{M}, \alpha_{\text{next}}) \) defines a strategy!

Remark: positional strategies are \( \mathcal{M}_{\text{triv}} \)-strategies, where \( \mathcal{M}_{\text{triv}} \) is

* Termination by Kopczyński
Example of chromatic memory

This skeleton is sufficient for the winning condition

\[ \text{Büchi}(a) \land \text{Büchi}(b) \]
Example of *chromatic memory*

This skeleton is sufficient for the winning condition

\[ \text{Büchi}(a) \land \text{Büchi}(b) \]

That is, for every game, if there is a winning strategy, there is one based on this skeleton.
Example of chromatic memory

This skeleton is sufficient for the winning condition

\[ \text{Büchi}(a) \land \text{Büchi}(b) \]

That is, for every game, if there is a winning strategy, there is one based on this skeleton.
Example of **chromatic memory**

This skeleton is sufficient for the winning condition

\[ \text{Büchi}(a) \land \text{Büchi}(b) \]

That is, for every game, if there is a winning strategy, there is one based on this skeleton

\[
\alpha_{\text{next}} : \quad M \times S_1 \quad \rightarrow \quad E
\]

\[ (m_1, s_2) \mapsto (s_2, b, s_2) \]

\[ (m_2, s_2) \mapsto (s_2, a, s_1) \]

\[ (m_\star, s_3) \mapsto (s_3, b, s_1) \]
Example of chromatic memory

This skeleton is sufficient for the winning condition
Büchi(a) ∧ Büchi(b)

That is, for every game, if there is a winning strategy, there is one based on this skeleton

\[ \alpha_{\text{next}} : M \times S_1 \rightarrow E \]

\[
\begin{align*}
(m_1, s_2) & \mapsto (s_2, c, s_3) \\
(m_2, s_2) & \mapsto (s_2, a, s_1) \\
(m_\star, s_3) & \mapsto (s_3, b, s_1)
\end{align*}
\]
Our goal

Understand well low-memory specifications
Our goal

Positional / finite-memory determinacy

Is it the case that positional (resp. finite-memory) strategies suffice to win/be optimal when winning/optimal strategies exist?

Understand well low-memory specifications
Our goal

Understand well low-memory specifications

Positional / finite-memory determinacy

Is it the case that positional (resp. finite-memory) strategies suffice to win/be optimal when winning/optimal strategies exist?
Our goal

- Understand well low-memory specifications

Positional / finite-memory determinacy

Is it the case that positional (resp. finite-memory) strategies suffice to win/be optimal when winning/optimal strategies exist?

- Finite vs infinite games
Characterizing positional and chromatic finite-memory determinacy in finite games
The approach

The approach

- Characterize winning objectives ensuring *memoryless determinacy*, that is, the existence of positional winning strategies (for both players) in all finite games

Characterize winning objectives ensuring memoryless determinacy, that is, the existence of positional winning strategies (for both players) in all finite games.

Should apply to reachability/safety objectives, mean-payoff, parity, ...
The approach

- Characterize winning objectives ensuring **memoryless determinacy**, that is, the existence of positional winning strategies (for both players) in all finite games

- Should apply to reachability/safety objectives, mean-payoff, parity, ...

- Fundamental reference: [GZ05]

Properties of preference relations
Properties of preference relations

- Let $\sqsubseteq$ be a preference relation (for $P_1$).
- Let $W \subseteq C^\omega$ be a winning objective (for $P_1$).
Properties of preference relations

- Let $\sqsubseteq$ be a preference relation (for $P_1$).
- Let $W \subseteq C^\omega$ be a winning objective (for $P_1$).
- It is said **monotone** whenever:
Properties of preference relations

- Let $\preceq$ be a preference relation (for $P_1$).
- Let $W \subseteq C^\omega$ be a winning objective (for $P_1$).

- It is said **monotone** whenever:

- It is said **selective** whenever:
Properties of preference relations

- Let $\sqsubseteq$ be a preference relation (for $P_1$).
- Let $W \subseteq C^\omega$ be a winning objective (for $P_1$).

