Characterizing Omega-Regularity through Strategy Complexity of Zero-Sum Games

Pierre Vandenhove

LaBRI, Université de Bordeaux

December 20, 2023 – BMS Young Scholar Day

université de BORDEAUX

LABORATOIRE BORDELAIS DE RECHERCHE EN INFORMATIQUE LaBRI

Overview

Goal

We want to understand **languages of (in)finite words**.
$\rightsquigarrow$ Connections to **logic** and **automata**.

Motivation

Representations for languages of **infinite** words are not well understood.

Result

Given a language, establish a connection between **automata** on infinite words and a **game-theoretic property**.

Plan:
I. Automata, **II**. Games, **III**. Connection between automata and games

I. Automata

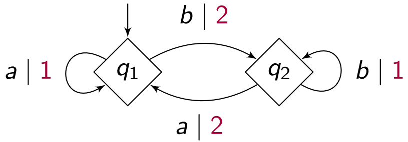
Automata on infinite words

We want to study languages of **infinite** words $\rightsquigarrow$ suitable kind of automata?

Deterministic parity automaton

A **deterministic parity automaton** is a tuple $\mathcal{P} = (Q, \Sigma, q_{\mathsf{init}}, \delta, p)$ where

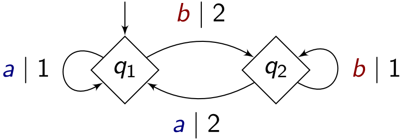
- Q is a finite set of states,
- Σ is an alphabet,
- $q_{\mathsf{init}} \in Q$ is an initial state,
- $\delta \colon Q \times \Sigma \to Q$ is a transition function,
- $p \colon Q \times \Sigma \to \{0, \dots, n\}$ associates an **integer** to transitions.



Acceptance condition

An **infinite** word is accepted if
the largest integer seen infinitely often is even.

Example, $\Sigma = \{a, b\}$:

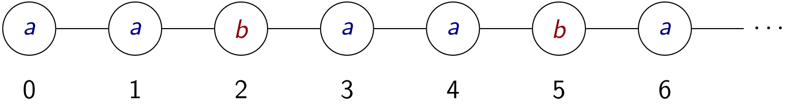


- Word $aabaabaab\ldots = (aab)^\omega \rightsquigarrow 112212212\ldots = 112(212)^\omega$. ✔
- Word $abaaa\ldots = aba^\omega \rightsquigarrow 12211\ldots = 1221^\omega$. ✗
- ...

$L = \{w \in \Sigma^\omega \mid a \text{ is seen } \infty\text{ly often and } b \text{ is seen } \infty\text{ly often along } w\}$

Historical side note #1

Infinite words are structures of the form $\langle \mathbb{N}, 0, \mathrm{succ}, <, (P_a)_{a \in \Sigma} \rangle$.



Connecting automata and logic [Büchi, 1962] [Emerson, Jutla, 1991]

A language $L \subseteq \Sigma^\omega$ is recognized by a **deterministic parity automaton**
if and only if
it is definable in *monadic* second-order logic.

Corollary

This monadic second-order theory is decidable.

Automata representation

ω-regularity

These languages of infinite words are called **ω-regular**.

Multiple mysteries remain about automata for ω-regular languages.

Open questions

- Characterizing the smallest deterministic automata for a language?
- How to **minimize** an automaton (complexity)?

However, languages of **finite** words are well-understood.

Case of finite words

Let $K \subseteq \Sigma^*$ be a language of **finite** words.

Myhill-Nerode congruence

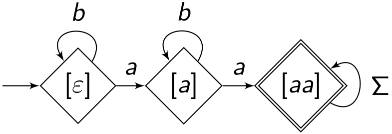
For $x, y \in \Sigma^*$, $x \sim_K y$ if for all $z \in \Sigma^*$, $xz \in K \iff yz \in K$.

I.e., x and y have the same accepting continuations in K.

Myhill-Nerode theorem [Nerode, 1958]

- K is **regular** iff $\sim_K$ has **finitely many equivalence classes**.
- The **equivalence classes** of $\sim_K$ are the **states of the minimal deterministic automaton for K**.

Example:
$K =$ "a at least twice"



Case of infinite words

Let $L \subseteq \Sigma^\omega$ be a language of **infinite** words.

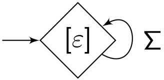
(Almost) Myhill-Nerode congruence

For $x, y \in \Sigma^*$, $x \sim_L y$ if for all $z \in \Sigma^\omega$, $xz \in L \iff yz \in L$.

