Games and complexity:
from Banach–Mazur to automata theory

Michał Skrzypczak

Workshop on Wadge Theory and Automata II

Torino 08.06.2018

Foundation for Polish Science

UNIVERSITY OF WARSAW

NATIONAL SCIENCE CENTRE POLAND

Part 1

Generic objects

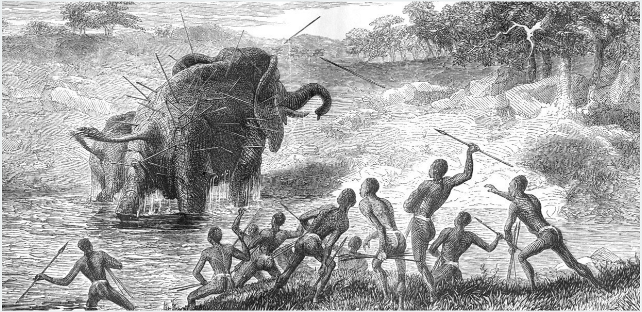
How to prove that **there exists a four-legged elephant**?

How to prove that **there exists a four-legged elephant**?

Option 1.: Find one.

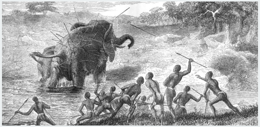
How to prove that **there exists a four-legged elephant**?

Option 1.: Find one.



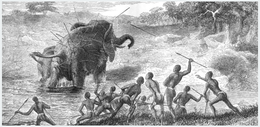
How to prove that **there exists a four-legged elephant**?

Option 1.: Find one.



How to prove that **there exists a four-legged elephant**?

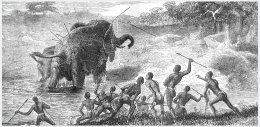
Option 1.: Find one.



Option 2.: Prove that being four legged is a **generic property**.

How to prove that **there exists a four-legged elephant**?

Option 1.: Find one.

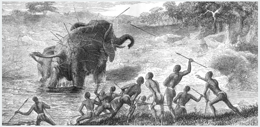


Option 2.: Prove that being four legged is a **generic property**.

Option 3.: Go contrapositive, etc...

How to prove that **there exists a four-legged elephant**?

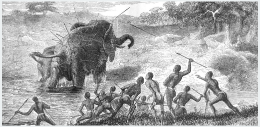
Option 1.: Find one.



Option 2.: Prove that being four legged is a **generic property**.

How to prove that **there exists a four-legged elephant**?

Option 1.: Find one.

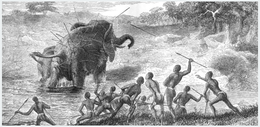


Option 2.: Prove that being four legged is a **generic property**.

Generic sets should form a σ-filter:

How to prove that **there exists a four-legged elephant**?

Option 1.: Find one.



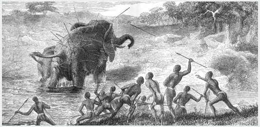
Option 2.: Prove that being four legged is a **generic property**.

Generic sets should form a σ-filter:

- If P is generic then $P \neq \emptyset$.

How to prove that **there exists a four-legged elephant**?

Option 1.: Find one.



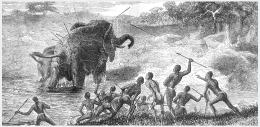
Option 2.: Prove that being four legged is a **generic property**.

Generic sets should form a σ-filter:

- If P is generic then $P \neq \emptyset$.
- If $P \subseteq P'$ is generic then P' is generic.

How to prove that **there exists a four-legged elephant**?

Option 1.: Find one.



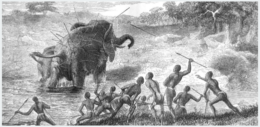
Option 2.: Prove that being four legged is a **generic property**.

Generic sets should form a σ-filter:

- If P is generic then $P \neq \emptyset$.
- If $P \subseteq P'$ is generic then P' is generic.
- If $(P_n)_{n\in\omega}$ are all generic then $\bigcap_{n\in\omega} P_n$ is generic.

How to prove that **there exists a four-legged elephant**?

Option 1.: Find one.



Option 2.: Prove that being four legged is a **generic property**.

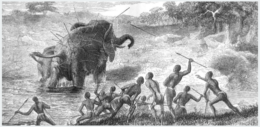
Generic sets should form a σ-filter:

- If P is generic then $P \neq \emptyset$.
- If $P \subseteq P'$ is generic then P' is generic.
- If $(P_n)_{n\in\omega}$ are all generic then $\bigcap_{n\in\omega} P_n$ is generic.

Example Probabilistic approach:

How to prove that **there exists a four-legged elephant**?

Option 1.: Find one.



Option 2.: Prove that being four legged is a **generic property**.

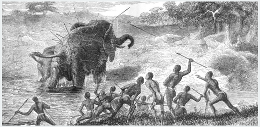
Generic sets should form a σ-filter:

- If P is generic then $P \neq \emptyset$.
- If $P \subseteq P'$ is generic then P' is generic.
- If $(P_n)_{n\in\omega}$ are all generic then $\bigcap_{n\in\omega} P_n$ is generic.

Example Probabilistic approach: P is generic iff $\mathbb{P}(P) = 1$

How to prove that **there exists a four-legged elephant**?

Option 1.: Find one.



Option 2.: Prove that being four legged is a **generic property**.

Generic sets should form a σ-filter:

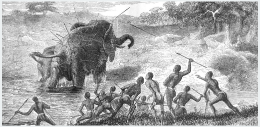
- If P is generic then $P \neq \emptyset$.
- If $P \subseteq P'$ is generic then P' is generic.
- If $(P_n)_{n\in\omega}$ are all generic then $\bigcap_{n\in\omega} P_n$ is generic.

Example Probabilistic approach: P is generic iff $\mathbb{P}(P) = 1$

$$\forall_{n\in\omega} \left(\mathbb{P}(P_n) = 1\right) \implies \mathbb{P}\left(\bigcap_{n\in\omega} P_n\right) = 1$$

How to prove that **there exists a four-legged elephant**?

Option 1.: Find one.



Option 2.: Prove that being four legged is a **generic property**.

Generic sets should form a σ-filter:

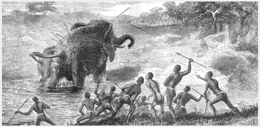
- If P is generic then $P \neq \emptyset$.
- If $P \subseteq P'$ is generic then P' is generic.
- If $(P_n)_{n\in\omega}$ are all generic then $\bigcap_{n\in\omega} P_n$ is generic.

Example Probabilistic approach: P is generic iff $\mathbb{P}(P) = 1$

$$\forall_{n\in\omega} \left(\mathbb{P}(P_n) = 1\right) \implies \mathbb{P}\left(\bigcap_{n\in\omega} P_n\right) = 1$$

How to prove that **there exists a four-legged elephant**?

Option 1.: Find one.



Option 2.: Prove that being four legged is a **generic property**.

Generic sets should form a σ-filter:

- If P is generic then $P \neq \varnothing$.
- If $P \subseteq P'$ is generic then P' is generic.
- If $(P_n)_{n\in\omega}$ are all generic then $\bigcap_{n\in\omega} P_n$ is generic.

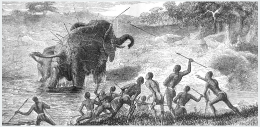
Example Probabilistic approach: P is generic iff $\mathbb{P}(P) = 1$

$$\forall_{n\in\omega} \left(\mathbb{P}(P_n) = 1\right) \implies \mathbb{P}\left(\bigcap_{n\in\omega} P_n\right) = 1$$

But:

How to prove that **there exists a four-legged elephant**?

Option 1.: Find one.



Option 2.: Prove that being four legged is a **generic property**.

Generic sets should form a σ-filter:

- If P is generic then $P \neq \emptyset$.
- If $P \subseteq P'$ is generic then P' is generic.
- If $(P_n)_{n\in\omega}$ are all generic then $\bigcap_{n\in\omega} P_n$ is generic.

Example Probabilistic approach: P is generic iff $\mathbb{P}(P) = 1$

$$\forall_{n\in\omega} \left(\mathbb{P}(P_n) = 1\right) \implies \mathbb{P}\left(\bigcap_{n\in\omega} P_n\right) = 1$$

But: limitations of **quantitativity**

Topological genericness: **comeagre sets**

Topological genericness: **comeagre sets**

$G \subseteq X$ is comeagre **iff**

Topological genericness: **comeagre sets**

$G \subseteq X$ is comeagre **iff** $G \supseteq \bigcap_{i \in \omega} U_i$

Topological genericness: **comeagre sets**

$G \subseteq X$ is comeagre **iff** $G \supseteq \bigcap_{i \in \omega} U_i$ and all U_i are dense and open

Topological genericness: **comeagre sets**

$G \subseteq X$ is comeagre **iff** $G \supseteq \bigcap_{i\in\omega} U_i$ and all U_i are dense and open

$$\forall_{n\in\omega} \left(G_n \text{ is comeagre}\right) \implies \left(\bigcap_{n\in\omega} G_n\right) \text{ is comeagre}$$

Topological genericness: **comeagre sets**

$G \subseteq X$ is comeagre **iff** $G \supseteq \bigcap_{i\in\omega} U_i$ and all U_i are <u>dense</u> and open

$$\forall_{n\in\omega} \left(G_n \text{ is comeagre}\right) \implies \left(\bigcap_{n\in\omega} G_n\right) \text{ is comeagre}$$

Example

Take $U_i = \mathbb{R} - \{q_i\}$.

Topological genericness: **comeagre sets**

$G \subseteq X$ is comeagre **iff** $G \supseteq \bigcap_{i\in\omega} U_i$ and all U_i are dense and open

$$\forall_{n\in\omega} \left(G_n \text{ is comeagre}\right) \implies \left(\bigcap_{n\in\omega} G_n\right) \text{ is comeagre}$$

Example

Take $U_i = \mathbb{R} - \{q_i\}$. Then $\bigcap_{i\in\omega} U_i = \mathbb{R} - \mathbb{Q}$ is comeagre.

Topological genericness: **comeagre sets**

$G \subseteq X$ is comeagre **iff** $G \supseteq \bigcap_{i\in\omega} U_i$ and all U_i are dense and open

$$\forall_{n\in\omega} \left(G_n \text{ is comeagre}\right) \implies \left(\bigcap_{n\in\omega} G_n\right) \text{ is comeagre}$$

Example

Take $U_i = \mathbb{R} - \{q_i\}$. Then $\bigcap_{i\in\omega} U_i = \mathbb{R} - \mathbb{Q}$ is comeagre.

Theorem (Baire [1899])

In nice spaces (i.e. Polish) every comeagre set is **dense**. [thus non-empty]

Topological genericness: **comeagre sets**

$G \subseteq X$ is comeagre **iff** $G \supseteq \bigcap_{i\in\omega} U_i$ and all U_i are dense and open

$$\forall_{n\in\omega} \left(G_n \text{ is comeagre}\right) \implies \left(\bigcap_{n\in\omega} G_n\right) \text{ is comeagre}$$

Example

Take $U_i = \mathbb{R}-\{q_i\}$. Then $\bigcap_{i\in\omega} U_i = \mathbb{R}-\mathbb{Q}$ is comeagre.

Theorem (Baire [1899])

In nice spaces (i.e. Polish) every comeagre set is **dense**. [thus non-empty]

⇝ the complement of a comeagre set is **not** comeagre

Topological genericness: **comeagre sets**

$G \subseteq X$ is comeagre **iff** $G \supseteq \bigcap_{i\in\omega} U_i$ and all U_i are dense and open

$$\forall_{n\in\omega} \left(G_n \text{ is comeagre}\right) \implies \left(\bigcap_{n\in\omega} G_n\right) \text{ is comeagre}$$

Example

Take $U_i = \mathbb{R} - \{q_i\}$. Then $\bigcap_{i\in\omega} U_i = \mathbb{R} - \mathbb{Q}$ is comeagre.

Theorem (Baire [1899])

[thus non-empty]

In nice spaces (i.e. Polish) every comeagre set is **dense**.

Topological genericness: **comeagre sets**

$G \subseteq X$ is comeagre **iff** $G \supseteq \bigcap_{i\in\omega} U_i$ and all U_i are <u>dense</u> and open

$$\forall_{n\in\omega} \left(G_n \text{ is comeagre}\right) \implies \left(\bigcap_{n\in\omega} G_n\right) \text{ is comeagre}$$

Example

Take $U_i = \mathbb{R} - \{q_i\}$. Then $\bigcap_{i\in\omega} U_i = \mathbb{R} - \mathbb{Q}$ is comeagre.

Theorem (Baire [1899])

[thus non-empty]

In nice spaces (i.e. Polish) every comeagre set is **dense**.

Corollaries (non-constructive proofs of existence)

Topological genericness: **comeagre sets**

$G \subseteq X$ is comeagre **iff** $G \supseteq \bigcap_{i\in\omega} U_i$ and all U_i are dense and open

$$\forall_{n\in\omega} \left(G_n \text{ is comeagre}\right) \implies \left(\bigcap_{n\in\omega} G_n\right) \text{ is comeagre}$$

Example

Take $U_i = \mathbb{R} - \{q_i\}$. Then $\bigcap_{i\in\omega} U_i = \mathbb{R} - \mathbb{Q}$ is comeagre.

Theorem (Baire [1899])

[thus non-empty]

In nice spaces (i.e. Polish) every comeagre set is **dense**.

Corollaries (non-constructive proofs of existence)

- a continuous function **nowhere** differentiable

Topological genericness: **comeagre sets**

$G \subseteq X$ is comeagre **iff** $G \supseteq \bigcap_{i\in\omega} U_i$ and all U_i are dense and open

$$\forall_{n\in\omega} \left(G_n \text{ is comeagre}\right) \implies \left(\bigcap_{n\in\omega} G_n\right) \text{ is comeagre}$$

Example

Take $U_i = \mathbb{R}-\{q_i\}$. Then $\bigcap_{i\in\omega} U_i = \mathbb{R}-\mathbb{Q}$ is comeagre.

Theorem (Baire [1899])

In nice spaces (i.e. Polish) every comeagre set is **dense**. [thus non-empty]

Corollaries (non-constructive proofs of existence)

- a continuous function **nowhere** differentiable
- a linear partial differential equation with **no** solutions

Topological genericness: **comeagre sets**

$G \subseteq X$ is comeagre **iff** $G \supseteq \bigcap_{i \in \omega} U_i$ and all U_i are dense and open

$$\forall_{n \in \omega} \left(G_n \text{ is comeagre}\right) \implies \left(\bigcap_{n \in \omega} G_n\right) \text{ is comeagre}$$

Example

Take $U_i = \mathbb{R} - \{q_i\}$. Then $\bigcap_{i \in \omega} U_i = \mathbb{R} - \mathbb{Q}$ is comeagre.

Theorem (Baire [1899])

[thus non-empty]

In nice spaces (i.e. Polish) every comeagre set is **dense**.

Corollaries (non-constructive proofs of existence)

- a continuous function **nowhere** differentiable
- a linear partial differential equation with **no** solutions
- . . .

Topological genericness: **comeagre sets**

$G \subseteq X$ is comeagre **iff** $G \supseteq \bigcap_{i\in\omega} U_i$ and all U_i are dense and open

$$\forall_{n\in\omega} (G_n \text{ is comeagre}) \implies \left(\bigcap_{n\in\omega} G_n\right) \text{ is comeagre}$$

Example

Take $U_i = \mathbb{R} - \{q_i\}$. Then $\bigcap_{i\in\omega} U_i = \mathbb{R} - \mathbb{Q}$ is comeagre.

Theorem (Baire [1899])

In nice spaces (i.e. Polish) every comeagre set is **dense**. [thus non-empty]

Corollaries (non-constructive proofs of existence) **forcing**

- a continuous function **nowhere** differentiable
- a linear partial differential equation with **no** solutions
- ...

Which sets are comeagre?

Which sets are comeagre? (**Banach–Mazur game**)

Which sets are comeagre? (**Banach–Mazur game**) (take $W \subseteq [0,1]$)

Which sets are comeagre? (**Banach–Mazur game**) (take $W \subseteq [0,1]$)

$\mathrm{BM}(W)$ is the infinite game:

Which sets are comeagre? (**Banach–Mazur game**) (take $W \subseteq [0,1]$)

$\mathrm{BM}(W)$ is the infinite game:

(I): $0,$
(II):

Which sets are comeagre? (**Banach–Mazur game**) (take $W \subseteq [0,1]$)

$\mathrm{BM}(W)$ is the infinite game:

(I): $0, \ \underline{43226}$
(II):

Which sets are comeagre? (**Banach–Mazur game**) (take $W \subseteq [0,1]$)

$\mathrm{BM}(W)$ is the infinite game:

$$\begin{array}{rl} \text{(I)}: & 0,\ \underline{43226} \\ \text{(II)}: & \qquad\qquad \underline{19743} \end{array}$$

Which sets are comeagre? (**Banach–Mazur game**) (take $W \subseteq [0,1]$)

$\mathrm{BM}(W)$ is the infinite game:

$$
\begin{array}{rllll}
\text{(I)}: & 0, & \underline{43226} & & \underline{13} \\
\text{(II)}: & & & \underline{19743} &
\end{array}
$$

Which sets are comeagre? (**Banach–Mazur game**) (take $W \subseteq [0,1]$)

$\mathrm{BM}(W)$ is the infinite game:

(I):	0,	$\underline{43226}$		$\underline{13}$	
(II):			$\underline{19743}$		$\underline{\ }$

Which sets are comeagre? (**Banach–Mazur game**) (take $W \subseteq [0,1]$)

$\mathrm{BM}(W)$ is the infinite game:

(I):	0,	$\underline{43226}$		$\underline{13}$		$\underline{8723466}$
(II):			$\underline{19743}$		$\underline{\ }$	

Which sets are comeagre? (**Banach–Mazur game**) (take $W \subseteq [0,1]$)

$\mathrm{BM}(W)$ is the infinite game:

(I):	0, <u>43226</u>		<u>13</u>		<u>8723466</u>	
(II):		<u>19743</u>		_		<u>54326</u>

Which sets are comeagre? (**Banach–Mazur game**) (take $W \subseteq [0,1]$)

$\mathrm{BM}(W)$ is the infinite game:

(I):	0,	43226		13		8723466		…
(II):			19743		_		54326	

Which sets are comeagre? (**Banach–Mazur game**) (take $W \subseteq [0,1]$)

$\mathrm{BM}(W)$ is the infinite game:

(I):	0,	43226		13		8723466		$\cdots \rightsquigarrow \pi \in [0,1]$
(II):			19743		_		54326	

Which sets are comeagre? (**Banach–Mazur game**) (take $W \subseteq [0,1]$)

$\mathrm{BM}(W)$ is the infinite game: (II) wins π **iff** $\pi \in W$

$$\begin{array}{lllllll} \text{(I):} & 0,\ \underline{43226} & & \underline{13} & \underline{8723466} & & \\ \text{(II):} & & \underline{19743} & \underline{\ } & & \underline{54326} & \cdots \rightsquigarrow \pi \in [0,1] \end{array}$$

Which sets are comeagre? (**Banach–Mazur game**) (take $W \subseteq [0,1]$)

$\mathrm{BM}(W)$ is the infinite game: (II) wins π **iff** $\pi \in W$

$$\begin{array}{ll} \text{(I):} & 0,\ \underline{43226} \qquad\qquad \underline{13} \quad \underline{8723466} \\ \text{(II):} & \qquad\qquad \underline{19743} \qquad \underline{\ } \qquad\qquad\qquad \underline{54326} \end{array} \quad \cdots \rightsquigarrow \pi \in [0,1]$$

Theorem (Banach–Mazur [1935], Oxtoby [1957])

Player (II) has a winning strategy in $\mathrm{BM}(W)$ **iff** W is comeagre.

Which sets are comeagre? (**Banach–Mazur game**) (take $W \subseteq [0,1]$)

$\mathrm{BM}(W)$ is the infinite game: (II) wins π **iff** $\pi \in W$

$$\begin{array}{llllllll} \text{(I):} & 0, \underline{43226} & & \underline{13} & & \underline{8723466} & & \\ \text{(II):} & & \underline{19743} & & \underline{\ } & & \underline{54326} & \cdots \rightsquigarrow \pi \in [0,1] \end{array}$$

Theorem (Banach–Mazur [1935], Oxtoby [1957])

Player (II) has a winning strategy in $\mathrm{BM}(W)$ **iff** W is comeagre.

$\left[W \supseteq \bigcap_{i\in\omega} U_i \text{ -open, dense}\right]$

Which sets are comeagre? (**Banach–Mazur game**) (take $W \subseteq [0,1]$)

$\mathrm{BM}(W)$ is the infinite game: (II) wins π **iff** $\pi \in W$

$$\begin{array}{llllllll} \text{(I):} & 0,\ \underline{43226} & & \underline{13} & & \underline{8723466} & & \\ \text{(II):} & & \underline{19743} & & \underline{\ } & & \underline{54326} & \cdots \rightsquigarrow \pi \in [0,1] \end{array}$$

Theorem (Banach–Mazur [1935], Oxtoby [1957])

Player (II) has a winning strategy in $\mathrm{BM}(W)$ **iff** W is $\underbrace{\text{comeagre}}$.

$\left[W \supseteq \bigcap_{i\in\omega} U_i \text{ -open, dense}\right]$

Proof

$(\Rightarrow)$

Which sets are comeagre? (**Banach–Mazur game**) (take $W \subseteq [0,1]$)

$\mathrm{BM}(W)$ is the infinite game: (II) wins π **iff** $\pi \in W$

$$\begin{array}{llllllll} \text{(I)}: & 0,\ \underline{43226} & & \underline{13} & & \underline{8723466} & & \\ \text{(II)}: & & \underline{19743} & & \underline{\ } & & \underline{54326} & \cdots \rightsquigarrow \pi \in [0,1] \end{array}$$

Theorem (Banach–Mazur [1935], Oxtoby [1957])

Player (II) has a winning strategy in $\mathrm{BM}(W)$ **iff** W is $\underbrace{\text{comeagre}}_{\left[W \supseteq \bigcap_{i\in\omega} U_i \text{ -open, dense}\right]}$.

Proof

($\Rightarrow$) Each strategy σ provides a family U_i

Which sets are comeagre? (**Banach–Mazur game**) (take $W \subseteq [0,1]$)

$\mathrm{BM}(W)$ is the infinite game: (II) wins π **iff** $\pi \in W$

$$\begin{array}{llllllll} \text{(I)}: & 0, & \underline{43226} & & \underline{13} & & \underline{8723466} & \\ \text{(II)}: & & & \underline{19743} & & \underline{\ } & & \underline{54326} \end{array} \cdots \rightsquigarrow \pi \in [0,1]$$

Theorem (Banach–Mazur [1935], Oxtoby [1957])

Player (II) has a winning strategy in $\mathrm{BM}(W)$ **iff** W is $\underbrace{\text{comeagre}}$.