- It is said **monotone** whenever:

  ![Monotone Diagram](Diagram)

- It is said **selective** whenever:

  ![Selective Diagram](Diagram)

If this is in $W$, then one of those is in $W$. 
Let $\sqsubseteq$ be a preference relation (for $P_1$).

**Characterization - Two-player games**

The two following assertions are equivalent:
1. All finite games have positional optimal strategies for both players;
2. Both $\sqsubseteq$ and $\sqsubseteq^{-1}$ are monotone and selective.
Two characterizations

Let $\sqsubseteq$ be a preference relation (for $P_1$).

### Characterization - Two-player games

The two following assertions are equivalent:
1. All finite games have positional optimal strategies for both players;
2. Both $\sqsubseteq$ and $\sqsubseteq^{-1}$ are monotone and selective.

### Characterization - One-player games

The two following assertions are equivalent:
1. All finite $P_1$-games have positional optimal strategies;
2. $\sqsubseteq$ is monotone and selective.
Applications

Lifting theorem

$P_i$ has positional optimal strategies in all finite $P_i$-games

$\downarrow$

Both players have positional optimal strategies in all finite 2-player games.
Applications

**Lifting theorem**

\[ P_i \] has positional optimal strategies in all finite \( P_i \)-games

\[ \downarrow \]

Both players have positional optimal strategies in all finite 2-player games.

**Very powerful and extremely useful in practice**

- Easy to analyse the one-player case (graph analysis)
  - Mean-payoff, average-energy [BMRL15]

Discussion of examples

- Reachability, safety:
  - Monotone (though not prefix-independent)
  - Selective

- Parity, mean-payoff:
  - Prefix-independent hence monotone
  - Selective

- Average-energy games [BMRL15]
  - Lifting theorem!!

Let $\succeq$ be a preference relation (for $P_1$).
Let $M$ be a memory skeleton.

- It is said $M$-monotone whenever:

- It is said $M$-selective whenever:
Properties of preference relations — Adding memory

- Let $\sqsubsetneq$ be a preference relation (for $P_1$). Let $\mathcal{M}$ be a memory skeleton.

- It is said $\mathcal{M}$-monotone whenever:

- It is said $\mathcal{M}$-selective whenever:
Properties of preference relations — Adding memory

- Let $\sqsubseteq$ be a preference relation (for $P_1$). Let $M$ be a memory skeleton.

- It is said $M$-monotone whenever:

- It is said $M$-selective whenever:
Properties of preference relations — Adding memory

- Let $\leq$ be a preference relation (for $P_1$). Let $\mathcal{M}$ be a memory skeleton.

- It is said $\mathcal{M}$-monotone whenever:

- It is said $\mathcal{M}$-selective whenever:
Properties of preference relations — Adding memory

- Let $\sqsubseteq$ be a preference relation (for $P_1$). Let $\mathcal{M}$ be a memory skeleton.

- It is said $\mathcal{M}$-monotone whenever:

- It is said $\mathcal{M}$-selective whenever:
Properties of preference relations — Adding memory

- Let $\sqsubseteq$ be a preference relation (for $P_1$).
  Let $\mathcal{M}$ be a memory skeleton.

- It is said $\mathcal{M}$-monotone whenever:

- It is said $\mathcal{M}$-selective whenever:
Let $\sqsubseteq$ be a preference relation (for $P_1$) and $\mathcal{M}$ be a memory skeleton.

**Characterization - Two-player games**

The two following assertions are equivalent:

1. All finite games have $\mathcal{M}$-based optimal strategies for both players;
2. Both $\sqsubseteq$ and $\sqsubseteq^{-1}$ are $\mathcal{M}$-monotone and $\mathcal{M}$-selective.
Let $\sqsubseteq$ be a preference relation (for $P_1$) and $\mathcal{M}$ be a memory skeleton.

**Characterization - Two-player games**

The two following assertions are equivalent:

1. All finite games have $\mathcal{M}$-based optimal strategies for both players;
2. Both $\sqsubseteq$ and $\sqsubseteq^{-1}$ are $\mathcal{M}$-monotone and $\mathcal{M}$-selective.