No Myhill-Nerode theorem

- If L is ω-**regular**, then $\sim_L$ has finitely many equivalence classes.
- The converse is false!

$L =$ "a and b ∞ly often"
A **single** equivalence class!

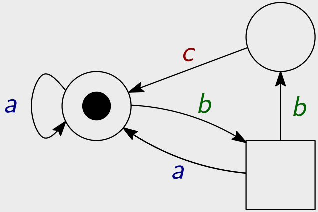


Still a "**prefix classifier**" automaton, but not informative enough to recognize the language. . .

II. Games

Games

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



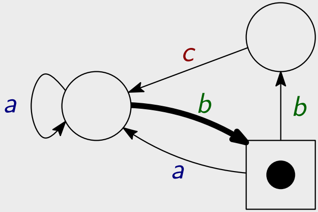
- **Alphabet** Σ, **arena** $\mathcal{A} = (V_1, V_2, E)$.
- Two **players** $\mathcal{P}_1$ ($\bigcirc$) and $\mathcal{P}_2$ ($\square$).

- **Objective** of $\mathcal{P}_1$ is a language $L \subseteq \Sigma^\omega$.
- **Zero-sum**: objective of $\mathcal{P}_2$ is $\Sigma^\omega \setminus L$.

Can $\mathcal{P}_1$ obtain a word in L, no matter what $\mathcal{P}_2$ does?

Games

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

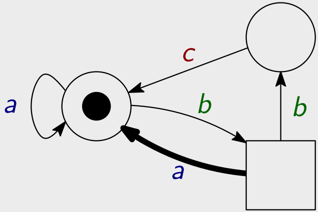


- **Alphabet** Σ, **arena** $\mathcal{A} = (V_1, V_2, E)$.
- Two **players** $\mathcal{P}_1$ ($\bigcirc$) and $\mathcal{P}_2$ ($\Box$).
  Infinite interaction
  $\rightsquigarrow$ **infinite word** $w = b$
- **Objective** of $\mathcal{P}_1$ is a language $L \subseteq \Sigma^\omega$.
- **Zero-sum**: objective of $\mathcal{P}_2$ is $\Sigma^\omega \setminus L$.

Can $\mathcal{P}_1$ obtain a word in L, no matter what $\mathcal{P}_2$ does?

Games

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

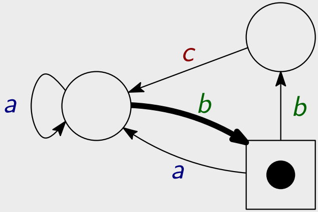


- **Alphabet** Σ, **arena** $\mathcal{A} = (V_1, V_2, E)$.
- Two **players** $\mathcal{P}_1$ ($\bigcirc$) and $\mathcal{P}_2$ ($\square$).
  Infinite interaction
  $\rightsquigarrow$ **infinite word** $w = ba$
- **Objective** of $\mathcal{P}_1$ is a language $L \subseteq \Sigma^\omega$.
- **Zero-sum**: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

Can $\mathcal{P}_1$ obtain a word in L, no matter what $\mathcal{P}_2$ does?

Games

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

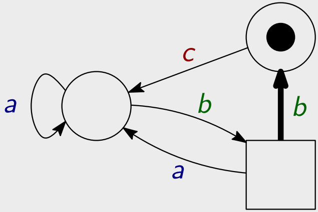


- **Alphabet** Σ, **arena** $\mathcal{A} = (V_1, V_2, E)$.
- Two **players** $\mathcal{P}_1$ ($\bigcirc$) and $\mathcal{P}_2$ ($\Box$).
  Infinite interaction
  $\rightsquigarrow$ **infinite word** $w = bab$
- **Objective** of $\mathcal{P}_1$ is a language $L \subseteq \Sigma^\omega$.
- **Zero-sum**: objective of $\mathcal{P}_2$ is $\Sigma^\omega \setminus L$.

Can $\mathcal{P}_1$ obtain a word in L, no matter what $\mathcal{P}_2$ does?

Games

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

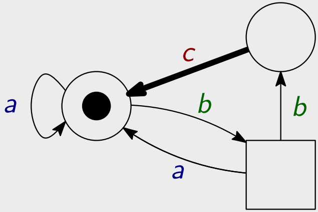


- **Alphabet** Σ, **arena** $\mathcal{A} = (V_1, V_2, E)$.
- Two **players** $\mathcal{P}_1$ ($\bigcirc$) and $\mathcal{P}_2$ ($\square$).
  Infinite interaction
  $\rightsquigarrow$ **infinite word** $w = babb$
- **Objective** of $\mathcal{P}_1$ is a language $L \subseteq \Sigma^\omega$.
- **Zero-sum**: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

Can $\mathcal{P}_1$ obtain a word in L, no matter what $\mathcal{P}_2$ does?