$\left[W \supseteq \bigcap_{i\in\omega} U_i \text{ -open, dense}\right]$

Proof

$(\Rightarrow)$ Each strategy σ provides a family U_i (modulo some technicalities).

Which sets are comeagre? (**Banach–Mazur game**) (take $W \subseteq [0,1]$)

$\mathrm{BM}(W)$ is the infinite game: (II) wins π **iff** $\pi \in W$

$$\begin{array}{llllllll} (\mathrm{I}): & 0, & \underline{43226} & & \underline{13} & & \underline{8723466} & \\ (\mathrm{II}): & & & \underline{19743} & & \underline{\ } & & \underline{54326} \end{array} \cdots \rightsquigarrow \pi \in [0,1]$$

Theorem (Banach–Mazur [1935], Oxtoby [1957])

Player (II) has a winning strategy in $\mathrm{BM}(W)$ **iff** W is comeagre.

$\left[W \supseteq \bigcap_{i\in\omega} U_i \text{ -open, dense}\right]$

Proof

$(\Rightarrow)$ Each strategy σ provides a family U_i (modulo some technicalities).

$(\Leftarrow)$

Which sets are comeagre? (**Banach–Mazur game**) (take $W \subseteq [0,1]$)

$\mathrm{BM}(W)$ is the infinite game: (II) wins π **iff** $\pi \in W$

$$\begin{array}{l} (\mathrm{I}){:}\ \ 0,\ \underline{43226}\qquad\qquad\ \underline{13}\qquad \underline{8723466} \\ (\mathrm{II}){:}\qquad\qquad\qquad \underline{19743}\qquad \underline{\ }\qquad\qquad\qquad \underline{54326} \end{array} \quad \cdots \rightsquigarrow \pi \in [0,1]$$

Theorem (Banach–Mazur [1935], Oxtoby [1957])

Player (II) has a winning strategy in $\mathrm{BM}(W)$ **iff** W is comeagre.

$\left[W \supseteq \bigcap_{i\in\omega} U_i \text{ -open, dense}\right]$

Proof

$(\Rightarrow)$ Each strategy σ provides a family U_i (modulo some technicalities).

$(\Leftarrow)$ Consider the strategy σ that in a round i falls into U_i.

Which sets are comeagre? (**Banach–Mazur game**) (take $W \subseteq [0,1]$)

$\mathrm{BM}(W)$ is the infinite game: (II) wins π **iff** $\pi \in W$

$$\begin{array}{lllllll} (\mathrm{I}): & 0, \ \underline{43226} & & \underline{13} & \underline{8723466} & & \\ (\mathrm{II}): & & \underline{19743} & \underline{\ } & & \underline{54326} & \cdots \rightsquigarrow \pi \in [0,1] \end{array}$$

Theorem (Banach–Mazur [1935], Oxtoby [1957])

Player (II) has a winning strategy in $\mathrm{BM}(W)$ **iff** W is $\underbrace{\text{comeagre}}_{\left[W \supseteq \bigcap_{i\in\omega} U_i \text{ -open, dense}\right]}$.

Proof

($\Rightarrow$) Each strategy σ provides a family U_i (modulo some technicalities).

($\Leftarrow$) Consider the strategy σ that in a round i falls into U_i.

Each play π consistent with σ belongs to $\bigcap_{i\in\omega} U_i \subseteq W$.

Which sets are comeagre? (**Banach–Mazur game**) (take $W \subseteq [0,1]$)

$\mathrm{BM}(W)$ is the infinite game: (II) wins π **iff** $\pi \in W$

$$\begin{array}{llllllll} \text{(I)}: & 0, \ \underline{43226} & & \underline{13} & \underline{8723466} & & \\ \text{(II)}: & & \underline{19743} & \underline{\ } & & \underline{54326} & \cdots \rightsquigarrow \pi \in [0,1] \end{array}$$

Theorem (Banach–Mazur [1935], Oxtoby [1957])

Player (II) has a winning strategy in $\mathrm{BM}(W)$ **iff** W is $\underbrace{\text{comeagre}}_{\left[W \supseteq \bigcap_{i\in\omega} U_i \text{ -open, dense}\right]}$.

Proof

($\Rightarrow$) Each strategy σ provides a family U_i (modulo some technicalities).

($\Leftarrow$) Consider the strategy σ that in a round i falls into U_i.

Each play π consistent with σ belongs to $\bigcap_{i\in\omega} U_i \subseteq W$. ■

Which sets are comeagre? (**Banach–Mazur game**) (take $W \subseteq [0,1]$)

$\mathrm{BM}(W)$ is the infinite game: (II) wins π **iff** $\pi \in W$

$$\begin{array}{lllllllll} \text{(I):} & 0, & \underline{43226} & & \underline{13} & & \underline{8723466} & & \\ \text{(II):} & & & \underline{19743} & & \underline{\ } & & \underline{54326} & \cdots \rightsquigarrow \pi \in [0,1] \end{array}$$

Theorem (Banach–Mazur [1935], Oxtoby [1957])

Player (II) has a winning strategy in $\mathrm{BM}(W)$ **iff** W is $\underbrace{\text{comeagre}}_{\left[W \supseteq \bigcap_{i\in\omega} U_i \text{ -open, dense}\right]}$.

Proof

($\Rightarrow$) Each strategy σ provides a family U_i (modulo some technicalities).

($\Leftarrow$) Consider the strategy σ that in a round i falls into U_i.

Each play π consistent with σ belongs to $\bigcap_{i\in\omega} U_i \subseteq W$. ∎

Corollary

Player (I) has a winning strategy in $\mathrm{BM}(W)$

Which sets are comeagre? (**Banach–Mazur game**) (take $W \subseteq [0,1]$)

$\mathrm{BM}(W)$ is the infinite game: (II) wins π **iff** $\pi \in W$

$$\begin{array}{llllllll} \text{(I)}: & 0, \underline{43226} & & \underline{13} & & \underline{8723466} & & \\ \text{(II)}: & & \underline{19743} & & \underline{\ } & & \underline{54326} & \cdots \rightsquigarrow \pi \in [0,1] \end{array}$$

Theorem (Banach–Mazur [1935], Oxtoby [1957])

Player (II) has a winning strategy in $\mathrm{BM}(W)$ **iff** W is $\underbrace{\text{comeagre}}$.

$[W \supseteq \bigcap_{i\in\omega} U_i$ -open, dense$]$

Proof

($\Rightarrow$) Each strategy σ provides a family U_i (modulo some technicalities).

($\Leftarrow$) Consider the strategy σ that in a round i falls into U_i.

Each play π consistent with σ belongs to $\bigcap_{i\in\omega} U_i \subseteq W$. ∎

Corollary

Player (I) has a winning strategy in $\mathrm{BM}(W)$ **iff**

$([0,1]-W)$ is comeagre on <u>some</u> interval.

Part 2

Determinacy

A game is **determined** if either (I) or (II) has a winning strategy.

A game is **determined** if either (I) or (II) has a winning strategy.

- Every game of finite duration is determined.

A game is **determined** if either (I) or (II) has a winning strategy.

- Every game of finite duration is determined.

  $\underbrace{\text{finite duration}}_{\textbf{no}\text{ infinite play } \equiv \text{ well-founded game graph}}$

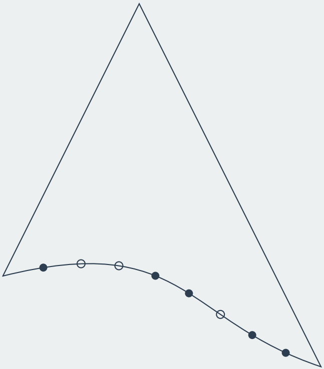
A game is **determined** if either (I) or (II) has a winning strategy.

- Every game of finite duration is determined.

  no infinite play $\equiv$ well-founded game graph

A game is **determined** if either (I) or (II) has a winning strategy.

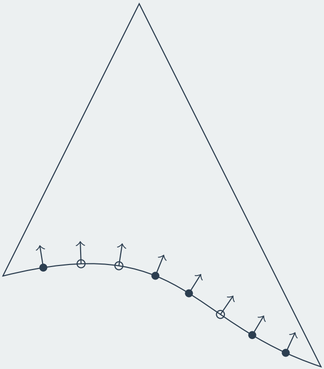
- Every game of $\underbrace{\text{finite duration}}_{\text{no infinite play} \equiv \text{well-founded game graph}}$ is determined.



A game is **determined** if either (I) or (II) has a winning strategy.

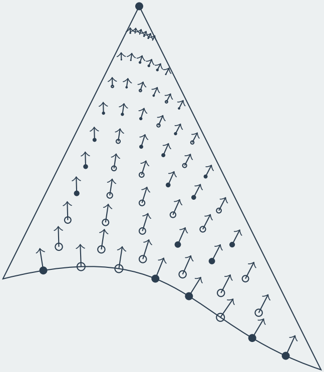
- Every game of finite duration is determined.

  no infinite play $\equiv$ well-founded game graph



A game is **determined** if either (I) or (II) has a winning strategy.

- Every game of finite duration is determined.
  no infinite play ≡ well-founded game graph



A game is **determined** if either (I) or (II) has a winning strategy.

- Every game of finite duration is determined.

A game is **determined** if either (I) or (II) has a winning strategy.

- Every game of finite duration is determined.
- There exist non-determined games of infinite duration **!**

A game is **determined** if either (I) or (II) has a winning strategy.

- Every game of finite duration is determined.
- There exist non-determined games of infinite duration !

Example (Kopczyński, Niwiński ['14] (also Khomskii ['10]; . . .))

Let $\mathrm{XOR} \subseteq \{0, 1\}^\omega$ satisfy

A game is **determined** if either (I) or (II) has a winning strategy.

- Every game of finite duration is determined.
- There exist non-determined games of infinite duration **!**

Example (Kopczyński, Niwiński ['14] (also Khomskii ['10]; ...))

Let $\mathrm{XOR} \subseteq \{0,1\}^{\omega}$ satisfy

$$0110011101011110111110101 \cdots \in \mathrm{XOR}$$

A game is **determined** if either (I) or (II) has a winning strategy.

- Every game of finite duration is determined.
- There exist non-determined games of infinite duration **!**

Example (Kopczyński, Niwiński ['14] (also Khomskii ['10]; ...))

Let $\mathrm{XOR} \subseteq \{0,1\}^{\omega}$ satisfy

$$0110011101011\mathbf{1}11011110101\cdots \in \mathrm{XOR}$$

iff

$$0110011101011\mathbf{0}11011110101\cdots \notin \mathrm{XOR}$$

A game is **determined** if either (I) or (II) has a winning strategy.

- Every game of finite duration is determined.
- There exist non-determined games of infinite duration **!**

Example (Kopczyński, Niwiński ['14] (also Khomskii ['10]; ...))

Let $\mathrm{XOR} \subseteq \{0,1\}^\omega$ satisfy

[hidden axiom of choice...]

$$01100111010\mathbf{1}11011110101\cdots \in \mathrm{XOR}$$

iff

$$01100111010\mathbf{0}11011110101\cdots \notin \mathrm{XOR}$$

A game is **determined** if either (I) or (II) has a winning strategy.

- Every game of finite duration is determined.
- There exist non-determined games of infinite duration **!**

Example (Kopczyński, Niwiński ['14] (also Khomskii ['10]; . . .))

Let $\mathrm{XOR} \subseteq \{0,1\}^\omega$ satisfy

[hidden axiom of choice. . .]

$0110011101011\mathbf{1}11011110101\cdots \in \mathrm{XOR}$

iff

$0110011101010\mathbf{0}11011110101\cdots \notin \mathrm{XOR}$

Then $\mathrm{BM}(\mathrm{XOR})$ is **non-determined !**

A game is **determined** if either (I) or (II) has a winning strategy.

- Every game of finite duration is determined.
- There exist non-determined games of infinite duration **!**

Example (Kopczyński, Niwiński ['14] (also Khomskii ['10]; . . .))

Let $\mathrm{XOR} \subseteq \{0,1\}^\omega$ satisfy

[hidden axiom of choice. . .]

$011001110101\mathbf{1}11011110101 \cdots \in \mathrm{XOR}$

iff

$011001110101\mathbf{0}11011110101 \cdots \notin \mathrm{XOR}$

Then $\mathrm{BM}(\mathrm{XOR})$ is **non-determined !**

(I):	01100		00		110010			
(II):		11011		1		00011	$\cdots$	$\rightsquigarrow \pi \in \{0,1\}^\omega$

A game is **determined** if either (I) or (II) has a winning strategy.

- Every game of finite duration is determined.
- There exist non-determined games of infinite duration **!**

Example (Kopczyński, Niwiński ['14] (also Khomskii ['10]; ...))

Let $\mathrm{XOR} \subseteq \{0,1\}^\omega$ satisfy

[hidden axiom of choice...]

$011001110101\mathbf{1}11011110101\cdots \in \mathrm{XOR}$

iff

$011001110101\mathbf{0}11011110101\cdots \notin \mathrm{XOR}$

Then $\mathrm{BM}(\mathrm{XOR})$ is **non-determined !**

(I): 01100 · 00 · 110010

(II): 11011 · 1 · 00011

$\cdots \rightsquigarrow \pi \in \{0,1\}^\omega$

(II) wins π **iff** $\pi \in \mathrm{XOR}$

A game is **determined** if either (I) or (II) has a winning strategy.

- Every game of finite duration is determined.
- There exist non-determined games of infinite duration !

Example (Kopczyński, Niwiński ['14] (also Khomskii ['10]; ...))

Let $\mathrm{XOR} \subseteq \{0,1\}^\omega$ satisfy

[hidden axiom of choice...]

$$0110011101011\mathbf{1}11011110101 \cdots \in \mathrm{XOR}$$

iff

$$0110011101010\mathbf{0}11011110101 \cdots \notin \mathrm{XOR}$$

Then $\mathrm{BM}(\mathrm{XOR})$ is **non-determined** !

(I):	01100		00		110010	
(II):		11011		1		00011

$\cdots \rightsquigarrow \pi \in \{0,1\}^\omega$

(II) wins π **iff** $\pi \in \mathrm{XOR}$

1. ((II) has a **w.s.**) $\implies$ ((I) has a **w.s.**)

A game is **determined** if either (I) or (II) has a winning strategy.

- Every game of finite duration is determined.
- There exist non-determined games of infinite duration **!**

Example (Kopczyński, Niwiński ['14] (also Khomskii ['10]; ...))

Let $\mathrm{XOR} \subseteq \{0,1\}^\omega$ satisfy

[hidden axiom of choice...]

$$01100111010\mathbf{1}11011110101\cdots \in \mathrm{XOR}$$

iff

$$01100111010\mathbf{0}11011110101\cdots \notin \mathrm{XOR}$$

Then $\mathrm{BM}(\mathrm{XOR})$ is **non-determined !**

(I):	01100		00		110010		$\cdots \rightsquigarrow \pi \in \{0,1\}^\omega$
(II):		11011		1		00011	

(II) wins π **iff** $\pi \in \mathrm{XOR}$

1. ((II) has a **w.s.**) $\implies$ ((I) has a **w.s.**)

2. ((I) has a **w.s.**) $\implies$ ((II) has a **w.s.**)

A game is **determined** if either (I) or (II) has a winning strategy.

- Every game of finite duration is determined.
- There exist non-determined games of infinite duration **!**

Example (Kopczyński, Niwiński ['14] (also Khomskii ['10]; ...))

Let $\mathrm{XOR} \subseteq \{0,1\}^\omega$ satisfy

[hidden axiom of choice...]

$$0110011101011\mathbf{1}11011110101\cdots \in \mathrm{XOR}$$

iff

$$0110011101010\mathbf{0}11011110101\cdots \notin \mathrm{XOR}$$

Then $\mathrm{BM}(\mathrm{XOR})$ is **non-determined !**

(I): $\underline{01100}$ $\underline{00}$ $\underline{110010}$

(II): $\underline{11011}$ $\underline{1}$ $\underline{00011}$ $\cdots \rightsquigarrow \pi \in \{0,1\}^\omega$

(II) wins π **iff** $\pi \in \mathrm{XOR}$

1. ((II) has a **w.s.**) $\implies$ ((I) has a **w.s.**)

2. ((I) has a **w.s.**) $\implies$ ((II) has a **w.s.**)

$\mathrm{BM}(\mathrm{XOR})$ is **non-determined!** $\qquad$ (II) wins π **iff** $\pi \in \mathrm{XOR}$

(I): $\underline{01100}$ $\qquad\qquad\qquad$ $\underline{00}$ $\qquad$ $\underline{110010}$

(II): $\qquad\qquad\qquad$ $\underline{11011}$ $\qquad\qquad$ $\underline{1}$ $\qquad\qquad\qquad$ $\underline{00011}$ $\quad \cdots \rightsquigarrow \pi \in \{0,1\}^\omega$

$$\big(\text{(I) has a } \textbf{w.s.}\big) \implies \big(\text{(II) has a } \textbf{w.s.}\big)$$

$\mathrm{BM}(\mathrm{XOR})$ is **non-determined!** $\quad$ (II) wins π **iff** $\pi \in \mathrm{XOR}$

$$\begin{array}{ll} (\mathrm{I}): & \underline{01100} \quad\quad\quad \underline{00} \quad \underline{110010} \\ (\mathrm{II}): & \quad\quad\quad \underline{11011} \quad\quad \underline{1} \quad\quad\quad\quad \underline{00011} \end{array} \cdots \rightsquigarrow \pi \in \{0,1\}^\omega$$

$\big((\mathrm{I})$ has a **w.s.**$\big) \implies \big((\mathrm{II})$ has a **w.s.**$\big)$

Proof: "strategy stealing"

$\mathrm{BM}(\mathrm{XOR})$ is **non-determined!** $\quad$ (II) wins π **iff** $\pi \in \mathrm{XOR}$

$$\begin{array}{ll} \text{(I):} & \underline{01100} \qquad\qquad \underline{00} \quad \underline{110010} \\ \text{(II):} & \qquad\quad \underline{11011} \qquad \underline{1} \qquad\qquad\quad \underline{00011} \end{array} \cdots \rightsquigarrow \pi \in \{0,1\}^\omega$$

$$\big(\text{(I) has a } \textbf{w.s.}\big) \implies \big(\text{(II) has a } \textbf{w.s.}\big)$$

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

$\mathrm{BM}(\mathrm{XOR})$ is **non-determined!** $\qquad$ (II) wins π **iff** $\pi \in \mathrm{XOR}$

$$\begin{array}{ll} (\mathrm{I}): & \underline{01100} \qquad\qquad \underline{00} \quad \underline{110010} \\ (\mathrm{II}): & \qquad\qquad \underline{11011} \qquad \underline{1} \qquad\qquad\qquad \underline{00011} \end{array} \cdots \rightsquigarrow \pi \in \{0,1\}^\omega$$

$((\mathrm{I})$ has a **w.s.**$)$ $\Longrightarrow$ $((\mathrm{II})$ has a **w.s.**$)$

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)

$\mathrm{BM}(\mathrm{XOR})$ is **non-determined!** $\quad$ (II) wins π **iff** $\pi \in \mathrm{XOR}$

(I): $\underline{01100}$ $\quad$ $\underline{00}$ $\quad$ $\underline{110010}$

(II): $\quad$ $\underline{11011}$ $\quad$ $\underline{1}$ $\quad$ $\underline{00011}$ $\quad \cdots \rightsquigarrow \pi \in \{0,1\}^\omega$

$$\big((\mathrm{I}) \text{ has a w.s.}\big) \implies \big((\mathrm{II}) \text{ has a w.s.}\big)$$

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)

σ_{I}:

(II):

$\mathrm{BM}(\mathrm{XOR})$ is **non-determined!** $\quad$ $\mathrm{(II)}$ wins π **iff** $\pi \in \mathrm{XOR}$

$$\begin{array}{lcccccccl} \text{(I):} & \underline{01100} & & \underline{00} & & \underline{110010} & & & \\ \text{(II):} & & \underline{11011} & & \underline{1} & & \underline{00011} & \cdots & \rightsquigarrow \pi \in \{0,1\}^\omega \end{array}$$

$$(\mathrm{(I)}\ \text{has a } \textbf{w.s.}) \implies (\mathrm{(II)}\ \text{has a } \textbf{w.s.})$$

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of $\mathrm{(I)}$

Construct σ_{II} — a **w.s.** of $\mathrm{(II)}$

σ_{I}:

$\mathrm{(II)}$:

$\mathrm{(I)}$:

σ_{II}:

$\mathrm{BM}(\mathrm{XOR})$ is **non-determined!** $\quad$ (II) wins π **iff** $\pi \in \mathrm{XOR}$

$$\begin{array}{ll} \text{(I):} & \underline{01100} \quad\quad\quad \underline{00} \quad\ \underline{110010} \\ \text{(II):} & \quad\quad\quad \underline{11011} \quad\quad \underline{1} \quad\quad\quad\quad\quad \underline{00011} \end{array} \cdots \rightsquigarrow \pi \in \{0,1\}^\omega$$

$$\big(\text{(I) has a **w.s.**}\big) \implies \big(\text{(II) has a **w.s.**}\big)$$

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)

σ_{I}:

(II):

(I): $\underline{r_0}$

σ_{II}:

$\mathrm{BM}(\mathrm{XOR})$ is **non-determined !** $\quad$ (II) wins π **iff** $\pi \in \mathrm{XOR}$

(I): $\underline{01100}$ $\quad$ $\underline{00}$ $\quad$ $\underline{110010}$

(II): $\quad$ $\underline{11011}$ $\quad$ $\underline{1}$ $\quad$ $\underline{00011}$ $\quad \cdots \rightsquigarrow \pi \in \{0,1\}^\omega$

$$\big((\mathrm{I}) \text{ has a } \textbf{w.s.}\big) \implies \big((\mathrm{II}) \text{ has a } \textbf{w.s.}\big)$$

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)

σ_{I}: $\underline{s_0}$

(II):

(I): $\underline{r_0}$

σ_{II}:

$\mathrm{BM}(\mathrm{XOR})$ is **non-determined!** $\quad$ (II) wins π **iff** $\pi \in \mathrm{XOR}$

$$\begin{array}{llllllll} (\mathrm{I}): & \underline{01100} & & \underline{00} & & \underline{110010} & & \\ (\mathrm{II}): & & \underline{11011} & & \underline{1} & & \underline{00011} & \cdots \rightsquigarrow \pi \in \{0,1\}^\omega \end{array}$$

$$\big((\mathrm{I}) \text{ has a } \textbf{w.s.}\big) \implies \big((\mathrm{II}) \text{ has a } \textbf{w.s.}\big)$$

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)

σ_{I}: $\underline{s_0}$

(II): $\underline{r_0 0}$

(I): $\underline{r_0}$

σ_{II}:

$\mathrm{BM}(\mathrm{XOR})$ is **non-determined !** $\qquad$ (II) wins π **iff** $\pi \in \mathrm{XOR}$

$$
\begin{array}{llllllll}
(\mathrm{I}): & \underline{01100} & & \underline{00} & & \underline{110010} & & \\
(\mathrm{II}): & & \underline{11011} & & \underline{1} & & \underline{00011} & \cdots \rightsquigarrow \pi \in \{0,1\}^\omega
\end{array}
$$

$$\big((\mathrm{I})\ \text{has a } \textbf{w.s.}\big) \implies \big((\mathrm{II})\ \text{has a } \textbf{w.s.}\big)$$

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)

$$
\begin{array}{llll}
\sigma_{\mathrm{I}}: & \underline{s_0} & & \underline{s_1} \\
(\mathrm{II}): & & \underline{r_0 0} & \\
(\mathrm{I}): & \underline{r_0} & & \\
\sigma_{\mathrm{II}}: & & &
\end{array}
$$

BM(XOR) is **non-determined**! (II) wins π **iff** $\pi \in \mathrm{XOR}$

(I): $\underline{01100}$ $\underline{00}$ $\underline{110010}$
(II): $\underline{11011}$ $\underline{1}$ $\underline{00011}$ $\cdots \rightsquigarrow \pi \in \{0,1\}^\omega$

$$\big(\text{(I) has a } \textbf{w.s.}\big) \implies \big(\text{(II) has a } \textbf{w.s.}\big)$$

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)

σ_{I}: $\underline{s_0}$ $\underline{s_1}$

(II): $\underline{r_0 0}$

(I): $\underline{r_0}$

σ_{II}: $\underline{s_0 1\ s_1}$

BM(XOR) is **non-determined!** (II) wins π **iff** $\pi \in \mathrm{XOR}$

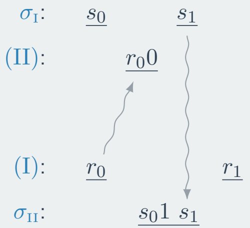
(I): 01100 | 00 | 110010
(II): 11011 | 1 | 00011
$\cdots \rightsquigarrow \pi \in \{0,1\}^\omega$

$$\big((\mathrm{I}) \text{ has a } \textbf{w.s.}\big) \implies \big((\mathrm{II}) \text{ has a } \textbf{w.s.}\big)$$

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)



BM(XOR) is **non-determined!** (II) wins π **iff** $\pi \in \mathrm{XOR}$

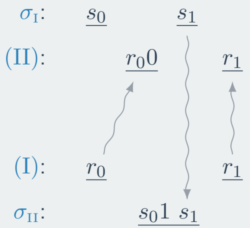
(I): $\underline{01100}$ $\underline{00}$ $\underline{110010}$
(II): $\underline{11011}$ $\underline{1}$ $\underline{00011}$ $\cdots \rightsquigarrow \pi \in \{0,1\}^\omega$

$$\big(\text{(I) has a **w.s.**}\big) \implies \big(\text{(II) has a **w.s.**}\big)$$

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)



$\mathrm{BM}(\mathrm{XOR})$ is **non-determined!** $\qquad$ (II) wins π **iff** $\pi \in \mathrm{XOR}$

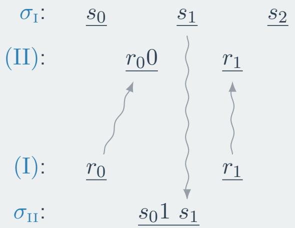
$$\begin{array}{llllllll} (\mathrm{I}): & \underline{01100} & & \underline{00} & & \underline{110010} & & \\ (\mathrm{II}): & & \underline{11011} & & \underline{1} & & \underline{00011} & \cdots \rightsquigarrow \pi \in \{0,1\}^\omega \end{array}$$

$$\big((\mathrm{I}) \text{ has a } \textbf{w.s.}\big) \implies \big((\mathrm{II}) \text{ has a } \textbf{w.s.}\big)$$

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)



$\mathrm{BM}(\mathrm{XOR})$ is **non-determined!**

(II) wins π **iff** $\pi \in \mathrm{XOR}$

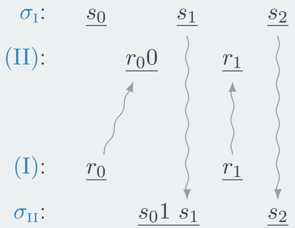
$$\begin{array}{lllllllll}
(\mathrm{I}): & \underline{01100} & & \underline{00} & & \underline{110010} & & & \\
(\mathrm{II}): & & \underline{11011} & & \underline{1} & & \underline{00011} & \cdots & \rightsquigarrow \pi \in \{0,1\}^\omega
\end{array}$$

$$\big((\mathrm{I}) \text{ has a } \textbf{w.s.}\big) \implies \big((\mathrm{II}) \text{ has a } \textbf{w.s.}\big)$$

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)



$\mathrm{BM}(\mathrm{XOR})$ is **non-determined!**

(II) wins π **iff** $\pi \in \mathrm{XOR}$

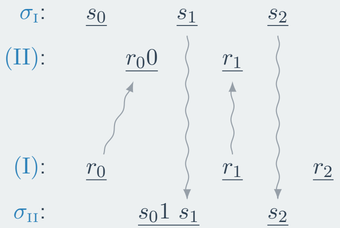
(I): $\underline{01100}$ $\underline{00}$ $\underline{110010}$
(II): $\underline{11011}$ $\underline{1}$ $\underline{00011}$ $\cdots \rightsquigarrow \pi \in \{0,1\}^\omega$

$$\big(\text{(I) has a \textbf{w.s.}}\big) \implies \big(\text{(II) has a \textbf{w.s.}}\big)$$

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)



BM(XOR) is **non-determined!** (II) wins π **iff** $\pi \in \mathrm{XOR}$

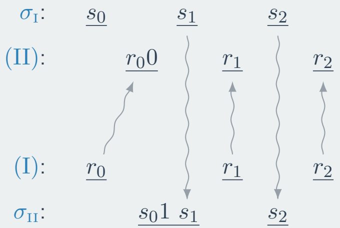
(I): 01100 00 110010
(II): 11011 1 00011 $\cdots \rightsquigarrow \pi \in \{0,1\}^\omega$

$$((\mathrm{I}) \text{ has a } \textbf{w.s.}) \implies ((\mathrm{II}) \text{ has a } \textbf{w.s.})$$

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)



BM(XOR) is **non-determined!** (II) wins π **iff** $\pi \in \mathrm{XOR}$

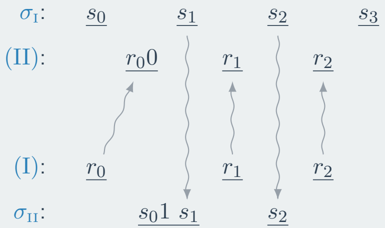
(I): $\underline{01100}$ $\underline{00}$ $\underline{110010}$
(II): $\underline{11011}$ $\underline{1}$ $\underline{00011}$ $\cdots \rightsquigarrow \pi \in \{0,1\}^\omega$

$$\big(\text{(I) has a \textbf{w.s.}}\big) \implies \big(\text{(II) has a \textbf{w.s.}}\big)$$

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)



BM(XOR) is **non-determined**! (II) wins π **iff** $\pi \in \mathrm{XOR}$

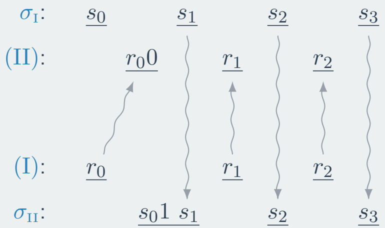
(I): $\underline{01100}$ $\underline{00}$ $\underline{110010}$
(II): $\underline{11011}$ $\underline{1}$ $\underline{00011}$ $\cdots \rightsquigarrow \pi \in \{0,1\}^\omega$

$$\big((\mathrm{I})\ \text{has a } \textbf{w.s.}\big) \implies \big((\mathrm{II})\ \text{has a } \textbf{w.s.}\big)$$

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)



BM(XOR) is **non-determined!** (II) wins π **iff** $\pi \in \text{XOR}$

(I): 01100 00 110010

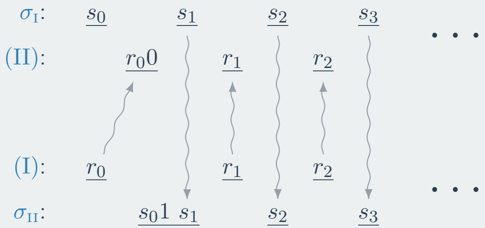
(II): 11011 1 00011 $\cdots \rightsquigarrow \pi \in \{0,1\}^\omega$

$$\big((\mathrm{I}) \text{ has a } \textbf{w.s.}\big) \implies \big((\mathrm{II}) \text{ has a } \textbf{w.s.}\big)$$

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)



BM(XOR) is **non-determined!** (II) wins π **iff** $\pi \in \mathrm{XOR}$

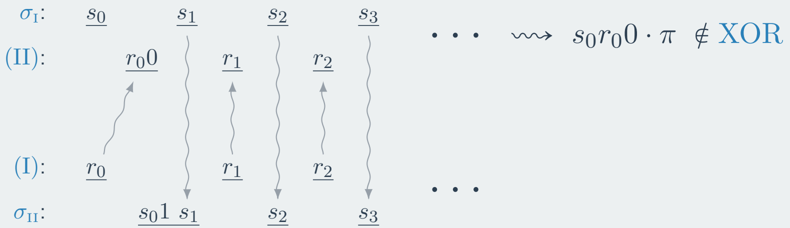
(I): 01100 | (II): 11011 | (I): 00 | (II): 1 | (I): 110010 | (II): 00011 | $\cdots \rightsquigarrow \pi \in \{0,1\}^\omega$

$$\big(\text{(I) has a \textbf{w.s.}}\big) \implies \big(\text{(II) has a \textbf{w.s.}}\big)$$

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)



$\mathrm{BM}(\mathrm{XOR})$ is **non-determined!** $\qquad$ (II) wins π **iff** $\pi \in \mathrm{XOR}$

(I): $\underline{01100}$ $\quad$ $\underline{00}$ $\quad$ $\underline{110010}$

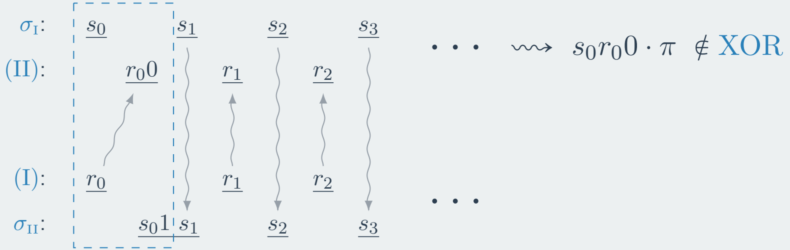
(II): $\quad$ $\underline{11011}$ $\quad$ $\underline{1}$ $\quad$ $\underline{00011}$ $\quad \cdots \rightsquigarrow \pi \in \{0,1\}^\omega$

$$\big((\mathrm{I}) \text{ has a } \textbf{w.s.}\big) \implies \big((\mathrm{II}) \text{ has a } \textbf{w.s.}\big)$$

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)



$\mathrm{BM}(\mathrm{XOR})$ is **non-determined!** $\qquad$ (II) wins π **iff** $\pi \in \mathrm{XOR}$

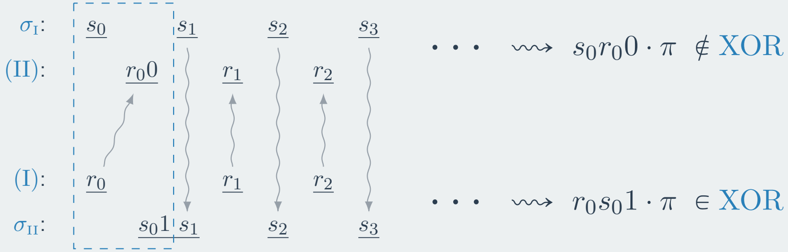
$$
\begin{array}{ll}
(\mathrm{I}): & \underline{01100} \quad\quad\quad\quad\quad \underline{00} \quad \underline{110010} \\
(\mathrm{II}): & \quad\quad\quad\quad \underline{11011} \quad\quad \underline{1} \quad\quad\quad\quad\quad\quad \underline{00011} \quad \cdots \rightsquigarrow \pi \in \{0,1\}^\omega
\end{array}
$$

$$\big((\mathrm{I}) \text{ has a } \textbf{w.s.}\big) \implies \big((\mathrm{II}) \text{ has a } \textbf{w.s.}\big)$$

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)



BM(XOR) is **non-determined!**

(II) wins π **iff** $\pi \in \mathrm{XOR}$

(I): $\underline{01100}$ $\underline{00}$ $\underline{110010}$

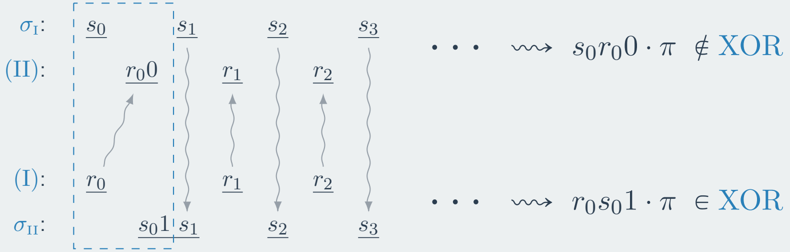
(II): $\underline{11011}$ $\underline{1}$ $\underline{00011}$ $\cdots \rightsquigarrow \pi \in \{0,1\}^\omega$

((I) has a **w.s.**) $\Longrightarrow$ ((II) has a **w.s.**)

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)



$\rightsquigarrow$ σ_{II} is a **winning strategy** of (II) ∎

BM(XOR) is **non-determined!**

(II) wins π **iff** $\pi \in \mathrm{XOR}$

(I): 01100 | 00 | 110010

(II): 11011 | 1 | 00011

$\cdots \rightsquigarrow \pi \in \{0,1\}^\omega$

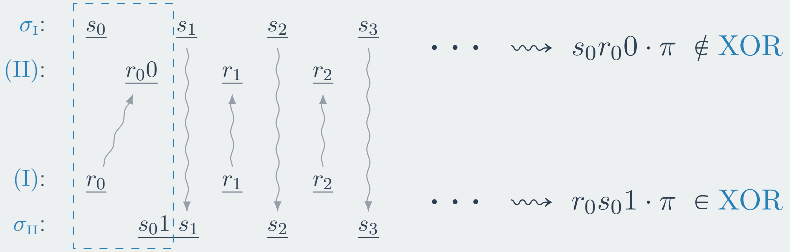
$$\big((\mathrm{I}) \text{ has a } \textbf{w.s.}\big) \implies \big((\mathrm{II}) \text{ has a } \textbf{w.s.}\big)$$

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)

$\mathrm{XOR} \cong \neg\mathrm{XOR}$



$\rightsquigarrow$ σ_{II} is a **winning strategy** of (II) ∎

Theorem (Martin ['75])

Determined are games which are:

Theorem (Martin ['75])

Determined are games which are:

- played by two players,

Theorem (Martin ['75])

Determined are games which are:

- played by two players,

- round-based,

Theorem (Martin ['75])

Determined are games which are:

- played by two players,
- round-based,
- of perfect information,

Theorem (Martin ['75])

Determined are games which are:

- played by two players,

- round-based,

- of perfect information,

- of length ω,

Theorem (Martin ['75])

Determined are games which are:

- played by two players,
- round-based,
- of perfect information,
- of length ω,

when the winning condition is **Borel**.

Theorem (Martin ['75])

Determined are games which are:

- played by two players,
- round-based,
- of perfect information,
- of length ω,

when the winning condition is **Borel**.

Corollary

All **Borel** sets have:

Theorem (Martin ['75])

Determined are games which are:

- played by two players,

- round-based,

- of perfect information,

- of length ω,

when the winning condition is **Borel**.

Corollary

All **Borel** sets have:

- perfect set property (by $*$-games),

Theorem (Martin ['75])

Determined are games which are:

- played by two players,

- round-based,

- of perfect information,

- of length ω,

when the winning condition is **Borel**.

Corollary

All **Borel** sets have:

- perfect set property (by $*$-games),
- Baire property and measurability (by BM-games),

Theorem (Martin ['75])

Determined are games which are:

- played by two players,
- round-based,
- of perfect information,
- of length ω,

when the winning condition is **Borel**.

Corollary

All **Borel** sets have:

- perfect set property (by $*$-games),
- Baire property and measurability (by BM-games),
- well-behaved Wadge hierarchy,

Theorem (Martin ['75])

Determined are games which are:

- played by two players,

- round-based,

- of perfect information,

- of length ω,

when the winning condition is **Borel**.

Corollary

All **Borel** sets have:

- perfect set property (by $*$-games),
- Baire property and measurability (by BM-games),
- well-behaved Wadge hierarchy,
- Ramsey-style dichotomies, . . .

Theorem (Martin ['75])

Determined are games which are:

- played by two players, ⇝
- round-based, ⇝
- of perfect information, ⇝
- of length ω, ⇝

Many variants:

· Blackwell games

· Nash equilibria

· ...

when the winning condition is **Borel**.

Corollary

All **Borel** sets have:

- perfect set property (by $*$-games),
- Baire property and measurability (by BM-games),
- well-behaved Wadge hierarchy,
- Ramsey-style dichotomies, ...

Part 3

Effectiveness

Regular languages

Regular languages

A set $L \subseteq A^\omega$ is regular if (equivalently) it can be:

Regular languages

A set $L \subseteq A^\omega$ is regular if (equivalently) it can be:

— defined in Monadic Second-order logic: $\mathrm{FO}[\leqslant, A] + \exists X$,

Regular languages

A set $L \subseteq A^\omega$ is regular if (equivalently) it can be:

— defined in Monadic Second-order logic: $\textsc{fo}[\leqslant, A] + \exists X$,

— recognised by a non-deterministic **Büchi** automaton,

Regular languages

A set $L \subseteq A^\omega$ is regular if (equivalently) it can be:

— defined in Monadic Second-order logic: $\mathrm{FO}[\leqslant, A] + \exists X$,

— recognised by a non-deterministic **Büchi** automaton,

— recognised by a deterministic **parity** automaton,

Regular languages

A set $L \subseteq A^\omega$ is regular if (equivalently) it can be:

— defined in Monadic Second-order logic: $\mathrm{FO}[\leqslant, A] + \exists X$,

— recognised by a non-deterministic **Büchi** automaton,

— recognised by a deterministic **parity** automaton,

— ...

Regular languages

A set $L \subseteq A^\omega$ is regular if (equivalently) it can be:

— defined in Monadic Second-order logic: $\mathrm{FO}[\leqslant, A] + \exists X$,

— recognised by a non-deterministic **Büchi** automaton,

— recognised by a deterministic **parity** automaton,

— ...

Regular sets is the smallest family $\mathbf{REG}$ that:

Regular languages

A set $L \subseteq A^\omega$ is regular if (equivalently) it can be:

— defined in Monadic Second-order logic: $\mathrm{FO}[\leqslant, A] + \exists X$,

— recognised by a non-deterministic **Büchi** automaton,

— recognised by a deterministic **parity** automaton,

— ...

Regular sets is the smallest family $\mathbf{REG}$ that:

— contains some basic languages,

Regular languages

A set $L \subseteq A^\omega$ is regular if (equivalently) it can be:

— defined in Monadic Second-order logic: $\mathrm{FO}[\leqslant, A] + \exists X$,

— recognised by a non-deterministic **Büchi** automaton,

— recognised by a deterministic **parity** automaton,

— ...

Regular sets is the smallest family $\mathbf{REG}$ that:

— contains some basic languages,

— is **closed** under Boolean operations,

Regular languages

A set $L \subseteq A^\omega$ is regular if (equivalently) it can be:

— defined in Monadic Second-order logic: $\textsc{fo}[\leqslant, A] + \exists X$,

— recognised by a non-deterministic **Büchi** automaton,

— recognised by a deterministic **parity** automaton,

— ...

Regular sets is the smallest family $\mathbf{REG}$ that:

— contains some basic languages,

— is **closed** under Boolean operations,

— and under projection $(A\times B)^\omega \twoheadrightarrow A^\omega$.

Regular languages

A set $L \subseteq A^\omega$ is regular if (equivalently) it can be:

— defined in Monadic Second-order logic: $\mathrm{FO}[\leqslant, A] + \exists X$,

— recognised by a non-deterministic **Büchi** automaton,

— recognised by a deterministic **parity** automaton,

— ...

Regular sets is the smallest family $\mathbf{REG}$ that:

— contains some basic languages,

— is **closed** under Boolean operations,

— and under projection $\big(A \times B\big)^\omega \twoheadrightarrow A^\omega$.

Facts:

Regular languages

A set $L \subseteq A^\omega$ is regular if (equivalently) it can be:

— defined in Monadic Second-order logic: $\mathrm{FO}[\leqslant, A] + \exists X$,

— recognised by a non-deterministic **Büchi** automaton,

— recognised by a deterministic **parity** automaton,

— ...

Regular sets is the smallest family $\mathbf{REG}$ that:

— contains some basic languages,

— is **closed** under Boolean operations,

— and under projection $(A \times B)^\omega \twoheadrightarrow A^\omega$.

Facts: $\mathbf{REG} \subseteq \mathbf{Borel}$,

Regular languages

A set $L \subseteq A^\omega$ is regular if (equivalently) it can be:

— defined in Monadic Second-order logic: $\text{FO}[\leqslant, A] + \exists X$,

— recognised by a non-deterministic **Büchi** automaton,

— recognised by a deterministic **parity** automaton,

— ...

Regular sets is the smallest family $\mathbf{REG}$ that:

— contains some basic languages,

— is **closed** under Boolean operations,

— and under projection $\left(A \times B\right)^\omega \twoheadrightarrow A^\omega$.

Facts: $\mathbf{REG} \subseteq \mathbf{Borel}$, $\text{proj}(\mathbf{REG}) \subseteq \mathbf{REG}$,

Regular languages

A set $L \subseteq A^\omega$ is regular if (equivalently) it can be:

— defined in Monadic Second-order logic: $\textsc{fo}[\leqslant, A] + \exists X$,

— recognised by a non-deterministic **Büchi** automaton,

— recognised by a deterministic **parity** automaton,

— ...

Regular sets is the smallest family $\mathbf{REG}$ that:

— contains some basic languages,

— is **closed** under Boolean operations,

— and under projection $\big(A \times B\big)^\omega \twoheadrightarrow A^\omega$.

Facts: $\mathbf{REG} \subseteq \mathbf{Borel}$, $\text{proj}(\mathbf{REG}) \subseteq \mathbf{REG}$, $\text{proj}(\mathbf{Borel}) \nsubseteq \mathbf{Borel}$.