**Characterization - One-player games**

The two following assertions are equivalent:

1. All finite $P_1$-games have $\mathcal{M}$-based optimal strategies;
2. $\sqsubseteq$ is $\mathcal{M}$-monotone and $\mathcal{M}$-selective.
Two characterizations

Let $\sqsubseteq$ be a preference relation (for $P_1$) and $\mathcal{M}$ be a memory skeleton.

**Characterization - Two-player games**

The two following assertions are equivalent:
1. All finite games have $\mathcal{M}$-based optimal strategies for both players;
2. Both $\sqsubseteq$ and $\sqsubseteq^{-1}$ are $\mathcal{M}$-monotone and $\mathcal{M}$-selective.

**Characterization - One-player games**

The two following assertions are equivalent:
1. All finite $P_1$-games have $\mathcal{M}$-based optimal strategies;
2. $\sqsubseteq$ is $\mathcal{M}$-monotone and $\mathcal{M}$-selective.

$\rightarrow$ We recover [GZ05] with $\mathcal{M} = \mathcal{M}_{\text{triv}}$
Applications

Lifting theorem

$P_i$ has $M_i$-based optimal strategies in all finite $P_i$-games

$\Downarrow$

Both players have $(M_1 \times M_2)$-based optimal strategies in all finite two-player games.
Applications

Lifting theorem

$P_i$ has $\mathcal{M}_i$-based optimal strategies in all finite $P_i$-games

$\Downarrow$

Both players have $(\mathcal{M}_1 \times \mathcal{M}_2)$-based optimal strategies in all finite two-player games.

Very powerful and extremely useful in practice

- Easy to analyse the one-player case (graph analysis)
  - Conjunction of $\omega$-regular objectives
Example of application

\[ W = \operatorname{Reach}(a) \land \operatorname{Reach}(b) \]
Example of application

$W = \text{Reach}(a) \land \text{Reach}(b)$

$\mathcal{M}_1$

$C \setminus \{a\}$

$\xrightarrow{a} m_1 \xrightarrow{a} m_2 \xrightarrow{C}$

$\subseteq_W$ is $\mathcal{M}_1$-monotone but not $\mathcal{M}_1$-selective
Example of application

\[ W = \text{Reach}(a) \land \text{Reach}(b) \]

\[ M_1 \quad C \setminus \{a\} \]

\[ M_2 \quad C \setminus \{a, b\} \]

\[ \subseteq_W \text{ is } M_1\text{-monotone but not } M_1\text{-selective} \]
Example of application

\[ W = \text{Reach}(a) \land \text{Reach}(b) \]

\[ M_1 \quad C \setminus \{a\} \quad a \quad m_1 \quad m_2 \quad C \]

\[ M_2 \quad C \setminus \{a, b\} \quad a \quad m_1 \quad m_2 \quad C \]

\[ C \setminus \{a\} \]

\[ \subseteq_W \text{ is } M_1\text{-monotone but not } M_1\text{-selective} \]

\[ \subseteq_W \text{ is } M_2\text{-selective} \]
Example of application

\[ W = \text{Reach}(a) \land \text{Reach}(b) \]

\( \mathcal{M}_1 \)

\[ C \setminus \{a\} \quad \xymatrix{ m_1 \ar[r]^{a} & m_2 \ar[r] & C } \]

\( \mathcal{M}_2 \)

\[ C \setminus \{a, b\} \quad \xymatrix{ m_1 \ar[r]^{a} & m_2 \ar[r] & C \quad \text{\(\equiv_W\) is \(\mathcal{M}_1\)-monotone but not \(\mathcal{M}_1\)-selective} \]

\[ m_3 \quad \xymatrix{ \quad \ar[r]^{b} & m_2 \ar[r]^{a} & m_1 \ar[r]^{a} & \quad \}

\[ C \setminus \{a\} \quad \text{\(\equiv_W\) is \(\mathcal{M}_2\)-selective} \]