Games

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

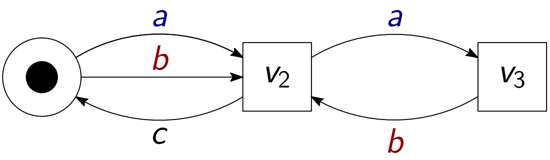


- **Alphabet** Σ, **arena** $\mathcal{A} = (V_1, V_2, E)$.
- Two **players** $\mathcal{P}_1$ ($\bigcirc$) and $\mathcal{P}_2$ ($\square$).
  Infinite interaction
  $\rightsquigarrow$ **infinite word** $w = babbc\ldots \in \Sigma^\omega$.
- **Objective** of $\mathcal{P}_1$ is a language $L \subseteq \Sigma^\omega$.
- **Zero-sum**: objective of $\mathcal{P}_2$ is $\Sigma^\omega \setminus L$.

Can $\mathcal{P}_1$ obtain a word in L, no matter what $\mathcal{P}_2$ does?

Example (1/3)

$$\Sigma = \{a, b, c\},$$
$$L = \{w \in \Sigma^\omega \mid a \text{ is seen } \infty\text{ly often and } b \text{ is seen } \infty\text{ly often}\}$$



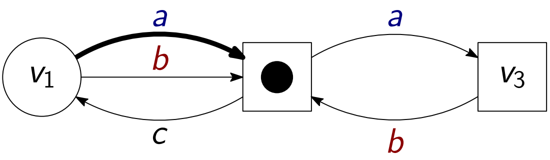
$\mathcal{P}_1$ has a winning strategy from every vertex.

Strategy representation

How to represent the strategy?
In general, strategies may not have a finite representation.
But here, a **finite memory with two memory states** suffices!

Example (1/3)

$$\Sigma = \{a, b, c\},$$
$$L = \{w \in \Sigma^\omega \mid a \text{ is seen } \infty\text{ly often and } b \text{ is seen } \infty\text{ly often}\}$$



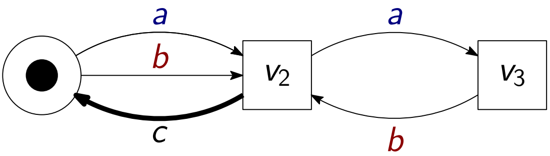
$\mathcal{P}_1$ has a winning strategy from every vertex.

Strategy representation

How to represent the strategy?
In general, strategies may not have a finite representation.
But here, a **finite memory with two memory states** suffices!

Example (1/3)

$$\Sigma = \{a, b, c\},$$
$$L = \{w \in \Sigma^\omega \mid a \text{ is seen } \infty\text{ly often and } b \text{ is seen } \infty\text{ly often}\}$$



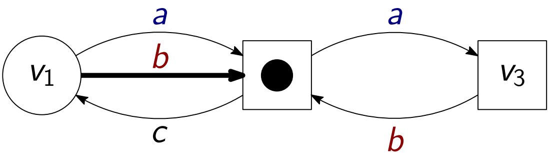
$\mathcal{P}_1$ has a winning strategy from every vertex.

Strategy representation

How to represent the strategy?
In general, strategies may not have a finite representation.
But here, a **finite memory with two memory states** suffices!

Example (1/3)

$$\Sigma = \{a, b, c\},$$
$$L = \{w \in \Sigma^\omega \mid a \text{ is seen } \infty\text{ly often and } b \text{ is seen } \infty\text{ly often}\}$$



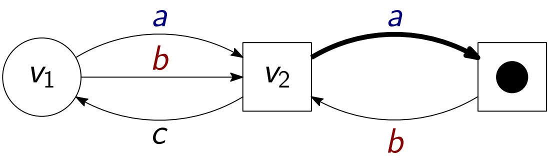
$\mathcal{P}_1$ has a winning strategy from every vertex.

Strategy representation

How to represent the strategy?
In general, strategies may not have a finite representation.
But here, a **finite memory with two memory states** suffices!