Regular languages

A set $L \subseteq A^\omega$ is regular if (equivalently) it can be:

— defined in Monadic Second-order logic: $\mathrm{FO}[\leqslant, A] + \exists X$,

— recognised by a non-deterministic **Büchi** automaton,

— recognised by a deterministic **parity** automaton,

— ...

Regular sets is the smallest family $\mathbf{REG}$ that:

— contains some basic languages,

— is **closed** under Boolean operations,

— and under projection $\big(A \times B\big)^\omega \twoheadrightarrow A^\omega$.

Facts: $\mathbf{REG} \subseteq \mathbf{Borel}$, $\mathrm{proj}(\mathbf{REG}) \subseteq \mathbf{REG}$, $\mathrm{proj}(\mathbf{Borel}) \nsubseteq \mathbf{Borel}$.

Theorem (Büchi ['62])

Given (a represenation of) $L \in \mathbf{REG}$ it is **decidable** if $L \neq \emptyset$.

Theorem (Büchi, Landweber ['69])

Theorem (Büchi, Landweber ['69])

Fix $W \subseteq A^\omega$ **regular** (i.e. $W \in \mathbf{REG}$).

Theorem (Büchi, Landweber ['69])

Fix $W \subseteq A^\omega$ **regular** (i.e. $W \in \mathbf{REG}$).

Consider a game $\mathcal{G}(W)$:

Theorem (Büchi, Landweber ['69])

Fix $W \subseteq A^\omega$ **regular** (i.e. $W \in \mathbf{REG}$).

Consider a game $\mathcal{G}(W)$:

$$\begin{array}{lccccccccccl} \text{(I):} & \underline{a_0} & & \underline{a_2} & & \underline{a_4} & & \underline{a_6} & & \underline{a_8} & \dots \\ \text{(II):} & & \underline{a_1} & & \underline{a_3} & & \underline{a_5} & & \underline{a_7} & & \end{array}$$

Theorem (Büchi, Landweber ['69])

Fix $W \subseteq A^\omega$ **regular** (i.e. $W \in \mathbf{REG}$).

Consider a game $\mathcal{G}(W)$:

$$\begin{array}{lccccccccccl} \text{(I):} & \underline{a_0} & & \underline{a_2} & & \underline{a_4} & & \underline{a_6} & & \underline{a_8} & \cdots & \rightsquigarrow \pi = \big(a_0 a_1 \cdots\big) \in A^\omega \\ \text{(II):} & & \underline{a_1} & & \underline{a_3} & & \underline{a_5} & & \underline{a_7} & & & \end{array}$$

Theorem (Büchi, Landweber ['69])

Fix $W \subseteq A^\omega$ **regular** (i.e. $W \in \mathbf{REG}$).

Consider a game $\mathcal{G}(W)$:

$$\begin{array}{lccccccccccl} \text{(I):} & \underline{a_0} & & \underline{a_2} & & \underline{a_4} & & \underline{a_6} & & \underline{a_8} & & \\ \text{(II):} & & \underline{a_1} & & \underline{a_3} & & \underline{a_5} & & \underline{a_7} & & \cdots & \rightsquigarrow\ \pi = (a_0 a_1 \cdots) \in A^\omega \end{array}$$

(II) wins π **iff** $\pi \in W$

Theorem (Büchi, Landweber ['69])

Fix $W \subseteq A^\omega$ **regular** (i.e. $W \in \mathbf{REG}$).

Consider a game $\mathcal{G}(W)$:

$$\begin{array}{ll} \text{(I)}: & \underline{a_0} \quad\quad \underline{a_2} \quad\quad \underline{a_4} \quad\quad \underline{a_6} \quad\quad \underline{a_8} \\ \text{(II)}: & \quad\quad \underline{a_1} \quad\quad \underline{a_3} \quad\quad \underline{a_5} \quad\quad \underline{a_7} \end{array} \cdots \rightsquigarrow \pi = (a_0 a_1 \cdots) \in A^\omega$$

(II) wins π **iff** $\pi \in W$

Then:

Theorem (Büchi, Landweber ['69])

Fix $W \subseteq A^\omega$ **regular** (i.e. $W \in \mathbf{REG}$).

Consider a game $\mathcal{G}(W)$:

$$\begin{array}{lccccccccccl} \text{(I):} & \underline{a_0} & & \underline{a_2} & & \underline{a_4} & & \underline{a_6} & & \underline{a_8} & & \\ \text{(II):} & & \underline{a_1} & & \underline{a_3} & & \underline{a_5} & & \underline{a_7} & & \cdots & \rightsquigarrow \pi = (a_0 a_1 \cdots) \in A^\omega \end{array}$$

$$\text{(II) wins } \pi \quad \textbf{iff} \quad \pi \in W$$

Then:

1. $\mathcal{G}(W)$ is determined. (because W is **Borel**)

Theorem (Büchi, Landweber ['69])

Fix $W \subseteq A^\omega$ **regular** (i.e. $W \in \mathbf{REG}$).

Consider a game $\mathcal{G}(W)$:

$$\begin{array}{lccccccccccl} \text{(I):} & \underline{a_0} & & \underline{a_2} & & \underline{a_4} & & \underline{a_6} & & \underline{a_8} & \cdots & \rightsquigarrow \pi = (a_0 a_1 \cdots) \in A^\omega \\ \text{(II):} & & \underline{a_1} & & \underline{a_3} & & \underline{a_5} & & \underline{a_7} & & & \end{array}$$

(II) wins π **iff** $\pi \in W$

Then:

1. $\mathcal{G}(W)$ is determined. (because W is **Borel**)
2. The winner of $\mathcal{G}(W)$ can be effectively computed.

Theorem (Büchi, Landweber ['69])

Fix $W \subseteq A^\omega$ **regular** (i.e. $W \in \mathbf{REG}$).

Consider a game $\mathcal{G}(W)$:

$$\begin{array}{lccccccccccc} \text{(I):} & \underline{a_0} & & \underline{a_2} & & \underline{a_4} & & \underline{a_6} & & \underline{a_8} & & \\ \text{(II):} & & \underline{a_1} & & \underline{a_3} & & \underline{a_5} & & \underline{a_7} & & \cdots & \rightsquigarrow \pi = (a_0 a_1 \cdots) \in A^\omega \end{array}$$

$$\text{(II) wins } \pi \quad \textbf{iff} \quad \pi \in W$$

Then:

1. $\mathcal{G}(W)$ is determined. (because W is **Borel**)
2. The winner of $\mathcal{G}(W)$ can be effectively computed.
3. The winner can use a finite memory winning strategy:

Theorem (Büchi, Landweber ['69])

Fix $W \subseteq A^\omega$ **regular** (i.e. $W \in \mathbf{REG}$).

Consider a game $\mathcal{G}(W)$:

$$\begin{array}{lccccccccccl} \text{(I)}: & \underline{a_0} & & \underline{a_2} & & \underline{a_4} & & \underline{a_6} & & \underline{a_8} & & \\ \text{(II)}: & & \underline{a_1} & & \underline{a_3} & & \underline{a_5} & & \underline{a_7} & & \cdots & \rightsquigarrow \pi = (a_0 a_1 \cdots) \in A^\omega \end{array}$$

$$\text{(II) wins } \pi \quad \textbf{iff} \quad \pi \in W$$

Then:

1. $\mathcal{G}(W)$ is determined. (because W is **Borel**)
2. The winner of $\mathcal{G}(W)$ can be effectively computed.
3. The winner can use a finite memory winning strategy:

   There is a finite set M of memory values,

Theorem (Büchi, Landweber ['69])

Fix $W \subseteq A^\omega$ **regular** (i.e. $W \in \mathbf{REG}$).

Consider a game $\mathcal{G}(W)$:

$$\begin{array}{ll} \text{(I)}: & \underline{a_0} \quad\quad \underline{a_2} \quad\quad \underline{a_4} \quad\quad \underline{a_6} \quad\quad \underline{a_8} \\ \text{(II)}: & \quad\quad \underline{a_1} \quad\quad \underline{a_3} \quad\quad \underline{a_5} \quad\quad \underline{a_7} \end{array} \quad \cdots \rightsquigarrow \pi = (a_0 a_1 \cdots) \in A^\omega$$

(II) wins π **iff** $\pi \in W$

Then:

1. $\mathcal{G}(W)$ is determined. (because W is **Borel**)
2. The winner of $\mathcal{G}(W)$ can be effectively computed.
3. The winner can use a finite memory winning strategy:

   There is a finite set M of memory values,

   initial memory $m_0 \in M$,

Theorem (Büchi, Landweber ['69])

Fix $W \subseteq A^\omega$ **regular** (i.e. $W \in \mathbf{REG}$).

Consider a game $\mathcal{G}(W)$:

$$\begin{array}{lccccccccccc} \text{(I):} & \underline{a_0} & & \underline{a_2} & & \underline{a_4} & & \underline{a_6} & & \underline{a_8} & & \\ \text{(II):} & & \underline{a_1} & & \underline{a_3} & & \underline{a_5} & & \underline{a_7} & & \cdots & \rightsquigarrow \pi = (a_0 a_1 \cdots) \in A^\omega \end{array}$$

$$\text{(II) wins } \pi \quad \textbf{iff} \quad \pi \in W$$

Then:

1. $\mathcal{G}(W)$ is determined. (because W is **Borel**)
2. The winner of $\mathcal{G}(W)$ can be effectively computed.
3. The winner can use a finite memory winning strategy:

   There is a finite set M of memory values,

   initial memory $m_0 \in M$, and update function $\delta\colon M \times A \to M$,

Theorem (Büchi, Landweber ['69])

Fix $W \subseteq A^\omega$ **regular** (i.e. $W \in \mathbf{REG}$).

Consider a game $\mathcal{G}(W)$:

$$\begin{array}{lccccccccccl} \text{(I)}: & \underline{a_0} & & \underline{a_2} & & \underline{a_4} & & \underline{a_6} & & \underline{a_8} & \cdots \rightsquigarrow \pi = (a_0 a_1 \cdots) \in A^\omega \\ \text{(II)}: & & \underline{a_1} & & \underline{a_3} & & \underline{a_5} & & \underline{a_7} & & \end{array}$$

(II) wins π **iff** $\pi \in W$

Then:

1. $\mathcal{G}(W)$ is determined. (because W is **Borel**)
2. The winner of $\mathcal{G}(W)$ can be effectively computed.
3. The winner can use a finite memory winning strategy:

   There is a finite set M of memory values,

   initial memory $m_0 \in M$, and update function $\delta\colon M \times A \to M$,

   such that for $m_{i+1} \stackrel{\text{def}}{=} \delta(m_i, a_i)$,

Theorem (Büchi, Landweber ['69])

Fix $W \subseteq A^\omega$ **regular** (i.e. $W \in \mathbf{REG}$).

Consider a game $\mathcal{G}(W)$:

$$\begin{array}{lccccccccccl} \text{(I):} & \underline{a_0} & & \underline{a_2} & & \underline{a_4} & & \underline{a_6} & & \underline{a_8} & \cdots & \rightsquigarrow \pi = (a_0 a_1 \cdots) \in A^\omega \\ \text{(II):} & & \underline{a_1} & & \underline{a_3} & & \underline{a_5} & & \underline{a_7} & & & \end{array}$$

$$\text{(II) wins } \pi \quad \textbf{iff} \quad \pi \in W$$

Then:

1. $\mathcal{G}(W)$ is determined. (because W is **Borel**)
2. The winner of $\mathcal{G}(W)$ can be effectively computed.
3. The winner can use a finite memory winning strategy:

   There is a finite set M of memory values,

   initial memory $m_0 \in M$, and update function $\delta\colon M \times A \to M$,

   such that for $m_{i+1} \stackrel{\text{def}}{=} \delta(m_i, a_i)$,

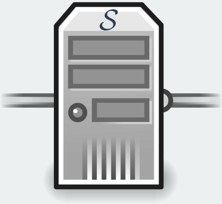
   the choice of a_i depends **only** on m_i.

Part 4

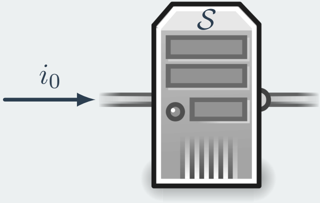
Applications

Synthesis

Synthesis

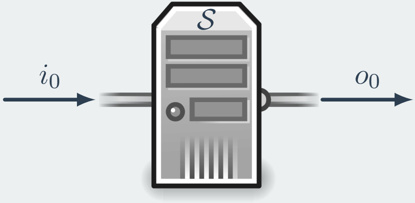


Synthesis



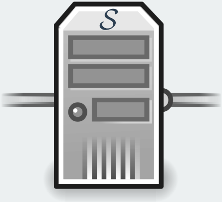
Trace $\tau = (i_0$

Synthesis



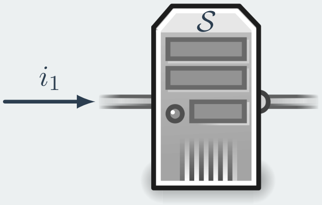
Trace $\tau = (i_0\ o_0$

Synthesis



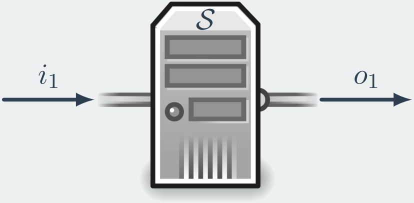
Trace $\tau = (i_0\ o_0$

Synthesis



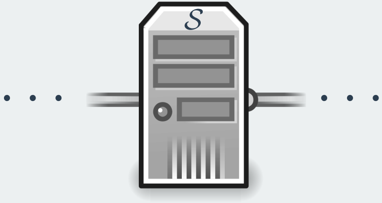
Trace $\tau = (i_0\ o_0\ i_1$

Synthesis



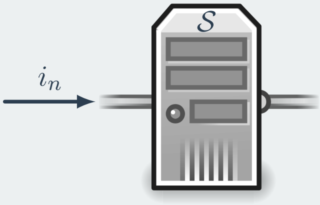
Trace $\tau = (i_0\ o_0\ i_1\ o_1$

Synthesis



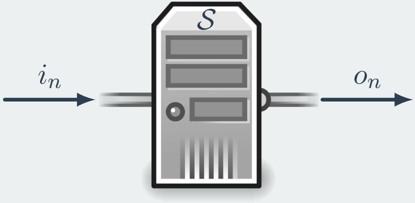
Trace $\tau = (i_0\ o_0\ i_1\ o_1\ \cdots$

Synthesis



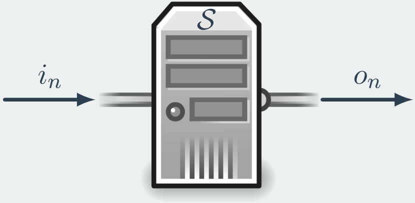
Trace $\tau = (i_0\ o_0\ i_1\ o_1\ \cdots\ i_n$

Synthesis



Trace $\tau = (i_0\ o_0\ i_1\ o_1\ \cdots\ i_n\ o_n$

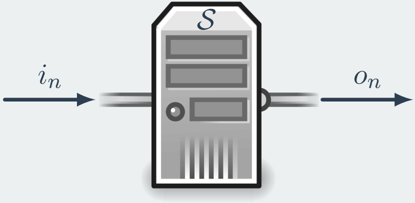
Synthesis



Trace $\tau = (i_0\ o_0\ i_1\ o_1\ \cdots\ i_n\ o_n\ \cdots) \in \big(I \sqcup O\big)^\omega$

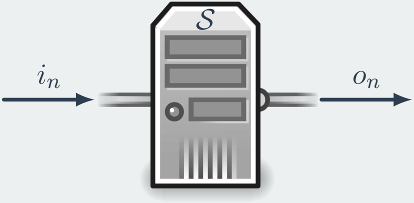
Synthesis

Specification
φ over $I \sqcup O$



Trace $\tau = (i_0\ o_0\ i_1\ o_1\ \cdots\ i_n\ o_n\ \cdots) \in (I \sqcup O)^\omega$

Specification φ over $I \sqcup O$ — **Synthesis** $\longrightarrow$ Implementation $\mathcal{S}\colon I \rightsquigarrow O$

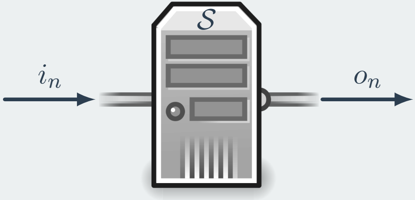


Trace $\tau = (i_0\ o_0\ i_1\ o_1\ \cdots\ i_n\ o_n\ \cdots) \in \big(I \sqcup O\big)^{\omega}$

Specification
φ over $I \sqcup O$

Synthesis

Implementation
$\mathcal{S}\colon I \rightsquigarrow O$
[whenever possible]

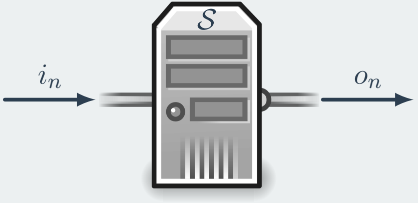


Trace $\tau = (i_0\ o_0\ i_1\ o_1\ \cdots\ i_n\ o_n\ \cdots) \in (I \sqcup O)^\omega$

Specification φ over $I \sqcup O$ **Synthesis** Implementation $\mathcal{S}\colon I \rightsquigarrow O$

[whenever possible]

$\varphi \equiv$ "$o_0 = i_1$"



Trace $\tau = (i_0\ o_0\ i_1\ o_1\ \cdots\ i_n\ o_n\ \cdots) \in (I \sqcup O)^\omega$

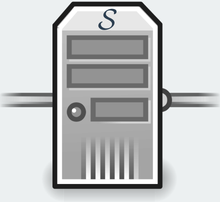
Specification
φ over $I \sqcup O$

Synthesis

Implementation
$\mathcal{S}\colon I \rightsquigarrow O$

[whenever possible]

$\varphi \equiv$ "$o_0 = i_1$"



Env.:

Impl.:

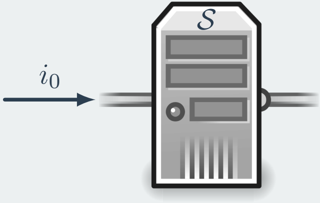
Specification
φ over $I \sqcup O$

Synthesis

Implementation
$\mathcal{S}\colon I \rightsquigarrow O$

[whenever possible]

$\varphi \equiv$ "$o_0 = i_1$"



Env.: $\underline{i_0}$

Impl.:

Specification
φ over $I \sqcup O$

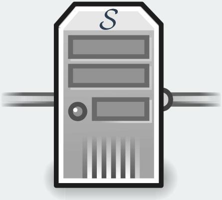
Synthesis

Implementation
$\mathcal{S}\colon I \rightsquigarrow O$

[whenever possible]

$\varphi \equiv$ "$o_0 = i_1$"

i_0



o_0

Env.: i_0

Impl.: o_0

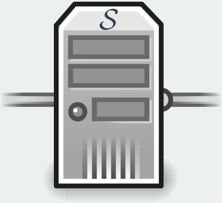
Specification
φ over $I \sqcup O$

Synthesis

Implementation
$\mathcal{S}\colon I \rightsquigarrow O$

[whenever possible]

$\varphi \equiv$ "$o_0 = i_1$"



Env.: $\underline{i_0}$

Impl.: $\underline{o_0}$

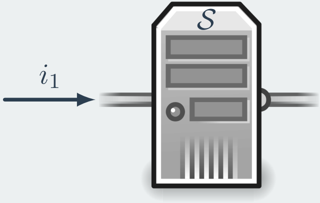
Specification
φ over $I \sqcup O$

Synthesis

Implementation
$\mathcal{S}\colon I \rightsquigarrow O$

[whenever possible]

$\varphi \equiv$ "$o_0 = i_1$"



Env.: $\underline{i_0}$ $\underline{i_1}$

Impl.: $\underline{o_0}$

Specification
φ over $I \sqcup O$

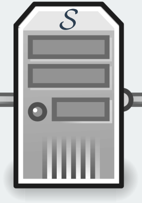
Synthesis

Implementation
$\mathcal{S}\colon I \rightsquigarrow O$

[whenever possible]

$\varphi \equiv$ "$o_0 = i_1$"

i_1



o_1

Env.: $\underline{i_0}$ $\underline{i_1}$

Impl.: $\underline{o_0}$ $\underline{o_1}$

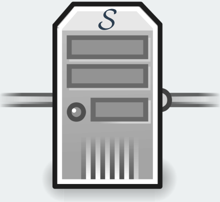
Specification
φ over $I \sqcup O$

Synthesis

Implementation
$\mathcal{S}\colon I \rightsquigarrow O$

[whenever possible]

$\varphi \equiv$ "$o_0 = i_1$"



Env.: $\underline{i_0}$ $\underline{i_1}$

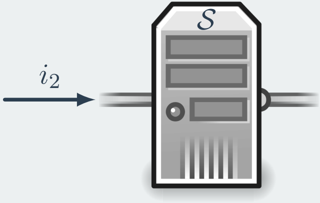
Impl.: $\underline{o_0}$ $\underline{o_1}$

Specification
φ over $I \sqcup O$

Synthesis ⟶

Implementation
$\mathcal{S}\colon I \rightsquigarrow O$

$\underbrace{\text{[whenever possible]}}_{\varphi \equiv \text{“}o_0 = i_1\text{”}}$



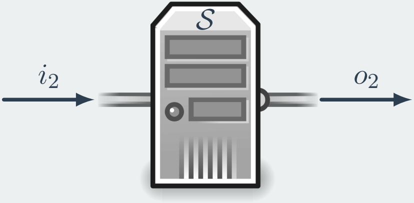
Env.: $\underline{i_0}$ $\underline{i_1}$ $\underline{i_2}$

Impl.: $\underline{o_0}$ $\underline{o_1}$

Specification
φ over $I \sqcup O$

Synthesis ⟶

Implementation
$\mathcal{S}\colon I \rightsquigarrow O$
[whenever possible]
$\varphi \equiv$ "$o_0 = i_1$"



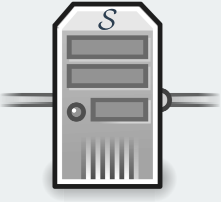
Env.: $\underline{i_0}$ $\underline{i_1}$ $\underline{i_2}$

Impl.: $\underline{o_0}$ $\underline{o_1}$ $\underline{o_2}$

Specification φ over $I \sqcup O$ —— **Synthesis** ——→ **Implementation** $\mathcal{S}\colon I \rightsquigarrow O$

[whenever possible]

$\varphi \equiv$ "$o_0 = i_1$"



Env.: $\underline{i_0}$ $\underline{i_1}$ $\underline{i_2}$

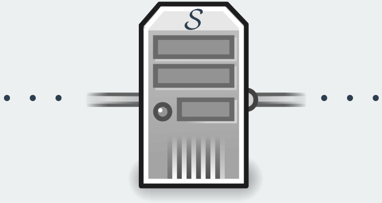
Impl.: $\underline{o_0}$ $\underline{o_1}$ $\underline{o_2}$