\[ \square \quad \begin{align*} & \equiv_W \text{ is } \mathcal{M}_1\text{-monotone and } \mathcal{M}_2\text{-selective} \quad \mid \quad \equiv_W^{-1} \text{ is } \mathcal{M}_1\text{-monotone and } \mathcal{M}_{\text{triv}}\text{-selective} \end{align*} \]
Example of application

\[ W = \text{Reach}(a) \land \text{Reach}(b) \]

\[ \mathcal{M}_1 \]

\[ C \setminus \{a\} \]

\[ m_1 \xrightarrow{a} m_2 \]

\[ C \]

\[ \equiv_W \text{ is } \mathcal{M}_1\text{-monotone but not } \mathcal{M}_1\text{-selective} \]

\[ \mathcal{M}_2 \]

\[ C \setminus \{a, b\} \]

\[ m_1 \xrightarrow{a} m_2 \]

\[ b \]

\[ m_3 \xrightarrow{a} m_2 \]

\[ C \setminus \{a\} \]

\[ \equiv_W \text{ is } \mathcal{M}_2\text{-selective} \]

\[ \equiv_W \text{ is } \mathcal{M}_1\text{-monotone and } \mathcal{M}_2\text{-selective} \]

\[ \equiv_W^{-1} \text{ is } \mathcal{M}_1\text{-monotone and } \mathcal{M}_{\text{triv}}\text{-selective} \]

→ Memory \( \mathcal{M}_2 \) is sufficient for both players in all finite games

43
Partial conclusion

Finite games
Partial conclusion

- Complete characterization of winning objectives (and even preference relations) that ensure **chromatic** finite-memory determinacy for both players
Partial conclusion

- Complete characterization of winning objectives (and even preference relations) that ensure chromatic finite-memory determinacy for both players

- One-to-two-player lifts
  (requires chromatic finite memory determinacy in one-player games for both players; ensures chromatic finite memory determinacy in two-players games for both players)
Complete characterization of winning objectives (and even preference relations) that ensure chromatic finite-memory determinacy for both players

One-to-two-player lifts
(requires chromatic finite memory determinacy in one-player games for both players; ensures chromatic finite memory determinacy in two-players games for both players)

Further questions:
- Can we reduce/optimize the memory?
- What about chaotic finite memory?
- Can we focus on one player (so-called half-positionality)?
Characterizing positional and chromatic finite-memory determinacy in infinite games
The case of mean-payoff

- Objective for $P_1$: get non-negative (limsup) mean-payoff
- In finite games: *positional* strategies are sufficient to win
- In infinite games: *infinite memory* is required to win
Let $W$ be a prefix-independent objective.

[CN06] Colcombet and Niwiński. On the positional determinacy of edge-labeled games (ICALP’06).
Let $W$ be a prefix-independent objective.

**Characterization - Two-player games**

The two following assertions are equivalent:

1. Positional optimal strategies are sufficient for $W$ in all (infinite) games for both players;

2. $W$ is a parity condition
   That is, there are $n \in \mathbb{N}$ and $\gamma : C \to \{0,1,\ldots,n\}$ such that
   
   $W = \{ c_1c_2\ldots \in C^\omega \mid \limsup_{i} \gamma(c_i) \text{ is even} \}$

---

[CN06] Colcombet and Niwiński. On the positional determinacy of edge-labeled games (ICALP'06).

Let $W$ be a prefix-independent objective.

**Characterization - Two-player games**

The two following assertions are equivalent:

1. Positional optimal strategies are sufficient for $W$ in all (infinite) games for both players;

2. $W$ is a parity condition
   That is, there are $n \in \mathbb{N}$ and $\gamma : C \rightarrow \{0,1,\ldots,n\}$ such that
   \[ W = \{ c_1c_2\ldots \in C^\omega \mid \limsup_{i} \gamma(c_i) \text{ is even} \} \]

---

[CN06] Colcombet and Niwiński. On the positional determinacy of edge-labeled games (ICALP'06).

Let $W$ be a prefix-independent objective.