Example (1/3)

$$\Sigma = \{a, b, c\},$$
$$L = \{w \in \Sigma^\omega \mid a \text{ is seen } \infty\text{ly often and } b \text{ is seen } \infty\text{ly often}\}$$



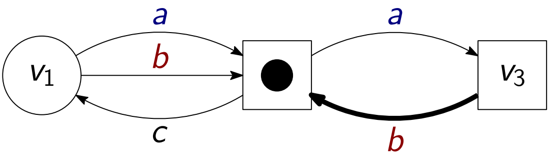
$\mathcal{P}_1$ has a winning strategy from every vertex.

Strategy representation

How to represent the strategy?
In general, strategies may not have a finite representation.
But here, a **finite memory with two memory states** suffices!

Example (1/3)

$$\Sigma = \{a, b, c\},$$
$$L = \{w \in \Sigma^\omega \mid a \text{ is seen } \infty\text{ly often and } b \text{ is seen } \infty\text{ly often}\}$$



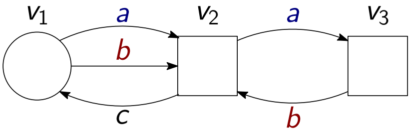
$\mathcal{P}_1$ has a winning strategy from every vertex.

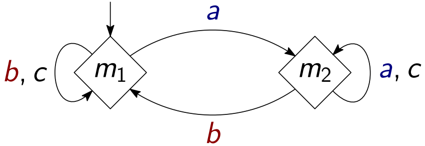
Strategy representation

How to represent the strategy?
In general, strategies may not have a finite representation.
But here, a **finite memory with two memory states** suffices!

Example (2/3): two memory states

$$\Sigma = \{a, b, c\},$$
$$L = \{w \in \Sigma^\omega \mid a \text{ is seen } \infty\text{ly often and } b \text{ is seen } \infty\text{ly often}\}$$

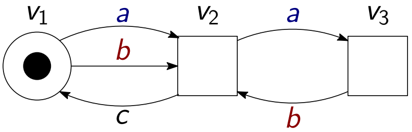




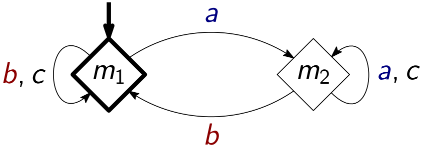
There is a winning strategy $\sigma\colon V_1 \times M \to E$.

Example (2/3): two memory states

$$\Sigma = \{a, b, c\},$$
$$L = \{w \in \Sigma^\omega \mid a \text{ is seen } \infty\text{ly often and } b \text{ is seen } \infty\text{ly often}\}$$



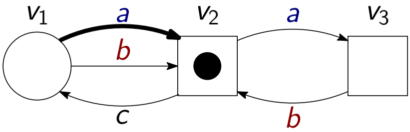
$$\sigma(v_1, m_1) = \xrightarrow{a} v_2$$



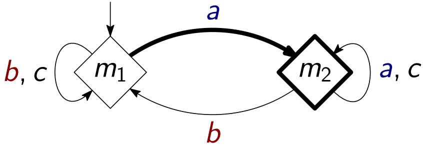
There is a winning strategy $\sigma\colon V_1 \times M \to E$.

Example (2/3): two memory states

$$\Sigma = \{a, b, c\},$$
$$L = \{w \in \Sigma^\omega \mid a \text{ is seen } \infty\text{ly often and } b \text{ is seen } \infty\text{ly often}\}$$



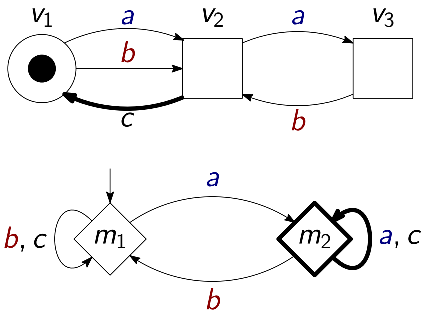
$\sigma(v_1, m_1) = \xrightarrow{a} v_2$



There is a winning strategy $\sigma\colon V_1 \times M \to E$.