Specification
φ over $I \sqcup O$

Synthesis

Implementation
$\mathcal{S}\colon I \rightsquigarrow O$

[whenever possible]
$\varphi \equiv$ "$o_0 = i_1$"



Env.:	$\underline{i_0}$		$\underline{i_1}$		$\underline{i_2}$		$\cdots$
Impl.:		$\underline{o_0}$		$\underline{o_1}$		$\underline{o_2}$	$\cdots$

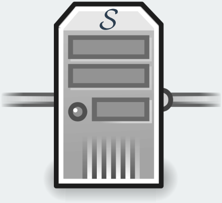
Specification
φ over $I \sqcup O$

Synthesis

Implementation
$\mathcal{S} \colon I \rightsquigarrow O$

[whenever possible]

$\varphi \equiv$ "$o_0 = i_1$"



Env.: $\underline{i_0}$ $\underline{i_1}$ $\underline{i_2}$ $\cdots$

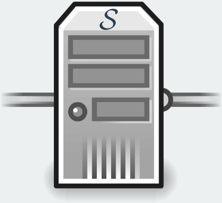
Impl.: $\underline{o_0}$ $\underline{o_1}$ $\underline{o_2}$ $\cdots$

Specification
φ over $I \sqcup O$

Synthesis ⟶

Implementation
$\mathcal{S}\colon I \rightsquigarrow O$
[whenever possible]
$\varphi \equiv$ "$o_0 = i_1$"

i_n →



Env.:	$\underline{i_0}$		$\underline{i_1}$		$\underline{i_2}$		$\cdots$	$\underline{i_n}$
Impl.:		$\underline{o_0}$		$\underline{o_1}$		$\underline{o_2}$	$\cdots$	

Specification
φ over $I \sqcup O$

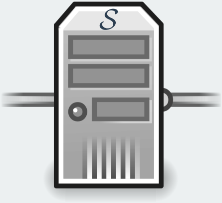
Synthesis ⟶

Implementation
$\mathcal{S}\colon I \rightsquigarrow O$

[whenever possible]

$\varphi \equiv$ "$o_0 = i_1$"

i_n ⟶



⟶ o_n

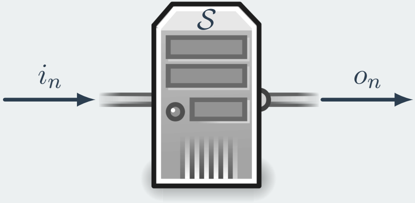
Env.: $\underline{i_0}$ $\underline{i_1}$ $\underline{i_2}$ $\cdots$ $\underline{i_n}$

Impl.: $\underline{o_0}$ $\underline{o_1}$ $\underline{o_2}$ $\cdots$ $\underline{o_n}$

Specification φ over $I \sqcup O$ —— **Synthesis** ——→ **Implementation** $\mathcal{S}\colon I \rightsquigarrow O$

[whenever possible]

$\varphi \equiv$ "$o_0 = i_1$"



Env.: $\underline{i_0}$ $\underline{i_1}$ $\underline{i_2}$ $\cdots$ $\underline{i_n}$

Impl.: $\underline{o_0}$ $\underline{o_1}$ $\underline{o_2}$ $\cdots$ $\underline{o_n}$

$\cdots \rightsquigarrow \tau \overset{?}{\models} \varphi$

Specification
φ over $I \sqcup O$

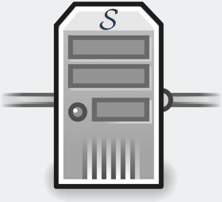
Synthesis $\longmapsto$

Implementation
$\mathcal{S}\colon I \rightsquigarrow O$

[whenever possible]

$\varphi \equiv$ "$o_0 = i_1$"

$i_n \rightarrow$



$\rightarrow o_n$

Env.: $\underline{i_0}$ $\underline{i_1}$ $\underline{i_2}$ $\cdots$ $\underline{i_n}$

Impl.: $\underline{o_0}$ $\underline{o_1}$ $\underline{o_2}$ $\cdots$ $\underline{o_n}$

$\cdots \rightsquigarrow \tau \overset{?}{\models} \varphi$

Solve $\mathcal{G}\big(\mathrm{L}(\varphi)\big)$

Specification
φ over $I \sqcup O$

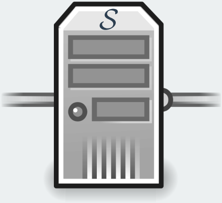
Synthesis

Implementation
$\mathcal{S}\colon I \rightsquigarrow O$

[whenever possible]

$\varphi \equiv$ "$o_0 = i_1$"

i_n



o_n

Env.: $\underline{i_0}$ $\underline{i_1}$ $\underline{i_2}$ $\cdots$ $\underline{i_n}$

Impl.: $\underline{o_0}$ $\underline{o_1}$ $\underline{o_2}$ $\cdots$ $\underline{o_n}$

$\cdots \rightsquigarrow \tau \overset{?}{\models} \varphi$

Solve $\mathcal{G}\big(\mathrm{L}(\varphi)\big)$

Env wins

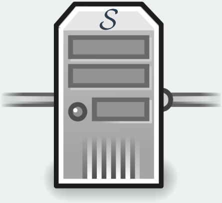
Specification φ over $I \sqcup O$

Synthesis $\longmapsto$

Implementation $\mathcal{S}\colon I \rightsquigarrow O$

[whenever possible]

$\varphi \equiv$ "$o_0 = i_1$"

$i_n \rightarrow$  $\rightarrow o_n$

Env.: $\underline{i_0}$ $\underline{i_1}$ $\underline{i_2}$ $\cdots$ $\underline{i_n}$

Impl.: $\underline{o_0}$ $\underline{o_1}$ $\underline{o_2}$ $\cdots$ $\underline{o_n}$

$\cdots \rightsquigarrow \tau \overset{?}{\models} \varphi$

Solve $\mathcal{G}\big(\mathrm{L}(\varphi)\big)$

Env wins

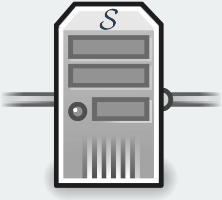
$\Downarrow$

φ is unrealisable

Specification φ over $I \sqcup O$ —— **Synthesis** ——> Implementation $\mathcal{S}\colon I \rightsquigarrow O$

[whenever possible] ($\varphi \equiv$ "$o_0 = i_1$")

i_n →



→ o_n

Env.: $\underline{i_0}$ $\underline{i_1}$ $\underline{i_2}$ $\cdots$ $\underline{i_n}$

Impl.: $\underline{o_0}$ $\underline{o_1}$ $\underline{o_2}$ $\cdots$ $\underline{o_n}$

$\cdots \rightsquigarrow \tau \overset{?}{\models} \varphi$

Solve $\mathcal{G}\big(\mathrm{L}(\varphi)\big)$

Env wins

$\Downarrow$

φ is unrealisable

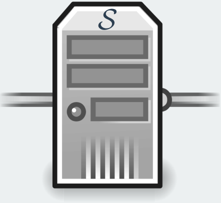
Impl wins

Specification
φ over $I \sqcup O$

Synthesis

Implementation
$\mathcal{S}\colon I \rightsquigarrow O$
[whenever possible]
$\varphi \equiv$ "$o_0 = i_1$"

i_n



o_n

Env.: $\underline{i_0}$ $\underline{i_1}$ $\underline{i_2}$ $\cdots$ $\underline{i_n}$

Impl.: $\underline{o_0}$ $\underline{o_1}$ $\underline{o_2}$ $\cdots$ $\underline{o_n}$

$\cdots \rightsquigarrow \tau \overset{?}{\models} \varphi$

Solve $\mathcal{G}\big(\mathrm{L}(\varphi)\big)$

Env wins
$\Downarrow$
φ is unrealisable

Impl wins
$\Downarrow$
his finite memory **w.s.**
is an implementation $\mathcal{S}$

Deciding if $G \in \mathbf{REG}$ is comeagre

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$.

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\big[G = \mathrm{L}(\mathcal{A}_G)\big]$

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\big[G = \mathrm{L}(\mathcal{A}_G)\big]$

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$:

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\left[G = \mathrm{L}(\mathcal{A}_G)\right]$

Construct a regular $W_G \subseteq \left(A \sqcup \{\flat\}\right)^\omega$: $\left[\mathcal{A}_G \mapsto \mathcal{A}_{W_G} \text{ s.t. } \mathrm{L}(\mathcal{A}_{W_G}) = W_G\right]$

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\big[G = \mathrm{L}(\mathcal{A}_G)\big]$

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$: $\big[\mathcal{A}_G \mapsto \mathcal{A}_{W_G}$ s.t. $\mathrm{L}(\mathcal{A}_{W_G}) = W_G\big]$

(I):
(II):

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\big[G = \mathrm{L}(\mathcal{A}_G)\big]$

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$: $\big[\mathcal{A}_G \mapsto \mathcal{A}_{W_G}$ s.t. $\mathrm{L}(\mathcal{A}_{W_G}) = W_G\big]$

(I): $\underline{a_0}$
(II):

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\big[G = \mathrm{L}(\mathcal{A}_G)\big]$

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$: $\big[\mathcal{A}_G \mapsto \mathcal{A}_{W_G}$ s.t. $\mathrm{L}(\mathcal{A}_{W_G}) = W_G\big]$

(I): $\underline{a_0}$
(II): $\underline{\flat}$

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\big[G = \mathrm{L}(\mathcal{A}_G)\big]$

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$: $\big[\mathcal{A}_G \mapsto \mathcal{A}_{W_G}$ s.t. $\mathrm{L}(\mathcal{A}_{W_G}) = W_G\big]$

(I): $\underline{a_0}$ $\underline{a_1}$
(II): $\underline{\flat}$

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\big[G = \mathrm{L}(\mathcal{A}_G)\big]$

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$: $\big[\mathcal{A}_G \mapsto \mathcal{A}_{W_G}$ s.t. $\mathrm{L}(\mathcal{A}_{W_G}) = W_G\big]$

(I): $\underline{a_0}$ $\underline{a_1}$
(II): $\underline{\flat}$ $\underline{\flat}$

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $[G = \mathrm{L}(\mathcal{A}_G)]$

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$: $[\mathcal{A}_G \mapsto \mathcal{A}_{W_G}$ s.t. $\mathrm{L}(\mathcal{A}_{W_G}) = W_G]$

(I):	$\underline{a_0}$		$\underline{a_1}$		$\underline{\flat}$
(II):		$\underline{\flat}$		$\underline{\flat}$	

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $[G = \mathrm{L}(\mathcal{A}_G)]$

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$: $[\mathcal{A}_G \mapsto \mathcal{A}_{W_G}$ s.t. $\mathrm{L}(\mathcal{A}_{W_G}) = W_G]$

(I): $\underline{a_0}$ $\underline{a_1}$ $\underline{\flat}$
(II): $\underline{\flat}$ $\underline{\flat}$ $\underline{a_2}$

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\big[G = \mathrm{L}(\mathcal{A}_G)\big]$

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$: $\big[\mathcal{A}_G \mapsto \mathcal{A}_{W_G}$ s.t. $\mathrm{L}(\mathcal{A}_{W_G}) = W_G\big]$

(I):	$\underline{a_0}$		$\underline{a_1}$		$\underline{\flat}$		$\underline{\flat}$
(II):		$\underline{\flat}$		$\underline{\flat}$		$\underline{a_2}$	

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\big[G = \mathrm{L}(\mathcal{A}_G)\big]$

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$: $\big[\mathcal{A}_G \mapsto \mathcal{A}_{W_G}$ s.t. $\mathrm{L}(\mathcal{A}_{W_G}) = W_G\big]$

(I):	$\underline{a_0}$		$\underline{a_1}$		$\underline{\flat}$		$\underline{\flat}$	
(II):		$\underline{\flat}$		$\underline{\flat}$		$\underline{a_2}$		$\underline{a_3}$

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\left[G = \mathrm{L}(\mathcal{A}_G)\right]$

Construct a regular $W_G \subseteq \left(A \sqcup \{\flat\}\right)^\omega$: $\left[\mathcal{A}_G \mapsto \mathcal{A}_{W_G} \text{ s.t. } \mathrm{L}(\mathcal{A}_{W_G}) = W_G\right]$

(I):	$\underline{a_0}$		$\underline{a_1}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{\flat}$
(II):		$\underline{\flat}$		$\underline{\flat}$		$\underline{a_2}$		$\underline{a_3}$	

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\big[G = \mathrm{L}(\mathcal{A}_G)\big]$

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$: $\big[\mathcal{A}_G \mapsto \mathcal{A}_{W_G}$ s.t. $\mathrm{L}(\mathcal{A}_{W_G}) = W_G\big]$

(I): $\underline{a_0}$ $\underline{a_1}$ $\underline{\flat}$ $\underline{\flat}$ $\underline{\flat}$

(II): $\underline{\flat}$ $\underline{\flat}$ $\underline{a_2}$ $\underline{a_3}$ $\underline{a_4}$

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\big[G = \mathrm{L}(\mathcal{A}_G)\big]$

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$: $\big[\mathcal{A}_G \mapsto \mathcal{A}_{W_G}$ s.t. $\mathrm{L}(\mathcal{A}_{W_G}) = W_G\big]$

(I):	$\underline{a_0}$		$\underline{a_1}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{\flat}$
(II):		$\underline{\flat}$		$\underline{\flat}$		$\underline{a_2}$		$\underline{a_3}$		$\underline{a_4}$	

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\big[G = \mathrm{L}(\mathcal{A}_G)\big]$

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$: $\big[\mathcal{A}_G \mapsto \mathcal{A}_{W_G}$ s.t. $\mathrm{L}(\mathcal{A}_{W_G}) = W_G\big]$

(I): $\underline{a_0}$ $\underline{a_1}$ $\underline{\flat}$ $\underline{\flat}$ $\underline{\flat}$ $\underline{\flat}$
(II): $\underline{\flat}$ $\underline{\flat}$ $\underline{a_2}$ $\underline{a_3}$ $\underline{a_4}$ $\underline{a_5}$

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\big[G = \mathrm{L}(\mathcal{A}_G)\big]$

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$: $\big[\mathcal{A}_G \mapsto \mathcal{A}_{W_G}$ s.t. $\mathrm{L}(\mathcal{A}_{W_G}) = W_G\big]$

(I):	$\underline{a_0}$		$\underline{a_1}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{\flat}$
(II):		$\underline{\flat}$		$\underline{\flat}$		$\underline{a_2}$		$\underline{a_3}$		$\underline{a_4}$		$\underline{a_5}$	

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\big[G = \mathrm{L}(\mathcal{A}_G)\big]$

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$: $\big[\mathcal{A}_G \mapsto \mathcal{A}_{W_G}$ s.t. $\mathrm{L}(\mathcal{A}_{W_G}) = W_G\big]$

(I):	$\underline{a_0}$		$\underline{a_1}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{\flat}$	$\underline{\flat}$
(II):		$\underline{\flat}$		$\underline{\flat}$		$\underline{a_2}$		$\underline{a_3}$		$\underline{a_4}$	$\underline{a_5}$	$\underline{\flat}$

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\big[G = \mathrm{L}(\mathcal{A}_G)\big]$

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$: $\big[\mathcal{A}_G \mapsto \mathcal{A}_{W_G}$ s.t. $\mathrm{L}(\mathcal{A}_{W_G}) = W_G\big]$

(I):	$\underline{a_0}$		$\underline{a_1}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{a_6}$
(II):		$\underline{\flat}$		$\underline{\flat}$		$\underline{a_2}$		$\underline{a_3}$		$\underline{a_4}$		$\underline{a_5}$		$\underline{\flat}$	

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\quad \big[G = \mathrm{L}(\mathcal{A}_G)\big]$

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$: $\quad \big[\mathcal{A}_G \mapsto \mathcal{A}_{W_G}$ s.t. $\mathrm{L}(\mathcal{A}_{W_G}) = W_G\big]$

(I):	$\underline{a_0}$		$\underline{a_1}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{a_6}$	$\dots$
(II):		$\underline{\flat}$		$\underline{\flat}$		$\underline{a_2}$		$\underline{a_3}$		$\underline{a_4}$		$\underline{a_5}$		$\underline{\flat}$		

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\big[G = \mathrm{L}(\mathcal{A}_G)\big]$

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$: $\big[\mathcal{A}_G \mapsto \mathcal{A}_{W_G}$ s.t. $\mathrm{L}(\mathcal{A}_{W_G}) = W_G\big]$

(I):	$\underline{a_0}$		$\underline{a_1}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{a_6}$		$\dots$
(II):		$\underline{\flat}$		$\underline{\flat}$		$\underline{a_2}$		$\underline{a_3}$		$\underline{a_4}$		$\underline{a_5}$		$\underline{\flat}$			

$$\big(\text{(II) wins } \mathrm{BM}(G)\big) \iff \big(\text{(II) wins } \mathcal{G}(W_G)\big)$$

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\left[G = \mathrm{L}(\mathcal{A}_G)\right]$

Construct a regular $W_G \subseteq \left(A \sqcup \{\flat\}\right)^\omega$: $\left[\mathcal{A}_G \mapsto \mathcal{A}_{W_G} \text{ s.t. } \mathrm{L}(\mathcal{A}_{W_G}) = W_G\right]$

(I):	$\underline{a_0}$		$\underline{a_1}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{a_6}$	$\dots$
(II):		$\underline{\flat}$		$\underline{\flat}$		$\underline{a_2}$		$\underline{a_3}$		$\underline{a_4}$		$\underline{a_5}$		$\underline{\flat}$		

$$\big(\text{(II) wins } \mathrm{BM}(G)\big) \iff \big(\text{(II) wins } \mathcal{G}(W_G)\big)$$

Solve $\mathcal{G}(W_G)$ to know if G is comeagre. ∎

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\big[G = \mathrm{L}(\mathcal{A}_G)\big]$

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$: $\big[\mathcal{A}_G \mapsto \mathcal{A}_{W_G}$ s.t. $\mathrm{L}(\mathcal{A}_{W_G}) = W_G\big]$

(I):	a_0		a_1		$\flat$		$\flat$		$\flat$		$\flat$		$\flat$		a_6	...
(II):		$\flat$		$\flat$		a_2		a_3		a_4		a_5		$\flat$		

$$\big(\text{(II) wins } \mathrm{BM}(G)\big) \iff \big(\text{(II) wins } \mathcal{G}(W_G)\big)$$

Solve $\mathcal{G}(W_G)$ to know if G is comeagre. ■

Similarly with other game-characterised properties for regular sets:

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\big[G = \mathrm{L}(\mathcal{A}_G)\big]$

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$: $\big[\mathcal{A}_G \mapsto \mathcal{A}_{W_G}$ s.t. $\mathrm{L}(\mathcal{A}_{W_G}) = W_G\big]$

(I):	a_0		a_1		$\flat$		$\flat$		$\flat$		$\flat$		$\flat$		a_6		...
(II):		$\flat$		$\flat$		a_2		a_3		a_4		a_5		$\flat$			

$$\big(\text{(II) wins } \mathrm{BM}(G)\big) \iff \big(\text{(II) wins } \mathcal{G}(W_G)\big)$$

Solve $\mathcal{G}(W_G)$ to know if G is comeagre. ■

Similarly with other game-characterised properties for regular sets:

— countability,

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\big[G = \mathrm{L}(\mathcal{A}_G)\big]$

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$: $\big[\mathcal{A}_G \mapsto \mathcal{A}_{W_G}$ s.t. $\mathrm{L}(\mathcal{A}_{W_G}) = W_G\big]$

(I):	$\underline{a_0}$		$\underline{a_1}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{\flat}$		$\underline{a_6}$	$\dots$
(II):		$\underline{\flat}$		$\underline{\flat}$		$\underline{a_2}$		$\underline{a_3}$		$\underline{a_4}$		$\underline{a_5}$		$\underline{\flat}$		

$$\big(\text{(II) wins } \mathrm{BM}(G)\big) \iff \big(\text{(II) wins } \mathcal{G}(W_G)\big)$$

Solve $\mathcal{G}(W_G)$ to know if G is comeagre. ■

Similarly with other game-characterised properties for regular sets:

— countability,

— measure 0,

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\big[G = \mathrm{L}(\mathcal{A}_G)\big]$

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$: $\big[\mathcal{A}_G \mapsto \mathcal{A}_{W_G}$ s.t. $\mathrm{L}(\mathcal{A}_{W_G}) = W_G\big]$

$$\begin{array}{lcccccccccccccccc} \text{(I):} & \underline{a_0} & & \underline{a_1} & & \underline{\flat} & & \underline{\flat} & & \underline{\flat} & & \underline{\flat} & & \underline{\flat} & & \underline{a_6} & \\ \text{(II):} & & \underline{\flat} & & \underline{\flat} & & \underline{a_2} & & \underline{a_3} & & \underline{a_4} & & \underline{a_5} & & \underline{\flat} & & \ldots \end{array}$$

$$\big(\text{(II) wins } \mathrm{BM}(G)\big) \iff \big(\text{(II) wins } \mathcal{G}(W_G)\big)$$

Solve $\mathcal{G}(W_G)$ to know if G is comeagre. ∎

Similarly with other game-characterised properties for regular sets:

- countability,
- measure 0,
- Wadge reductions (in a moment), ...

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\big[G = \mathrm{L}(\mathcal{A}_G)\big]$

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$: $\big[\mathcal{A}_G \mapsto \mathcal{A}_{W_G}$ s.t. $\mathrm{L}(\mathcal{A}_{W_G}) = W_G\big]$

$$
\begin{array}{lccccccccccccccc}
\text{(I):} & \underline{a_0} & & \underline{a_1} & & \underline{\flat} & & \underline{\flat} & & \underline{\flat} & & \underline{\flat} & & \underline{\flat} & & \underline{a_6} \quad \ldots \\
\text{(II):} & & \underline{\flat} & & \underline{\flat} & & \underline{a_2} & & \underline{a_3} & & \underline{a_4} & & \underline{a_5} & & \underline{\flat} &
\end{array}
$$

$$\big(\text{(II) wins } \mathrm{BM}(G)\big) \iff \big(\text{(II) wins } \mathcal{G}(W_G)\big)$$

Solve $\mathcal{G}(W_G)$ to know if G is comeagre. ■

Similarly with other game-characterised properties for regular sets:

— countability,

— measure 0,

— Wadge reductions (in a moment), ...