**Characterization - Two-player games**

The two following assertions are equivalent:

1. **Positional** optimal strategies are sufficient for $W$ in all (infinite) games for both players;

2. $W$ is a parity condition
   That is, there are $n \in \mathbb{N}$ and $\gamma : C \rightarrow \{0,1,\ldots,n\}$ such that
   $$W = \{ c_1 c_2 \ldots \in C^\omega \mid \limsup_{i} \gamma(c_i) \text{ is even} \}$$

[CN06] Colcombet and Niwiński. On the positional determinacy of edge-labeled games (ICALP'06).

Some language theory (1)

- Let $L \subseteq C^*$ be a language of finite words

**Right congruence**

- Given $x, y \in C^*$,

$$x \sim_L y \iff \forall z \in C^*, \left( x \cdot z \in L \iff y \cdot z \in L \right)$$
Some language theory (1)

- Let $L \subseteq C^*$ be a language of finite words

**Right congruence**

- Given $x, y \in C^*$,

$$x \sim_L y \iff \forall z \in C^*, (x \cdot z \in L \iff y \cdot z \in L)$$

**Myhill-Nerode Theorem**

- $L$ is regular if and only if $\sim_L$ has finite index;
  - There is an automaton whose states are classes of $\sim_L$, which recognizes $L$. 


Some language theory (2)

- Let $L \subseteq C^\omega$ be a language of infinite words

**Right congruence**

- Given $x, y \in C^*$,

  $$x \sim_L y \iff \forall z \in C^\omega, \left( x \cdot z \in L \iff y \cdot z \in L \right)$$
Some language theory (2)

- Let $L \subseteq C^\omega$ be a language of infinite words

**Right congruence**

- Given $x, y \in C^*$,

  $$x \sim_L y \iff \forall z \in C^\omega, (x \cdot z \in L \iff y \cdot z \in L)$$

**Link with $\omega$-regularity?**

- If $L$ is $\omega$-regular, then $\sim_L$ has finite index;
  - The automaton based on $\sim_L$ is a so-called prefix-classifier;
- The converse does not hold (e.g. all prefix-independent languages are such that $\sim_L$ has only one element).
Four examples

<table>
<thead>
<tr>
<th>Objective</th>
<th>Prefix classifier $\mathcal{M}_\omega$</th>
<th>Sufficient memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parity objective</td>
<td>$\rightarrow\diamondsuit \ C$</td>
<td>$\rightarrow\diamondsuit \ C$</td>
</tr>
<tr>
<td>Mean-payoff $\geq 0$</td>
<td>$\rightarrow\diamondsuit \ C$</td>
<td>No finite automaton</td>
</tr>
<tr>
<td>$C = {a, b}$</td>
<td>$\rightarrow\diamondsuit \ C$</td>
<td></td>
</tr>
<tr>
<td>$W = b^*ab^*aC^\omega$</td>
<td>$\rightarrow\diamondsuit \ C$</td>
<td></td>
</tr>
<tr>
<td>$C = {a, b}$</td>
<td>$\rightarrow\diamondsuit \ C$</td>
<td></td>
</tr>
<tr>
<td>$W = C^*(ab)^\omega$</td>
<td>$\rightarrow\diamondsuit \ C$</td>
<td></td>
</tr>
</tbody>
</table>
Let $W \subseteq C^\omega$ be a winning objective.

If a finite memory structure $\mathcal{M}$ suffices to play optimally in one-player infinite arenas for both players, then the prefix-classifier $\mathcal{M}_\sim$ is finite and $W$ is recognized by a parity automaton $(\mathcal{M}_\sim \otimes \mathcal{M}, \gamma)$, with $\gamma: M \times C \rightarrow \{0,1,\ldots,n\}$.