Example (2/3): two memory states

$$\Sigma = \{a, b, c\},$$
$$L = \{w \in \Sigma^\omega \mid a \text{ is seen } \infty\text{ly often and } b \text{ is seen } \infty\text{ly often}\}$$

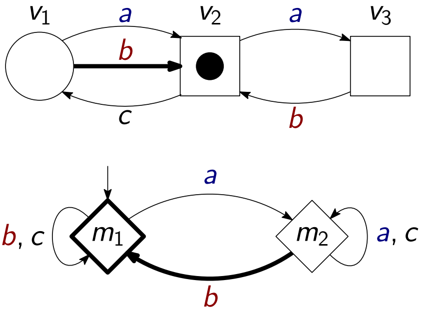


$$\sigma(v_1, m_1) = \xrightarrow{a} v_2$$
$$\sigma(v_1, m_2) = \xrightarrow{b} v_2$$

There is a winning strategy $\sigma\colon V_1 \times M \to E$.

Example (2/3): two memory states

$$\Sigma = \{a, b, c\},$$
$$L = \{w \in \Sigma^\omega \mid a \text{ is seen } \infty\text{ly often and } b \text{ is seen } \infty\text{ly often}\}$$

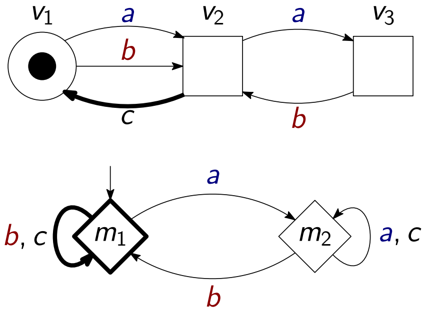


$$\sigma(v_1, m_1) = \xrightarrow{a} v_2$$
$$\sigma(v_1, m_2) = \xrightarrow{b} v_2$$

There is a winning strategy $\sigma\colon V_1 \times M \to E$.

Example (2/3): two memory states

$$\Sigma = \{a, b, c\},$$
$$L = \{w \in \Sigma^\omega \mid a \text{ is seen } \infty\text{ly often and } b \text{ is seen } \infty\text{ly often}\}$$



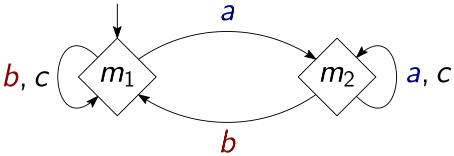
$$\sigma(v_1, m_1) = \xrightarrow{a} v_2$$
$$\sigma(v_1, m_2) = \xrightarrow{b} v_2$$

There is a winning strategy $\sigma \colon V_1 \times M \to E$.

Example (3/3)

$$\Sigma = \{a, b, c\},$$
$$L = \{w \in \Sigma^\omega \mid a \text{ is seen } \infty\text{ly often and } b \text{ is seen } \infty\text{ly often}\}$$

More generally, this memory structure suffices in **all** games with objective L!



$\rightsquigarrow$ We say that L is **finite-memory determined**.

Finite-memory determinacy

Finite-memory determinacy

A **language** is **finite-memory determined** if
there exists a finite **memory structure** $\mathcal{M}$ such that in all games,
one of the players has a **winning strategy that uses memory $\mathcal{M}$**.

Theorem [Gurevich, Harrington, 1982]

All ω-**regular languages** are **finite-memory determined**.

Historical side note #2 [Rabin, 1969]

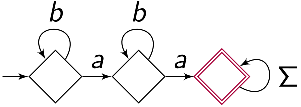
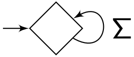
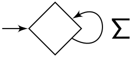
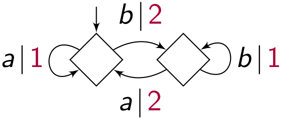
Used to show that the MSO theory of the complete binary tree is decidable.

III. Connection between automata and games

Two examples

Defined two objects: **prefix classifiers** and **memory structures**.

Let $\Sigma = \{a, b\}$.

Language	**Prefix classifier $\mathcal{S}_L$**	**Sufficient memory**
$L =$ "a at least twice"		
$L =$ "a and b ∞ly often"		

From memory to automaton

Let $L \subseteq \Sigma^\omega$.

Theorem [Bouyer, Randour, V., 2023]

If L is **finite-memory determined** with memory structure $\mathcal{M}$,

then L **is recognized by a *parity automaton*** $(\mathcal{S}_L \otimes \mathcal{M}, p)$.

In particular,

L is **finite-memory determined** over all arenas

$$\Longleftrightarrow$$

L is ω-**regular**.

Conclusion

Summary

- **Strategic characterization** of ω-regular languages.
- **"Myhill-Nerode-like" theorem** for languages of infinite words.

Remaining questions

- Characterize **minimal memory structures** for ω-regular objectives?
- Use this characterization to better understand and **minimize** deterministic parity automata?

Thanks!