Sometimes works even for infinite trees:

Deciding if $G \in \mathbf{REG}$ is comeagre

Take a regular $G \subseteq A^\omega$. $\quad [G = \mathrm{L}(\mathcal{A}_G)]$

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$: $\quad \big[\mathcal{A}_G \mapsto \mathcal{A}_{W_G}$ s.t. $\mathrm{L}(\mathcal{A}_{W_G}) = W_G\big]$

$$
\begin{array}{lcccccccccccccc}
\text{(I)}: & \underline{a_0} & & \underline{a_1} & & \underline{\flat} & & \underline{\flat} & & \underline{\flat} & & \underline{\flat} & & \underline{\flat} & & \underline{a_6} & \dots \\
\text{(II)}: & & \underline{\flat} & & \underline{\flat} & & \underline{a_2} & & \underline{a_3} & & \underline{a_4} & & \underline{a_5} & & \underline{\flat} & &
\end{array}
$$

$$\big(\text{(II) wins } \mathrm{BM}(G)\big) \iff \big(\text{(II) wins } \mathcal{G}(W_G)\big)$$

Solve $\mathcal{G}(W_G)$ to know if G is comeagre. ■

Similarly with other game-characterised properties for regular sets:

— countability,

— measure 0,

— Wadge reductions (in a moment), ...

Sometimes works even for infinite trees:

Theorem (Michalewski, Mio, S. ['17])

It is decidable if $\mathrm{L}(\mathcal{A})$ is comeagre for game-automata $\mathcal{A}$.

Wadge order for regular languages

Wadge order for regular languages

Input: Regular $K \subseteq A^\omega$ and $L \subseteq B^\omega$

Wadge order for regular languages

Input: Regular $K \subseteq A^\omega$ and $L \subseteq B^\omega$

Output: Does $K \leqslant_{\mathrm{W}} L$?

Wadge order for regular languages

Input: Regular $K \subseteq A^\omega$ and $L \subseteq B^\omega$

Output: Does $K \leqslant_{\mathrm{W}} L$?

Wadge game $\mathcal{W}(K, L)$:

Wadge order for regular languages

Input: Regular $K \subseteq A^\omega$ and $L \subseteq B^\omega$

Output: Does $K \leqslant_{\mathrm{W}} L$?

Wadge game $\mathcal{W}(K, L)$:

(I):

(II):

Wadge order for regular languages

Input: Regular $K \subseteq A^\omega$ and $L \subseteq B^\omega$

Output: Does $K \leqslant_{\mathrm{W}} L$?

Wadge game $\mathcal{W}(K, L)$:

(I): a_0

(II):

Wadge order for regular languages

Input: Regular $K \subseteq A^\omega$ and $L \subseteq B^\omega$

Output: Does $K \leqslant_{\mathrm{W}} L$?

Wadge game $\mathcal{W}(K, L)$:

(I): $a_0 \in A$

(II):

Wadge order for regular languages

Input: Regular $K \subseteq A^\omega$ and $L \subseteq B^\omega$

Output: Does $K \leqslant_{\mathrm{W}} L$?

Wadge game $\mathcal{W}(K, L)$:

$$
\begin{array}{ll}
 & \quad \in A \\
(\mathrm{I}): & a_0 \\
(\mathrm{II}): & b_0
\end{array}
$$

Wadge order for regular languages

Input: Regular $K \subseteq A^\omega$ and $L \subseteq B^\omega$

Output: Does $K \leqslant_{\mathrm{W}} L$?

Wadge game $\mathcal{W}(K, L)$:

$$\begin{array}{ll} & \quad \in A \\ \text{(I):} & a_0 \\ \text{(II):} & b_0 \\ & \quad \in B \sqcup \{\epsilon\} \end{array}$$

Wadge order for regular languages

Input: Regular $K \subseteq A^\omega$ and $L \subseteq B^\omega$

Output: Does $K \leqslant_{\mathrm{W}} L$?

Wadge game $\mathcal{W}(K, L)$:

$$\begin{array}{lll} & \in A & \\ \text{(I)}: & a_0 & a_1 \\ \text{(II)}: & b_0 & \\ & \in B \sqcup \{\epsilon\} & \end{array}$$

Wadge order for regular languages

Input: Regular $K \subseteq A^\omega$ and $L \subseteq B^\omega$

Output: Does $K \leqslant_{\mathrm{W}} L$?

Wadge game $\mathcal{W}(K, L)$:

$$\begin{array}{lcc} & \in A & \\ \text{(I)}: & a_0 & a_1 \\ \text{(II)}: & b_0 & b_1 \\ & \in B \sqcup \{\epsilon\} & \end{array}$$

Wadge order for regular languages

Input: Regular $K \subseteq A^\omega$ and $L \subseteq B^\omega$

Output: Does $K \leqslant_{\mathrm{W}} L$?

Wadge game $\mathcal{W}(K, L)$:

	$\in A$				
(I):	a_0	a_1	a_2	a_3	a_4
(II):	b_0	b_1	b_2	b_3	b_4
	$\in B \sqcup \{\epsilon\}$				

Wadge order for regular languages

Input: Regular $K \subseteq A^\omega$ and $L \subseteq B^\omega$

Output: Does $K \leqslant_{\mathrm{W}} L$?

Wadge game $\mathcal{W}(K, L)$:

$$
\begin{array}{lllllllll}
 & \overset{A}{\ni} & & & & & & & \\
\text{(I)}: & a_0 & a_1 & a_2 & a_3 & a_4 & \cdots & \rightsquigarrow \alpha \in A^\omega \\
\text{(II)}: & b_0 & b_1 & b_2 & b_3 & b_4 & \cdots & \rightsquigarrow \beta \in B^{\leqslant\omega} \\
 & \in B \sqcup \{\epsilon\} & & & & & & &
\end{array}
$$

Wadge order for regular languages

Input: Regular $K \subseteq A^\omega$ and $L \subseteq B^\omega$

Output: Does $K \leqslant_{\mathrm{W}} L$?

Wadge game $\mathcal{W}(K, L)$:

$$
\begin{array}{lllllll}
 & \overset{A}{\in} & & & & & \\
\text{(I):} & a_0 & a_1 & a_2 & a_3 & a_4 & \cdots \rightsquigarrow \alpha \in A^\omega \\
\text{(II):} & b_0 & b_1 & b_2 & b_3 & b_4 & \cdots \rightsquigarrow \beta \in B^{\leqslant\omega} \\
 & \underset{B \sqcup \{\epsilon\}}{\in} & & & & &
\end{array}
$$

$$W \equiv \beta \in B^\omega \wedge \big(\alpha \in L \Leftrightarrow \beta \in K\big)$$

Wadge order for regular languages

Input: Regular $K \subseteq A^\omega$ and $L \subseteq B^\omega$

Output: Does $K \leqslant_{\mathrm{W}} L$?

Wadge game $\mathcal{W}(K, L)$:

	$\in A$						
(I):	a_0	a_1	a_2	a_3	a_4	$\cdots$	$\rightsquigarrow \alpha \in A^\omega$
(II):	b_0	b_1	b_2	b_3	b_4	$\cdots$	$\rightsquigarrow \beta \in B^{\leqslant\omega}$
	$\in B \sqcup \{\epsilon\}$						

$$W \equiv \underbrace{\beta \in B^\omega \wedge \big(\alpha \in L \Leftrightarrow \beta \in K\big)}_{\text{regular property over } A \cup B \sqcup \{\epsilon\}}$$

Wadge order for regular languages

Input: Regular $K \subseteq A^\omega$ and $L \subseteq B^\omega$

Output: Does $K \leqslant_{\mathrm{W}} L$?

Wadge game $\mathcal{W}(K, L)$:

$$
\begin{array}{lllllllll}
 & \overset{\in A}{} & & & & & & & \\
\text{(I):} & a_0 & a_1 & a_2 & a_3 & a_4 & \cdots & \rightsquigarrow \alpha \in A^\omega \\
\text{(II):} & b_0 & b_1 & b_2 & b_3 & b_4 & \cdots & \rightsquigarrow \beta \in B^{\leqslant\omega} \\
 & \underset{\in B \sqcup \{\epsilon\}}{} & & & & & & &
\end{array}
$$

$$W \equiv \underbrace{\beta \in B^\omega \wedge \big(\alpha \in L \Leftrightarrow \beta \in K\big)}_{\text{regular property over } A \cup B \sqcup \{\epsilon\}}$$

Effectively solve $\mathcal{W}(K, L)$ to know if $K \leqslant_{\mathrm{W}} L$.

Parity index

Parity index

Fix a pair $i \leqslant j$.

Parity index

Fix a pair $i \leqslant j$.

$$P_{i,j} \stackrel{\text{def}}{=} \left\{\tau \in \{i, \ldots, j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\right\}$$

Parity index

Fix a pair $i \leqslant j$.

$$P_{i,j} \stackrel{\text{def}}{=} \{\tau \in \{i, \ldots, j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\}$$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** (i, j)-parity automaton

Parity index

Fix a pair $i \leqslant j$.

$$P_{i,j} \stackrel{\text{def}}{=} \{\tau \in \{i,\ldots,j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\}$$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\overbrace{(i,j)\text{-parity}}^{\Omega\colon Q \to \{i,\ldots,j\}}$ automaton

Parity index

Fix a pair $i \leqslant j$.

$$P_{i,j} \stackrel{\text{def}}{=} \left\{\tau \in \{i, \ldots, j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\right\}$$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\overbrace{(i,j)\text{-parity}}^{\Omega\colon Q \to \{i,\ldots,j\}}$ automaton then $L \leqslant_{\mathrm{W}} P_{i,j}$.

Parity index

Fix a pair $i \leqslant j$.

$$P_{i,j} \stackrel{\text{def}}{=} \{\tau \in \{i, \ldots, j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\}$$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\underbrace{(i,j)\text{-parity}}_{\Omega\colon Q \to \{i,\ldots,j\}}$ automaton then $L \leqslant_{\mathrm{W}} P_{i,j}$.

Proof

$\mathcal{A}$ reads $\alpha = a_0 a_1 \cdots$ and produces $\rho = q_0 q_1 \cdots$

Parity index

Fix a pair $i \leqslant j$.

$P_{i,j} \stackrel{\text{def}}{=} \{\tau \in \{i, \ldots, j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\}$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\overbrace{(i,j)\text{-parity}}^{\Omega\colon Q \to \{i,\ldots,j\}}$ automaton then $L \leqslant_{\mathrm{W}} P_{i,j}$.

Proof

$\mathcal{A}$ reads $\alpha = a_0 a_1 \cdots$ and produces $\rho = q_0 q_1 \cdots$

$\alpha = \quad a_0 \quad a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6 \quad \cdots$

Parity index

Fix a pair $i \leqslant j$.

$$P_{i,j} \stackrel{\text{def}}{=} \left\{\tau \in \{i,\ldots,j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\right\}$$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\overbrace{(i,j)\text{-parity}}^{\Omega\colon Q \to \{i,\ldots,j\}}$ automaton then $L \leqslant_{\mathrm{W}} P_{i,j}$.

Proof

$\mathcal{A}$ reads $\alpha = a_0 a_1 \cdots$ and produces $\rho = q_0 q_1 \cdots$

$$\alpha = \quad a_0 \quad a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6 \quad \cdots$$

$$\rho = q_0 \to q_1 \to q_2 \to q_3 \to q_4 \to q_5 \to q_6 \to q_7 \quad \cdots$$

Parity index

Fix a pair $i \leqslant j$.

$$P_{i,j} \overset{\text{def}}{=} \left\{\tau \in \{i,\ldots,j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\right\}$$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\overbrace{(i,j)\text{-parity}}^{\Omega\colon Q \to \{i,\ldots,j\}}$ automaton then $L \leqslant_{\mathrm{W}} P_{i,j}$.

Proof

$\mathcal{A}$ reads $\alpha = a_0 a_1 \cdots$ and produces $\rho = q_0 q_1 \cdots$

$$\begin{array}{rccccccccc}
\alpha = & & a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & \cdots \\
\rho = & q_0 \curvearrowright & q_1 \curvearrowright & q_2 \curvearrowright & q_3 \curvearrowright & q_4 \curvearrowright & q_5 \curvearrowright & q_6 \curvearrowright & q_7 & \cdots \\
\Omega\colon & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \\
\tau = & 1 & 2 & 0 & 2 & 1 & 0 & 2 & 1 & \cdots
\end{array}$$

Parity index

Fix a pair $i \leqslant j$.

$$P_{i,j} \stackrel{\text{def}}{=} \left\{\tau \in \{i, \ldots, j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\right\}$$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\overbrace{(i,j)\text{-parity}}^{\Omega\colon Q \to \{i,\ldots,j\}}$ automaton then $L \leqslant_{\mathrm{W}} P_{i,j}$.

Proof

$\mathcal{A}$ reads $\alpha = a_0 a_1 \cdots$ and produces $\rho = q_0 q_1 \cdots$

$$\begin{array}{rccccccccc}
\alpha = & & a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & \cdots \\
\rho = & q_0 \curvearrowright & q_1 \curvearrowright & q_2 \curvearrowright & q_3 \curvearrowright & q_4 \curvearrowright & q_5 \curvearrowright & q_6 \curvearrowright & q_7 & \cdots \\
\Omega\colon & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \\
\tau = & 1 & 2 & 0 & 2 & 1 & 0 & 2 & 1 & \cdots
\end{array}$$

$$\alpha \in \mathrm{L}(\mathcal{A}) \quad \text{iff} \quad \tau \in P_{i,j}$$

■

Parity index

Fix a pair $i \leqslant j$.

$$P_{i,j} \stackrel{\text{def}}{=} \{\tau \in \{i, \ldots, j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\}$$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\overbrace{(i,j)\text{-parity}}^{\Omega\colon Q \to \{i,\ldots,j\}}$ automaton then $L \leqslant_{\mathrm{W}} P_{i,j}$.

Parity index

Fix a pair $i \leqslant j$.

$$P_{i,j} \stackrel{\text{def}}{=} \left\{\tau \in \{i, \ldots, j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\right\}$$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\overbrace{(i,j)\text{-parity}}^{\Omega\colon Q \to \{i,\ldots,j\}}$ automaton then $L \leqslant_{\mathrm{W}} P_{i,j}$.

Proposition

If $L \leqslant_{\mathrm{W}} P_{i,j}$ and $L \in \mathbf{REG}$

Parity index

Fix a pair $i \leqslant j$.

$$P_{i,j} \stackrel{\text{def}}{=} \big\{\tau \in \{i, \ldots, j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\big\}$$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\overbrace{(i,j)\text{-parity}}^{\Omega\colon Q \to \{i,\ldots,j\}}$ automaton then $L \leqslant_{\mathrm{W}} P_{i,j}$.

Proposition

If $L \leqslant_{\mathrm{W}} P_{i,j}$ and $L \in \mathbf{REG}$ then $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** (i, j)-parity.

Parity index

Fix a pair $i \leqslant j$.

$$P_{i,j} \stackrel{\text{def}}{=} \{\tau \in \{i, \ldots, j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\}$$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\overbrace{(i,j)\text{-parity}}^{\Omega\colon Q \to \{i,\ldots,j\}}$ automaton then $L \leqslant_{\mathrm{W}} P_{i,j}$.

Proposition

If $L \leqslant_{\mathrm{W}} P_{i,j}$ and $L \in \mathbf{REG}$ then $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** (i,j)-parity.

Proof — game:

Parity index

Fix a pair $i \leqslant j$.

$$P_{i,j} \stackrel{\text{def}}{=} \{\tau \in \{i, \ldots, j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\}$$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\overbrace{(i,j)\text{-parity}}^{\Omega\colon Q \to \{i,\ldots,j\}}$ automaton then $L \leqslant_{\mathrm{W}} P_{i,j}$.

Proposition

If $L \leqslant_{\mathrm{W}} P_{i,j}$ and $L \in \mathbf{REG}$ then $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** (i,j)-parity.

Proof — game:

(I):

(II):

Parity index

Fix a pair $i \leqslant j$.

$$P_{i,j} \stackrel{\text{def}}{=} \{\tau \in \{i,\ldots,j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\}$$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\underbrace{(i,j)\text{-parity}}_{\Omega\colon Q \to \{i,\ldots,j\}}$ automaton then $L \leqslant_{\mathrm{W}} P_{i,j}$.

Proposition

If $L \leqslant_{\mathrm{W}} P_{i,j}$ and $L \in \mathbf{REG}$ then $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** (i,j)-parity.

Proof — game:

(I): a_0

(II):

Parity index

Fix a pair $i \leqslant j$.

$$P_{i,j} \stackrel{\text{def}}{=} \left\{\tau \in \{i, \ldots, j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\right\}$$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\overbrace{(i,j)\text{-parity}}^{\Omega\colon Q \to \{i,\ldots,j\}}$ automaton then $L \leqslant_{\mathrm{W}} P_{i,j}$.

Proposition

If $L \leqslant_{\mathrm{W}} P_{i,j}$ and $L \in \mathbf{REG}$ then $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** (i,j)-parity.

Proof — game:

(I): $a_0 \in A$

(II):

Parity index

Fix a pair $i \leqslant j$.

$P_{i,j} \overset{\text{def}}{=} \{\tau \in \{i, \ldots, j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\}$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\overbrace{(i,j)\text{-parity}}^{\Omega\colon Q \to \{i,\ldots,j\}}$ automaton then $L \leqslant_{\mathrm{W}} P_{i,j}$.

Proposition

If $L \leqslant_{\mathrm{W}} P_{i,j}$ and $L \in \mathbf{REG}$ then $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** (i,j)-parity.

Proof — game:

(I): $a_0 \in A$

(II): p_0

Parity index

Fix a pair $i \leqslant j$.

$$P_{i,j} \stackrel{\text{def}}{=} \left\{\tau \in \{i,\ldots,j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\right\}$$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\overbrace{(i,j)\text{-parity}}^{\Omega\colon Q \to \{i,\ldots,j\}}$ automaton then $L \leqslant_{\mathrm{W}} P_{i,j}$.

Proposition

If $L \leqslant_{\mathrm{W}} P_{i,j}$ and $L \in \mathbf{REG}$ then $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** (i,j)-parity.

Proof — game:

(I): $\underset{\in A}{a_0}$

(II): $\underset{\in \{i,\ldots,j\}}{p_0}$

Parity index

Fix a pair $i \leqslant j$.

$$P_{i,j} \stackrel{\text{def}}{=} \left\{\tau \in \{i,\ldots,j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\right\}$$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\overbrace{(i,j)\text{-parity}}^{\Omega\colon Q \to \{i,\ldots,j\}}$ automaton then $L \leqslant_{\mathrm{W}} P_{i,j}$.

Proposition

If $L \leqslant_{\mathrm{W}} P_{i,j}$ and $L \in \mathbf{REG}$ then $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** (i,j)-parity.

Proof — game:

(I): $a_0 \in A$ $\quad a_1$

(II): $p_0 \in \{i,\ldots,j\}$

Parity index

Fix a pair $i \leqslant j$.

$$P_{i,j} \stackrel{\text{def}}{=} \{\tau \in \{i,\ldots,j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\}$$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\overbrace{(i,j)\text{-parity}}^{\Omega\colon Q \to \{i,\ldots,j\}}$ automaton then $L \leqslant_{\mathrm{W}} P_{i,j}$.

Proposition

If $L \leqslant_{\mathrm{W}} P_{i,j}$ and $L \in \mathbf{REG}$ then $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** (i,j)-parity.

Proof — game:

$a_0 \in A$

(I): a_0 a_1

(II): p_0 p_1

$p_0 \in \{i,\ldots,j\}$

Parity index

Fix a pair $i \leqslant j$.

$$P_{i,j} \stackrel{\text{def}}{=} \{\tau \in \{i,\ldots,j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\}$$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\overbrace{(i,j)\text{-parity}}^{\Omega\colon Q\to\{i,\ldots,j\}}$ automaton then $L \leqslant_{\mathrm{W}} P_{i,j}$.

Proposition

If $L \leqslant_{\mathrm{W}} P_{i,j}$ and $L \in \mathbf{REG}$ then $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** (i,j)-parity.

Proof — game:

$$a_0 \in A$$

(I): a_0 a_1 a_2 a_3 a_4

(II): p_0 p_1 p_2 p_3 p_4

$$p_0 \in \{i,\ldots,j\}$$

Parity index

Fix a pair $i \leqslant j$.

$$P_{i,j} \stackrel{\text{def}}{=} \left\{\tau \in \{i, \ldots, j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\right\}$$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\underbrace{(i,j)\text{-parity}}_{\Omega\colon Q \to \{i,\ldots,j\}}$ automaton then $L \leqslant_{\mathrm{W}} P_{i,j}$.

Proposition

If $L \leqslant_{\mathrm{W}} P_{i,j}$ and $L \in \mathbf{REG}$ then $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** (i,j)-parity.

Proof — game:

$$\begin{array}{lcccccll} & \in A & & & & & & \\ (\mathrm{I}): & a_0 & a_1 & a_2 & a_3 & a_4 & \cdots & \rightsquigarrow \alpha \in A^\omega \\ (\mathrm{II}): & p_0 & p_1 & p_2 & p_3 & p_4 & \cdots & \rightsquigarrow \tau \in \{i,\ldots,j\}^\omega \\ & \in \{i,\ldots,j\} & & & & & & \end{array}$$

Parity index

Fix a pair $i \leqslant j$.

$$P_{i,j} \stackrel{\text{def}}{=} \left\{\tau \in \{i, \ldots, j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\right\}$$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\overbrace{(i,j)\text{-parity}}^{\Omega\colon Q \to \{i,\ldots,j\}}$ automaton then $L \leqslant_{\mathrm{W}} P_{i,j}$.

Proposition

If $L \leqslant_{\mathrm{W}} P_{i,j}$ and $L \in \mathbf{REG}$ then $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** (i,j)-parity.

Proof — game:

$$\begin{array}{lcccccc} & \in A & & & & & \\ (\mathrm{I}): & a_0 & a_1 & a_2 & a_3 & a_4 & \bullet\bullet\bullet \rightsquigarrow \alpha \in A^\omega \\ (\mathrm{II}): & p_0 & p_1 & p_2 & p_3 & p_4 & \bullet\bullet\bullet \rightsquigarrow \tau \in \{i,\ldots,j\}^\omega \\ & \in \{i,\ldots,j\} & & & & & \end{array}$$

$$W \equiv \alpha \in L \Leftrightarrow \tau \in P_{i,j}$$

Parity index

Fix a pair $i \leqslant j$.

$$P_{i,j} \stackrel{\text{def}}{=} \{\tau \in \{i,\ldots,j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\}$$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\overbrace{(i,j)\text{-parity}}^{\Omega\colon Q \to \{i,\ldots,j\}}$ automaton then $L \leqslant_{\mathrm{W}} P_{i,j}$.

Proposition

If $L \leqslant_{\mathrm{W}} P_{i,j}$ and $L \in \mathbf{REG}$ then $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** (i,j)-parity.

Proof — game:

$a_0 \in A$

(I): a_0 a_1 a_2 a_3 a_4 $\cdots$ $\rightsquigarrow \alpha \in A^\omega$

(II): p_0 p_1 p_2 p_3 p_4 $\cdots$ $\rightsquigarrow \tau \in \{i,\ldots,j\}^\omega$

$p_0 \in \{i,\ldots,j\}$

$W \equiv \alpha \in L \Leftrightarrow \tau \in P_{i,j}$ — regular condition

Parity index

Fix a pair $i \leqslant j$.