→ Generalizes [CN06] where both $\mathcal{M}$ and $\mathcal{M}_\sim$ are trivial

<table>
<thead>
<tr>
<th>Objective</th>
<th>Prefix classifier $\mathcal{M}$</th>
<th>One-player memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parity objective</td>
<td>$\Diamond C$</td>
<td>$\Diamond C \leftrightarrow {0,1,\ldots,n}$</td>
</tr>
<tr>
<td>Mean-payoff $\geq 0$</td>
<td>$\Diamond C$</td>
<td>No finite automaton</td>
</tr>
<tr>
<td>$C = {a, b}$ $W = b^*ab^*aC^\omega$</td>
<td>$\Diamond a \rightarrow 1 \rightarrow a \rightarrow b \rightarrow 1 \rightarrow b \rightarrow 1 \rightarrow a \rightarrow b \rightarrow c \rightarrow C$</td>
<td>$\Diamond C$</td>
</tr>
<tr>
<td>$C = {a, b}$ $W = C^*(ab)^\omega$</td>
<td>$\Diamond C$</td>
<td>$1 b \rightarrow a 0 \rightarrow b 0 \rightarrow a 1$</td>
</tr>
</tbody>
</table>
Corollaries

Lifting theorem

If $W$ and $W^c$ are finite-memory-determined in one-player infinite games, then $W$ and $W^c$ are finite-memory-determined in two-player infinite games.
Corollaries

Lifting theorem

If $W$ and $W^c$ are finite-memory-determined in one-player infinite games, then $W$ and $W^c$ are finite-memory-determined in two-player infinite games.

Characterization

$W$ is finite-memory-determined in (two-player) infinite games if and only if $W$ is $\omega$-regular.
Some consequences

- Mean-payoff $\geq 0$ is not $\omega$-regular (even though it is positionally determined in finite games)

- Some discounted objectives are $\omega$-regular:
  
  e.g. condition $\text{DS}_{\frac{1}{\lambda}}^{\geq 0}$ (with $\lambda \in (0,1) \cap \mathbb{Q}$, $C = [-k, k] \cap \mathbb{Z}$) is $\omega$-regular if and only if $k < \frac{1}{\lambda} - 1$ or $\lambda = \frac{1}{n}$ for some $n \in \mathbb{N}_{>0}$
Partial conclusion

Infinite games
Partial conclusion

- Complete characterization of winning objectives that ensure chromatic finite-memory determinacy in infinite games = $\omega$-regular
Partial conclusion

- Complete characterization of winning objectives that ensure chromatic finite-memory determinacy in infinite games = $\omega$-regular

- One-to-two-player lift
  (requires chromatic finite memory determinacy in one-player games for both players; ensures chromatic finite memory determinacy in two-players games for both players)
Complete characterization of winning objectives that ensure chromatic finite-memory determinacy in infinite games = $\omega$-regular

One-to-two-player lift
(requires chromatic finite memory determinacy in one-player games for both players; ensures chromatic finite memory determinacy in two-players games for both players)

Further questions:
• Can be reduce/optimize the memory?
  E.g. is $\mathcal{M}_\sim$ necessary in the memory for two players?
• What about chaotic finite memory?
• Can we focus on one player (so-called half-positionality)?
• What about finite branching?
Conclusion
What you can bring home
What you can bring home

- Use of models and **concepts from game theory** in formal methods (e.g. controller in reactive systems)
What you can bring home

- Use of models and **concepts from game theory** in formal methods (e.g. controller in reactive systems)

- These concepts (like winning strategies) require manipulating information
  - For simpler strategies, use **low memory**!
  - ... even though low memory does not mean it is easy...
What you can bring home

- Use of models and concepts from game theory in formal methods (e.g. controller in reactive systems)

- These concepts (like winning strategies) require manipulating information
  - For simpler strategies, use low memory!
  - ... even though low memory does not mean it is easy...

- Understand chromatic finite-memory determined objectives
Use of models and concepts from game theory in formal methods (e.g. controller in reactive systems)

These concepts (like winning strategies) require manipulating information
- For simpler strategies, use low memory!
- ... even though low memory does not mean it is easy...

Understand chromatic finite-memory determined objectives

Going further:
- Games under partial observation, e.g. players with their own knowledge (of the game, of the other’s choices, ...)
- Half-positionality or half-finite-memory of objectives (preliminary result [BCRV22])