$P_{i,j} \stackrel{\text{def}}{=} \{\tau \in \{i,\ldots,j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\}$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\overbrace{(i,j)\text{-parity}}^{\Omega\colon Q\to\{i,\ldots,j\}}$ automaton then $L \leqslant_{\mathrm{W}} P_{i,j}$.

Proposition

If $L \leqslant_{\mathrm{W}} P_{i,j}$ and $L \in \mathbf{REG}$ then $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** (i,j)-parity.

Proof — game:

$$
\begin{array}{lllllllll}
 & a_0 \in A & & & & & & \\
(\mathrm{I}): & a_0 & a_1 & a_2 & a_3 & a_4 & \cdots & \rightsquigarrow \alpha \in A^\omega \\
(\mathrm{II}): & p_0 & p_1 & p_2 & p_3 & p_4 & \cdots & \rightsquigarrow \tau \in \{i,\ldots,j\}^\omega \\
 & p_0 \in \{i,\ldots,j\} & & & & & &
\end{array}
$$

$W \equiv \alpha \in L \Leftrightarrow \tau \in P_{i,j}$ — regular condition

1. (II) wins the game (because $L \leqslant_{\mathrm{W}} P_{i,j}$).

Parity index

Fix a pair $i \leqslant j$.

$$P_{i,j} \stackrel{\text{def}}{=} \{\tau \in \{i,\ldots,j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\}$$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\underbrace{(i,j)\text{-parity}}_{\Omega\colon Q \to \{i,\ldots,j\}}$ automaton then $L \leqslant_{\mathrm{W}} P_{i,j}$.

Proposition

If $L \leqslant_{\mathrm{W}} P_{i,j}$ and $L \in \mathbf{REG}$ then $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** (i,j)-parity.

Proof — game:

$a_0 \in A$, $p_0 \in \{i,\ldots,j\}$

(I): $a_0 \quad a_1 \quad a_2 \quad a_3 \quad a_4 \quad \cdots \rightsquigarrow \alpha \in A^\omega$

(II): $p_0 \quad p_1 \quad p_2 \quad p_3 \quad p_4 \quad \cdots \rightsquigarrow \tau \in \{i,\ldots,j\}^\omega$

$$W \equiv \alpha \in L \Leftrightarrow \tau \in P_{i,j} \text{ — regular condition}$$

1. (II) wins the game (because $L \leqslant_{\mathrm{W}} P_{i,j}$).

2. So (II) wins using finite memory

Parity index

Fix a pair $i \leqslant j$.

$$P_{i,j} \stackrel{\text{def}}{=} \{\tau \in \{i, \ldots, j\}^\omega \mid \limsup_{n\to\infty} \tau(n) \equiv 0 \pmod 2\}$$

Fact

If $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** $\overbrace{(i,j)\text{-parity}}^{\Omega\colon Q \to \{i,\ldots,j\}}$ automaton then $L \leqslant_{\mathrm{W}} P_{i,j}$.

Proposition

If $L \leqslant_{\mathrm{W}} P_{i,j}$ and $L \in \mathbf{REG}$ then $L = \mathrm{L}(\mathcal{A})$ with $\mathcal{A}$ **det.** (i,j)-parity.

Proof — game:

$$\begin{array}{lllllll} & a_0 \in A & & & & & \\ \text{(I)}: & a_0 & a_1 & a_2 & a_3 & a_4 & \cdots \rightsquigarrow \alpha \in A^\omega \\ \text{(II)}: & p_0 & p_1 & p_2 & p_3 & p_4 & \cdots \rightsquigarrow \tau \in \{i,\ldots,j\}^\omega \\ & p_0 \in \{i,\ldots,j\} & & & & & \end{array}$$

$W \equiv \alpha \in L \Leftrightarrow \tau \in P_{i,j}$ — regular condition

1. (II) wins the game (because $L \leqslant_{\mathrm{W}} P_{i,j}$).
2. So (II) wins using finite memory $\rightsquigarrow$ **det.** (i,j)-parity aut. for L. ■

Trivia:

Trivia: $P_{i,j} \leqslant_{\mathrm{W}} P_{i,j+1}$,

Trivia: $P_{i,j} \leqslant_{\mathrm{W}} P_{i,j+1}, \quad P_{i,j} \equiv_{\mathrm{W}} P_{i+2,j+2},$

Trivia: $P_{i,j} \leqslant_{\mathrm{W}} P_{i,j+1}, \quad P_{i,j} \equiv_{\mathrm{W}} P_{i+2,j+2}, \quad P_{i,j} \equiv_{\mathrm{W}} P^{\mathrm{c}}_{i+1,j+1}$

Trivia: $P_{i,j} \leqslant_{\mathrm{W}} P_{i,j+1}, \quad P_{i,j} \equiv_{\mathrm{W}} P_{i+2,j+2}, \quad P_{i,j} \equiv_{\mathrm{W}} P^{\mathrm{c}}_{i+1,j+1}$

$$P_{0,0} \quad P_{0,1} \quad P_{0,2} \quad P_{0,3} \quad P_{0,4} \quad P_{0,5} \quad P_{0,6} \quad P_{0,7} \quad P_{0,8} \quad P_{0,9} \quad P_{0,10} \quad P_{0,11} \quad \cdots$$

$$\leqslant_{\mathrm{W}} \quad \leqslant_{\mathrm{W}} \quad \leqslant_{\mathrm{W}} \quad \leqslant_{\mathrm{W}} \quad \leqslant_{\mathrm{W}} \quad \leqslant_{\mathrm{W}} \quad \leqslant_{\mathrm{W}} \quad \leqslant_{\mathrm{W}} \quad \leqslant_{\mathrm{W}} \quad \leqslant_{\mathrm{W}} \quad \leqslant_{\mathrm{W}} \quad \leqslant_{\mathrm{W}} \quad \cdots$$

$$P_{1,0+1} \quad P_{1,1+1} \quad P_{1,2+1} \quad P_{1,3+1} \quad P_{1,4+1} \quad P_{1,5+1} \quad P_{1,6+1} \quad P_{1,7+1} \quad P_{1,8+1} \quad P_{1,9+1} \quad P_{1,10+1} \quad P_{1,11+1} \quad \cdots$$

Trivia: $P_{i,j} \leqslant_{\mathrm{W}} P_{i,j+1}$, $P_{i,j} \equiv_{\mathrm{W}} P_{i+2,j+2}$, $P_{i,j} \equiv_{\mathrm{W}} P^{\mathrm{c}}_{i+1,j+1}$

$$
\begin{array}{cccccccccccccc}
P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} & P_{0,4} & P_{0,5} & P_{0,6} & P_{0,7} & P_{0,8} & P_{0,9} & P_{0,10} & P_{0,11} & \cdots \\
\leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \cdots \\
P_{1,0+1} & P_{1,1+1} & P_{1,2+1} & P_{1,3+1} & P_{1,4+1} & P_{1,5+1} & P_{1,6+1} & P_{1,7+1} & P_{1,8+1} & P_{1,9+1} & P_{1,10+1} & P_{1,11+1} & \cdots
\end{array}
$$

Theorem $P_{i,j} \not\leqslant_{\mathrm{W}} P_{i+1,j+1}$

Trivia: $P_{i,j} \leqslant_{\mathrm{W}} P_{i,j+1}$, $P_{i,j} \equiv_{\mathrm{W}} P_{i+2,j+2}$, $P_{i,j} \equiv_{\mathrm{W}} P^{\mathrm{c}}_{i+1,j+1}$

$$\begin{array}{cccccccccccccc} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} & P_{0,4} & P_{0,5} & P_{0,6} & P_{0,7} & P_{0,8} & P_{0,9} & P_{0,10} & P_{0,11} & \cdots \\ \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \cdots \\ P_{1,0+1} & P_{1,1+1} & P_{1,2+1} & P_{1,3+1} & P_{1,4+1} & P_{1,5+1} & P_{1,6+1} & P_{1,7+1} & P_{1,8+1} & P_{1,9+1} & P_{1,10+1} & P_{1,11+1} & \cdots \end{array}$$

Theorem $P_{i,j} \not\leqslant_{\mathrm{W}} P_{i+1,j+1}$

Proof Assume that $P_{i,j} \leqslant_{\mathrm{W}} P_{i+1,j+1}$

Trivia: $P_{i,j} \leqslant_{\mathrm{W}} P_{i,j+1}$, $P_{i,j} \equiv_{\mathrm{W}} P_{i+2,j+2}$, $P_{i,j} \equiv_{\mathrm{W}} P^{\mathrm{c}}_{i+1,j+1}$

$$\begin{array}{cccccccccccc} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} & P_{0,4} & P_{0,5} & P_{0,6} & P_{0,7} & P_{0,8} & P_{0,9} & P_{0,10} & P_{0,11} \cdots \\ \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \cdots \\ P_{1,0+1} & P_{1,1+1} & P_{1,2+1} & P_{1,3+1} & P_{1,4+1} & P_{1,5+1} & P_{1,6+1} & P_{1,7+1} & P_{1,8+1} & P_{1,9+1} & P_{1,10+1} & P_{1,11+1} \cdots \end{array}$$

Theorem $P_{i,j} \not\leqslant_{\mathrm{W}} P_{i+1,j+1}$

Proof Assume that $P_{i,j} \leqslant_{\mathrm{W}} P_{i+1,j+1}$

$\rightsquigarrow \mathrm{L}(\mathcal{A}) = P_{i,j}$ with $\mathcal{A}$ **det.** $(i+1, j+1)$-parity automaton

Trivia: $P_{i,j} \leqslant_{\mathrm{W}} P_{i,j+1}$, $P_{i,j} \equiv_{\mathrm{W}} P_{i+2,j+2}$, $P_{i,j} \equiv_{\mathrm{W}} P^{\mathrm{c}}_{i+1,j+1}$

$$
\begin{array}{cccccccccccccc}
P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} & P_{0,4} & P_{0,5} & P_{0,6} & P_{0,7} & P_{0,8} & P_{0,9} & P_{0,10} & P_{0,11} & \cdots \\
\leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \cdots \\
P_{1,0+1} & P_{1,1+1} & P_{1,2+1} & P_{1,3+1} & P_{1,4+1} & P_{1,5+1} & P_{1,6+1} & P_{1,7+1} & P_{1,8+1} & P_{1,9+1} & P_{1,10+1} & P_{1,11+1} & \cdots
\end{array}
$$

Theorem $P_{i,j} \nleqslant_{\mathrm{W}} P_{i+1,j+1}$

Proof Assume that $P_{i,j} \leqslant_{\mathrm{W}} P_{i+1,j+1}$

$\rightsquigarrow \mathrm{L}(\mathcal{A}) = P_{i,j}$ with $\mathcal{A}$ **det.** $(i+1, j+1)$-parity automaton

$\alpha =$

$\rho = \quad q_0$

Trivia: $P_{i,j} \leqslant_{\mathrm{W}} P_{i,j+1}$, $P_{i,j} \equiv_{\mathrm{W}} P_{i+2,j+2}$, $P_{i,j} \equiv_{\mathrm{W}} P^{\mathrm{c}}_{i+1,j+1}$

$$\begin{array}{cccccccccccc} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} & P_{0,4} & P_{0,5} & P_{0,6} & P_{0,7} & P_{0,8} & P_{0,9} & P_{0,10} & \cdots \\ \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \cdots \\ P_{1,0+1} & P_{1,1+1} & P_{1,2+1} & P_{1,3+1} & P_{1,4+1} & P_{1,5+1} & P_{1,6+1} & P_{1,7+1} & P_{1,8+1} & P_{1,9+1} & P_{1,10+1} & \cdots \end{array}$$

Theorem $P_{i,j} \not\leqslant_{\mathrm{W}} P_{i+1,j+1}$

Proof Assume that $P_{i,j} \leqslant_{\mathrm{W}} P_{i+1,j+1}$

$\rightsquigarrow$ $\mathrm{L}(\mathcal{A}) = P_{i,j}$ with $\mathcal{A}$ **det.** $(i+1, j+1)$-parity automaton

$\alpha =$

$\rho = \quad q_0$

$\Omega: \quad \downarrow$

$\tau = \quad 1$

Trivia: $P_{i,j} \leqslant_W P_{i,j+1}$, $P_{i,j} \equiv_W P_{i+2,j+2}$, $P_{i,j} \equiv_W P^c_{i+1,j+1}$

$$P_{0,0} \quad P_{0,1} \quad P_{0,2} \quad P_{0,3} \quad P_{0,4} \quad P_{0,5} \quad P_{0,6} \quad P_{0,7} \quad P_{0,8} \quad P_{0,9} \quad P_{0,10} \quad P_{0,11}$$

$$\leqslant_W \quad \leqslant_W \quad \leqslant_W \quad \leqslant_W \quad \leqslant_W \quad \leqslant_W \quad \leqslant_W \quad \leqslant_W \quad \leqslant_W \quad \leqslant_W \quad \leqslant_W \quad \leqslant_W$$

$$P_{1,0+1} \quad P_{1,1+1} \quad P_{1,2+1} \quad P_{1,3+1} \quad P_{1,4+1} \quad P_{1,5+1} \quad P_{1,6+1} \quad P_{1,7+1} \quad P_{1,8+1} \quad P_{1,9+1} \quad P_{1,10+1} \quad P_{1,11+1}$$

Theorem $P_{i,j} \not\leqslant_W P_{i+1,j+1}$

Proof Assume that $P_{i,j} \leqslant_W P_{i+1,j+1}$

$\rightsquigarrow \mathrm{L}(\mathcal{A}) = P_{i,j}$ with $\mathcal{A}$ **det.** $(i+1, j+1)$-parity automaton

$\alpha = \quad 0$

$\rho = \quad q_0$

$\Omega: \quad \downarrow$

$\tau = \quad 1$

Trivia: $P_{i,j} \leqslant_W P_{i,j+1}$, $P_{i,j} \equiv_W P_{i+2,j+2}$, $P_{i,j} \equiv_W P^c_{i+1,j+1}$

$$\begin{array}{cccccccccccc} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} & P_{0,4} & P_{0,5} & P_{0,6} & P_{0,7} & P_{0,8} & P_{0,9} & P_{0,10} & P_{0,11} \; \cdots \\ \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W \; \cdots \\ P_{1,0+1} & P_{1,1+1} & P_{1,2+1} & P_{1,3+1} & P_{1,4+1} & P_{1,5+1} & P_{1,6+1} & P_{1,7+1} & P_{1,8+1} & P_{1,9+1} & P_{1,10+1} & P_{1,11+1} \; \cdots \end{array}$$

Theorem $P_{i,j} \not\leqslant_W P_{i+1,j+1}$

Proof Assume that $P_{i,j} \leqslant_W P_{i+1,j+1}$

$\rightsquigarrow \mathrm{L}(\mathcal{A}) = P_{i,j}$ with $\mathcal{A}$ **det.** $(i{+}1, j{+}1)$-parity automaton

$\alpha =$ 0

$\rho =$ $q_0 \to q_1$

$\Omega:$ $\downarrow$

$\tau =$ 1

Trivia: $P_{i,j} \leqslant_W P_{i,j+1}$, $P_{i,j} \equiv_W P_{i+2,j+2}$, $P_{i,j} \equiv_W P^c_{i+1,j+1}$

$$\begin{matrix} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} & P_{0,4} & P_{0,5} & P_{0,6} & P_{0,7} & P_{0,8} & P_{0,9} & P_{0,10} & P_{0,11} & \cdots \\ \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \cdots \\ P_{1,0+1} & P_{1,1+1} & P_{1,2+1} & P_{1,3+1} & P_{1,4+1} & P_{1,5+1} & P_{1,6+1} & P_{1,7+1} & P_{1,8+1} & P_{1,9+1} & P_{1,10+1} & P_{1,11+1} & \cdots \end{matrix}$$

Theorem $P_{i,j} \nleqslant_W P_{i+1,j+1}$

Proof Assume that $P_{i,j} \leqslant_W P_{i+1,j+1}$

$\rightsquigarrow \mathrm{L}(\mathcal{A}) = P_{i,j}$ with $\mathcal{A}$ **det.** $(i+1, j+1)$-parity automaton

$$\begin{matrix} \alpha = & & 0 & \\ \rho = & q_0 & \rightarrow & q_1 \\ \Omega: & \downarrow & & \downarrow \\ \tau = & 1 & & 3 \end{matrix}$$

Trivia: $P_{i,j} \leqslant_{\mathrm{W}} P_{i,j+1}$, $P_{i,j} \equiv_{\mathrm{W}} P_{i+2,j+2}$, $P_{i,j} \equiv_{\mathrm{W}} P^{\mathrm{c}}_{i+1,j+1}$

$$\begin{array}{cccccccccccc} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} & P_{0,4} & P_{0,5} & P_{0,6} & P_{0,7} & P_{0,8} & P_{0,9} & P_{0,10} & \cdots \\ \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \cdots \\ P_{1,0+1} & P_{1,1+1} & P_{1,2+1} & P_{1,3+1} & P_{1,4+1} & P_{1,5+1} & P_{1,6+1} & P_{1,7+1} & P_{1,8+1} & P_{1,9+1} & P_{1,10+1} & \cdots \end{array}$$

Theorem $P_{i,j} \nleqslant_{\mathrm{W}} P_{i+1,j+1}$

Proof Assume that $P_{i,j} \leqslant_{\mathrm{W}} P_{i+1,j+1}$

$\rightsquigarrow \mathrm{L}(\mathcal{A}) = P_{i,j}$ with $\mathcal{A}$ **det.** $(i+1, j+1)$-parity automaton

$$\begin{array}{rcc} \alpha = & 0 & 2 \\ \rho = & q_0 & \to q_1 \\ \Omega: & \downarrow & \downarrow \\ \tau = & 1 & 3 \end{array}$$

Trivia: $P_{i,j} \leqslant_{\mathrm{W}} P_{i,j+1}$, $P_{i,j} \equiv_{\mathrm{W}} P_{i+2,j+2}$, $P_{i,j} \equiv_{\mathrm{W}} P^{\mathrm{c}}_{i+1,j+1}$

$$\begin{array}{cccccccccccccc}
P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} & P_{0,4} & P_{0,5} & P_{0,6} & P_{0,7} & P_{0,8} & P_{0,9} & P_{0,10} & P_{0,11} & \cdots \\
\leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \cdots \\
P_{1,0+1} & P_{1,1+1} & P_{1,2+1} & P_{1,3+1} & P_{1,4+1} & P_{1,5+1} & P_{1,6+1} & P_{1,7+1} & P_{1,8+1} & P_{1,9+1} & P_{1,10+1} & P_{1,11+1} & \cdots
\end{array}$$

Theorem $P_{i,j} \not\leqslant_{\mathrm{W}} P_{i+1,j+1}$

Proof Assume that $P_{i,j} \leqslant_{\mathrm{W}} P_{i+1,j+1}$

$\rightsquigarrow \mathrm{L}(\mathcal{A}) = P_{i,j}$ with $\mathcal{A}$ **det.** $(i+1, j+1)$-parity automaton

$$\begin{array}{rccc}
\alpha = & 0 & 2 & \\
\rho = & q_0 & \rightarrow q_1 & \rightarrow q_2 \\
\Omega: & \downarrow & \downarrow & \\
\tau = & 1 & 3 &
\end{array}$$

Trivia: $P_{i,j} \leqslant_{\mathrm{W}} P_{i,j+1}$, $P_{i,j} \equiv_{\mathrm{W}} P_{i+2,j+2}$, $P_{i,j} \equiv_{\mathrm{W}} P^{\mathrm{c}}_{i+1,j+1}$

$$\begin{array}{cccccccccccc} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} & P_{0,4} & P_{0,5} & P_{0,6} & P_{0,7} & P_{0,8} & P_{0,9} & P_{0,10} & \dots \\ \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \dots \\ P_{1,0+1} & P_{1,1+1} & P_{1,2+1} & P_{1,3+1} & P_{1,4+1} & P_{1,5+1} & P_{1,6+1} & P_{1,7+1} & P_{1,8+1} & P_{1,9+1} & P_{1,10+1} & \dots \end{array}$$

Theorem $P_{i,j} \not\leqslant_{\mathrm{W}} P_{i+1,j+1}$

Proof Assume that $P_{i,j} \leqslant_{\mathrm{W}} P_{i+1,j+1}$

$\rightsquigarrow \mathrm{L}(\mathcal{A}) = P_{i,j}$ with $\mathcal{A}$ **det.** $(i+1, j+1)$-parity automaton

$$\begin{array}{rcccccccccc} \alpha = & & 0 & 2 & 4 & 1 & 0 & 5 & 1 & & \cdots \\ \rho = & q_0 & q_1 & q_2 & q_3 & q_4 & q_5 & q_6 & q_7 & & \cdots \\ \Omega: & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & & \\ \tau = & 1 & 3 & 5 & 2 & 1 & 6 & 2 & 3 & & \cdots \end{array}$$

Trivia: $P_{i,j} \leqslant_{\mathrm{W}} P_{i,j+1}$, $\quad P_{i,j} \equiv_{\mathrm{W}} P_{i+2,j+2}$, $\quad P_{i,j} \equiv_{\mathrm{W}} P^{\mathrm{c}}_{i+1,j+1}$

$$\begin{matrix} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} & P_{0,4} & P_{0,5} & P_{0,6} & P_{0,7} & P_{0,8} & P_{0,9} & P_{0,10} & P_{0,11} & \cdots \\ \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \leqslant_{\mathrm{W}} & \cdots \\ P_{1,0+1} & P_{1,1+1} & P_{1,2+1} & P_{1,3+1} & P_{1,4+1} & P_{1,5+1} & P_{1,6+1} & P_{1,7+1} & P_{1,8+1} & P_{1,9+1} & P_{1,10+1} & P_{1,11+1} & \cdots \end{matrix}$$

Theorem $P_{i,j} \nleqslant_{\mathrm{W}} P_{i+1,j+1}$

Proof Assume that $P_{i,j} \leqslant_{\mathrm{W}} P_{i+1,j+1}$

⇝ $\mathrm{L}(\mathcal{A}) = P_{i,j}$ with $\mathcal{A}$ **det.** $(i+1, j+1)$-parity automaton

$\alpha =$		0	2	4	1	0	5	1	$\cdots$
$\rho =$	q_0 →	q_1 →	q_2 →	q_3 →	q_4 →	q_5 →	q_6 →	q_7	$\cdots$
Ω:	↓	↓	↓	↓	↓	↓	↓	↓	
$\tau =$	1	3	5	2	1	6	2	3	$\cdots$

$$1 + \limsup_{n\to\infty} \alpha(n) = \limsup_{n\to\infty} \tau(n)$$

Trivia: $P_{i,j} \leqslant_W P_{i,j+1}$, $P_{i,j} \equiv_W P_{i+2,j+2}$, $P_{i,j} \equiv_W P^c_{i+1,j+1}$

$$\begin{array}{ccccccccccccc} P_{0,0} & P_{0,1} & P_{0,2} & P_{0,3} & P_{0,4} & P_{0,5} & P_{0,6} & P_{0,7} & P_{0,8} & P_{0,9} & P_{0,10} & P_{0,11} & \cdots \\ \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \leqslant_W & \cdots \\ P_{1,0+1} & P_{1,1+1} & P_{1,2+1} & P_{1,3+1} & P_{1,4+1} & P_{1,5+1} & P_{1,6+1} & P_{1,7+1} & P_{1,8+1} & P_{1,9+1} & P_{1,10+1} & P_{1,11+1} & \cdots \end{array}$$

Theorem $P_{i,j} \not\leqslant_W P_{i+1,j+1}$

Proof Assume that $P_{i,j} \leqslant_W P_{i+1,j+1}$

$\rightsquigarrow$ $L(\mathcal{A}) = P_{i,j}$ with $\mathcal{A}$ **det.** $(i+1, j+1)$-parity automaton

$$\begin{array}{lcccccccccc} \alpha = & & 0 & 2 & 4 & 1 & 0 & 5 & 1 & & \cdots \\ \rho = & q_0 & \to q_1 & \to q_2 & \to q_3 & \to q_4 & \to q_5 & \to q_6 & \to q_7 & & \cdots \\ \Omega: & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & & \\ \tau = & 1 & 3 & 5 & 2 & 1 & 6 & 2 & 3 & & \cdots \end{array}$$

$$1 + \limsup_{n\to\infty} \alpha(n) = \limsup_{n\to\infty} \tau(n)$$

$$\alpha \in L(\mathcal{A}) \text{ iff } \tau \in P_{i,j} \text{ iff } \alpha \notin P_{i,j} \qquad \blacksquare$$

Part 5

Effective characterisations

Task: **understand** which $L \in \mathbf{REG}$ are **simple**.

Task: **understand** which $L \in \mathbf{REG}$ are **simple**.

Procedure:

Input: $\mathcal{A}$

Output: Is $\mathrm{L}(\mathcal{A})$ simple?

Task: **understand** which $L \in \mathbf{REG}$ are **simple**.

definable in a weaker logic (e.g. FO)

Procedure:

Input: $\mathcal{A}$

Output: Is $\mathrm{L}(\mathcal{A})$ simple?

Task: **understand** which $L \in \mathbf{REG}$ are **simple**.

- definable in a weaker logic (e.g. FO)
- finite / countable / meagre / ...

Procedure:

Input: $\mathcal{A}$

Output: Is $\mathrm{L}(\mathcal{A})$ simple?

Task: **understand** which $L \in \mathbf{REG}$ are **simple**.

Procedure:

Input: $\mathcal{A}$

Output: Is $\mathrm{L}(\mathcal{A})$ simple?

- definable in a weaker logic (e.g. $\textsc{fo}$)
- finite / countable / meagre / ...
- topologically simple (e.g. **Borel**)

Task: **understand** which $L \in \mathbf{REG}$ are **simple**.

Procedure:

Input: $\mathcal{A}$

Output: Is $\mathrm{L}(\mathcal{A})$ simple?

- definable in a weaker logic (e.g. $\textsc{fo}$)
- finite / countable / meagre / ...
- topologically simple (e.g. **Borel**)
- ...

Task: **understand** which $L \in \mathbf{REG}$ are **simple**.

Procedure:

Input: $\mathcal{A}$

Output: Is $\mathrm{L}(\mathcal{A})$ simple?

- definable in a weaker logic (e.g. FO)
- finite / countable / meagre / ...
- topologically simple (e.g. **Borel**)
- ...

Theorem (Schutzenberger ['65]; McNaughton, Papert ['71]; Thomas ['79])

It is decidable if $L \in \mathbf{REG}$ is First-order (i.e. FO) definable.

Task: **understand** which $L \in \mathbf{REG}$ are **simple**.

Procedure:

Input: $\mathcal{A}$

Output: Is $\mathrm{L}(\mathcal{A})$ simple?

- definable in a weaker logic (e.g. FO)
- finite / countable / meagre / ...
- topologically simple (e.g. **Borel**)
- ...

Theorem (Schutzenberger ['65]; McNaughton, Papert ['71]; Thomas ['79])

It is decidable if $L \in \mathbf{REG}$ is First-order (i.e. FO) definable.

Theorem (Bojańczyk, Walukiewicz ['04])

It is decidable if a regular language of finite trees is EF definable.

Task: **understand** which $L \in \mathbf{REG}$ are **simple**.

Procedure:

Input: $\mathcal{A}$

Output: Is $\mathrm{L}(\mathcal{A})$ simple?

- definable in a weaker logic (e.g. FO)
- finite / countable / meagre / ...
- topologically simple (e.g. **Borel**)
- ...

Theorem (Schutzenberger ['65]; McNaughton, Papert ['71]; Thomas ['79])

It is decidable if $L \in \mathbf{REG}$ is First-order (i.e. FO) definable.

Theorem (Bojańczyk, Walukiewicz ['04])

It is decidable if a regular language of finite trees is EF definable.

Theorem (Murlak ['06])

Topological complexity is dec. for deterministic languages of inf. trees.

Task: **understand** which $L \in \mathbf{REG}$ are **simple**.

Procedure:

Input: $\mathcal{A}$

Output: Is $\mathrm{L}(\mathcal{A})$ simple?

- definable in a weaker logic (e.g. FO)
- finite / countable / meagre / ...
- topologically simple (e.g. **Borel**)
- ...

Theorem (Schutzenberger ['65]; McNaughton, Papert ['71]; Thomas ['79])

It is decidable if $L \in \mathbf{REG}$ is First-order (i.e. FO) definable.

Theorem (Bojańczyk, Walukiewicz ['04])

It is decidable if a regular language of finite trees is EF definable.

Theorem (Murlak ['06])

Topological complexity is dec. for deterministic languages of inf. trees.

[Bárány, Bojańczyk, Colcombet, Duparc, Facchini, Idziaszek, Kuperberg, Michalewski, Murlak, Niwiński, Place, Sreejith, Walukiewicz, ...]

Pattern method for **rigid representations**

Pattern method for rigid representations

1. Input $L = \mathrm{L}(\mathcal{A})$

Pattern method for **rigid representations**

1. Input $L = \mathrm{L}(\mathcal{A})$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_L)$

Pattern method for **rigid representations**

1. Input $L = \mathrm{L}(\mathcal{A})$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_L)$ [Properties of $\mathcal{A}_L$ are properties of L]

Pattern method for **rigid representations**

1. Input $L = \mathrm{L}(\mathcal{A})$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_L)$ $\left[\begin{array}{c}\text{Properties of } \mathcal{A}_L \\ \text{are properties of } L\end{array}\right]$

$$\varphi \equiv \big(\varphi \wedge \Psi\big) \vee \big(\varphi \wedge \neg\Psi\big)$$

Pattern method for **rigid representations**

1. Input $L = \mathrm{L}(\mathcal{A})$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_L)$ [Properties of $\mathcal{A}_L$ are properties of L]

Pattern method for **rigid representations**

1. Input $L = \mathrm{L}(\mathcal{A})$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_L)$ [Properties of $\mathcal{A}_L$ are properties of L]

3. Search in $\mathcal{A}_L$ for a complicated **pattern**

Pattern method for **rigid representations**

1. Input $L = \mathrm{L}(\mathcal{A})$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_L)$ [Properties of $\mathcal{A}_L$ are properties of L]

3. Search in $\mathcal{A}_L$ for a complicated **pattern**

[e.g. 

]

Pattern method for rigid representations

1. Input $L = \mathrm{L}(\mathcal{A})$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_L)$ [Properties of $\mathcal{A}_L$ are properties of L]

3. Search in $\mathcal{A}_L$ for a complicated **pattern**

[e.g. 

or $x^M \neq x^M \cdot x$]

Pattern method for rigid representations

1. Input $L = \mathrm{L}(\mathcal{A})$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_L)$ [Properties of $\mathcal{A}_L$ are properties of L]

3. Search in $\mathcal{A}_L$ for a complicated **pattern**

[e.g. 

or $x^M \neq x^M \cdot x$]

found

3.a Prove that L is **hard**

Pattern method for **rigid representations**

1. Input $L = \mathrm{L}(\mathcal{A})$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_L)$ [Properties of $\mathcal{A}_L$ are properties of L]

3. Search in $\mathcal{A}_L$ for a complicated **pattern**

[e.g. 

or $x^M \neq x^M \cdot x$]

3.a Prove that L is **hard**

3.b Use it to show that L is simple

Pattern method for rigid representations

1. Input $L = \mathrm{L}(\mathcal{A})$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_L)$ [Properties of $\mathcal{A}_L$ are properties of L]

3. Search in $\mathcal{A}_L$ for a complicated **pattern**

[e.g. 

or $x^M \neq x^M \cdot x$]

3.a Prove that L is **hard**

3.b Use it to show that L is simple

Limitations:

Pattern method for rigid representations

1. Input $L = \mathrm{L}(\mathcal{A})$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_L)$ [Properties of $\mathcal{A}_L$ are properties of L]

3. Search in $\mathcal{A}_L$ for a complicated **pattern**

[e.g. 

or $x^M \neq x^M \cdot x$]

3.a Prove that L is **hard**

3.b Use it to show that L is simple

Limitations:

- **3.a** uses complexity in $\mathcal{A}_L$ to prove complexity of L

Pattern method for **rigid representations**

1. Input $L = \mathrm{L}(\mathcal{A})$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_L)$ [Properties of $\mathcal{A}_L$ are properties of L]

3. Search in $\mathcal{A}_L$ for a complicated **pattern**

[e.g. 

or $x^M \neq x^M \cdot x$]

3.a Prove that L is **hard**

3.b Use it to show that L is simple

Limitations:

- **3.a** uses complexity in $\mathcal{A}_L$ to prove complexity of L

  ⇝ requires **rigid representations**

Pattern method for rigid representations

1. Input $L = \mathrm{L}(\mathcal{A})$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_L)$ [Properties of $\mathcal{A}_L$ are properties of L]

3. Search in $\mathcal{A}_L$ for a complicated **pattern**

[e.g. 

or $x^M \neq x^M \cdot x$]

3.a Prove that L is **hard** **3.b** Use it to show that L is simple

Limitations:

- **3.a** uses complexity in $\mathcal{A}_L$ to prove complexity of L
  $\rightsquigarrow$ requires **rigid representations** **No such** for infinite trees**!**

Pattern method for **rigid representations**

1. Input $L = \mathrm{L}(\mathcal{A})$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_L)$ [Properties of $\mathcal{A}_L$ are properties of L]

3. Search in $\mathcal{A}_L$ for a complicated **pattern**

[e.g. 

or $x^M \neq x^M \cdot x$]

3.a Prove that L is **hard**

3.b Use it to show that L is simple

Limitations:

- **3.a** uses complexity in $\mathcal{A}_L$ to prove complexity of L
  ⇝ requires **rigid representations** — **No such** for infinite trees**!**
- **3.b** works under the assumption of lack of obstruction

Pattern method for **rigid representations**

1. Input $L = \mathrm{L}(\mathcal{A})$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_L)$ [Properties of $\mathcal{A}_L$ are properties of L]

3. Search in $\mathcal{A}_L$ for a complicated **pattern**

[e.g. 

or $x^M \neq x^M \cdot x$]

3.a Prove that L is **hard**

3.b Use it to show that L is simple

Limitations:

- **3.a** uses complexity in $\mathcal{A}_L$ to prove complexity of L
  - ⇝ requires **rigid representations**

No such for infinite trees**!**

- **3.b** works under the assumption of lack of obstruction
  - ⇝ difficult proofs

Game method

Game method

1. Input $L = \mathrm{L}(\varphi)$

Game method

1. Input $L = \mathrm{L}(\varphi)$ **2.** Construct a game $\mathcal{G}_\varphi$

Game method

1. Input $L = \mathrm{L}(\varphi)$ **2.** Construct a game $\mathcal{G}_\varphi$ **3.** Solve $\mathcal{G}_\varphi$

Game method

1. Input $L = \mathrm{L}(\varphi)$ **2.** Construct a game $\mathcal{G}_\varphi$ **3.** Solve $\mathcal{G}_\varphi$

3.a Take his **w.s.** σ_{I}

Game method

1. Input $L = \mathrm{L}(\varphi)$ **2.** Construct a game $\mathcal{G}_\varphi$ **3.** Solve $\mathcal{G}_\varphi$

3.a Take his **w.s.** σ_{I}

Use σ_{I} to prove that L is hard

Game method

1. Input $L = \mathrm{L}(\varphi)$ **2.** Construct a game $\mathcal{G}_\varphi$ **3.** Solve $\mathcal{G}_\varphi$

3.a Take his **w.s.** σ_{I} **3.b** Take his **w.s.** σ_{II}

Use σ_{I} to prove that L is hard

Game method

1. Input $L = \mathrm{L}(\varphi)$ **2.** Construct a game $\mathcal{G}_\varphi$ **3.** Solve $\mathcal{G}_\varphi$

3.a Take his **w.s.** σ_{I}

Use σ_{I} to prove that L is hard

3.b Take his **w.s.** σ_{II}

Use σ_{II} to prove that L is simple

Game method

1. Input $L = \mathrm{L}(\varphi)$ **2.** Construct a game $\mathcal{G}_\varphi$ **3.** Solve $\mathcal{G}_\varphi$

3.a Take his **w.s.** σ_{I}

Use σ_{I} to prove that L is hard

3.b Take his **w.s.** σ_{II}

Use σ_{II} to prove that L is simple

$\rightsquigarrow$ In both cases we are on the positive side.

Game method

1. Input $L = \mathrm{L}(\varphi)$ **2.** Construct a game $\mathcal{G}_\varphi$ **3.** Solve $\mathcal{G}_\varphi$

3.a Take his **w.s.** σ_{I}

Use σ_{I} to prove that L is hard

3.b Take his **w.s.** σ_{II}

Use σ_{II} to prove that L is simple

⇝ In both cases we are on the positive side.

⇝ If $\mathcal{G}_\varphi$ is regular then σ_{I} and σ_{II} are **finite memory**.

Game method

1. Input $L = \mathrm{L}(\varphi)$ **2.** Construct a game $\mathcal{G}_\varphi$ **3.** Solve $\mathcal{G}_\varphi$

3.a Take his **w.s.** σ_{I}

Use σ_{I} to prove that L is hard

3.b Take his **w.s.** σ_{II}

Use σ_{II} to prove that L is simple

$\rightsquigarrow$ In both cases we are on the positive side.

$\rightsquigarrow$ If $\mathcal{G}_\varphi$ is regular then σ_{I} and σ_{II} are **finite memory**.

$\rightsquigarrow$ $\mathcal{G}_\varphi$ can work with a non-rigid representation φ

Game method

1. Input $L = \mathrm{L}(\varphi)$ **2.** Construct a game $\mathcal{G}_\varphi$ **3.** Solve $\mathcal{G}_\varphi$

3.a Take his **w.s.** σ_{I}

Use σ_{I} to prove that L is hard

3.b Take his **w.s.** σ_{II}

Use σ_{II} to prove that L is simple

⇝ In both cases we are on the positive side.

⇝ If $\mathcal{G}_\varphi$ is regular then σ_{I} and σ_{II} are **finite memory**.

⇝ $\mathcal{G}_\varphi$ can work with a non-rigid representation φ

(e.g. deal with non-determinism).

Game method

1. Input $L = \mathrm{L}(\varphi)$ **2.** Construct a game $\mathcal{G}_\varphi$ **3.** Solve $\mathcal{G}_\varphi$

3.a Take his **w.s.** σ_{I}

Use σ_{I} to prove that L is hard

3.b Take his **w.s.** σ_{II}

Use σ_{II} to prove that L is simple

⇝ In both cases we are on the positive side.

⇝ If $\mathcal{G}_\varphi$ is regular then σ_{I} and σ_{II} are **finite memory**.

⇝ $\mathcal{G}_\varphi$ can work with a non-rigid representation φ

(e.g. deal with non-determinism).

Examples

Game method

1. Input $L = \mathrm{L}(\varphi)$ **2.** Construct a game $\mathcal{G}_\varphi$ **3.** Solve $\mathcal{G}_\varphi$

3.a Take his **w.s.** σ_{I}

Use σ_{I} to prove that L is hard

3.b Take his **w.s.** σ_{II}

Use σ_{II} to prove that L is simple

$\rightsquigarrow$ In both cases we are on the positive side.

$\rightsquigarrow$ If $\mathcal{G}_\varphi$ is regular then σ_{I} and σ_{II} are **finite memory**.

$\rightsquigarrow$ $\mathcal{G}_\varphi$ can work with a non-rigid representation φ

(e.g. deal with non-determinism).

Examples

-(Kirsten ['05]; Colcombet ['09]; Toruńczyk ['11]; Bojańczyk ['15]): **star-height**

Game method

1. Input $L = \mathrm{L}(\varphi)$ **2.** Construct a game $\mathcal{G}_\varphi$ **3.** Solve $\mathcal{G}_\varphi$

3.a Take his **w.s.** σ_{I}

Use σ_{I} to prove that L is hard

3.b Take his **w.s.** σ_{II}

Use σ_{II} to prove that L is simple

$\rightsquigarrow$ In both cases we are on the positive side.

$\rightsquigarrow$ If $\mathcal{G}_\varphi$ is regular then σ_{I} and σ_{II} are **finite memory**.

$\rightsquigarrow$ $\mathcal{G}_\varphi$ can work with a non-rigid representation φ

(e.g. deal with non-determinism).

Examples

-(Kirsten ['05]; Colcombet ['09]; Toruńczyk ['11]; Bojańczyk ['15]): **star-height**

-(Colcombet, Löding ['08] + Kuperberg, Vanden Boom ['13]):

a variant of **Rabin-Mostowski** index problem

Part 4

Two examples

Theorem (Colcombet et al. ['13]; S., Walukiewicz ['14])

It is decidable if a Büchi language of infinite trees is WMSO definable.

Theorem (Colcombet et al. ['13]; S., Walukiewicz ['14])

It is decidable if a $\underbrace{\text{Büchi language of infinite trees}}_{\textbf{no}\text{ rigid representation}}$ is WMSO definable.

Theorem (Colcombet et al. ['13]; S., Walukiewicz ['14])

It is decidable if a $\underbrace{\text{Büchi language of infinite trees}}_{\textbf{no}\text{ rigid representation}}$ is $\underbrace{\text{WMSO}}_{\text{weaker logic}}$ definable.

Theorem (Colcombet et al. ['13]; S., Walukiewicz ['14])

It is decidable if a $\underbrace{\text{Büchi language of infinite trees}}_{\textbf{no}\text{ rigid representation}}$ is $\underbrace{\text{WMSO definable}}_{\text{weaker logic}}$.

Proof

Take $L = \mathrm{L}(\mathcal{B})$ and construct a game $\mathcal{G}_{\mathcal{B}}$.

Theorem (Colcombet et al. ['13]; S., Walukiewicz ['14])

It is decidable if a $\underbrace{\text{Büchi language of infinite trees}}_{\textbf{no}\text{ rigid representation}}$ is $\underbrace{\text{WMSO definable}}_{\text{weaker logic}}$.

Proof

Take $L = \mathrm{L}(\mathcal{B})$ and construct a game $\mathcal{G}_{\mathcal{B}}$. $\big[W \equiv A \vee \big(B \wedge C\big)\big]$

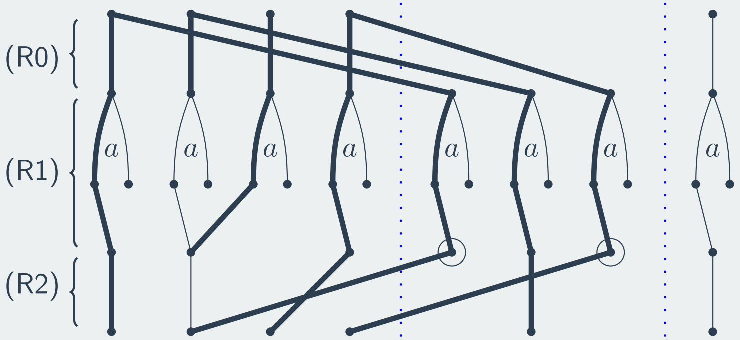
Theorem (Colcombet et al. ['13]; S., Walukiewicz ['14])

It is decidable if a Büchi language of infinite trees is WMSO definable.

no rigid representation — weaker logic

Proof

Take $L = \mathrm{L}(\mathcal{B})$ and construct a game $\mathcal{G}_{\mathcal{B}}$. $\big[W \equiv A \vee \big(B \wedge C\big)\big]$



Theorem (Colcombet et al. ['13]; S., Walukiewicz ['14])

It is decidable if a $\underbrace{\text{Büchi language of infinite trees}}_{\textbf{no}\text{ rigid representation}}$ is $\underbrace{\text{WMSO definable}}_{\text{weaker logic}}$.

Proof

Take $L = \mathrm{L}(\mathcal{B})$ and construct a game $\mathcal{G}_{\mathcal{B}}$. $\big[W \equiv A \vee \big(B \wedge C\big)\big]$

Theorem (Colcombet et al. ['13]; S., Walukiewicz ['14])

It is decidable if a $\underbrace{\text{Büchi language of infinite trees}}_{\textbf{no}\text{ rigid representation}}$ is $\underbrace{\text{WMSO definable}}_{\text{weaker logic}}$.

Proof

Take $L = \mathrm{L}(\mathcal{B})$ and construct a game $\mathcal{G}_{\mathcal{B}}$. $\quad \big[W \equiv A \vee \big(B \wedge C\big)\big]$

$\sigma_{\mathrm{I}} \rightsquigarrow L$ is **not** WMSO def.

Theorem (Colcombet et al. ['13]; S., Walukiewicz ['14])

It is decidable if a Büchi language of infinite trees is WMSO definable.

no rigid representation — weaker logic

Proof

Take $L = \mathrm{L}(\mathcal{B})$ and construct a game $\mathcal{G}_{\mathcal{B}}$. $\big[W \equiv A \vee \big(B \wedge C\big)\big]$

$\sigma_{\mathrm{I}} \rightsquigarrow L$ is **not** WMSO def. $\sigma_{\mathrm{II}} \rightsquigarrow$ WMSO formula for L

Theorem (Colcombet et al. ['13]; S., Walukiewicz ['14])

It is decidable if a Büchi language of infinite trees (**no** rigid representation) is WMSO definable (weaker logic).

Proof

Take $L = \mathrm{L}(\mathcal{B})$ and construct a game $\mathcal{G}_{\mathcal{B}}$. $\big[W \equiv A \vee \big(B \wedge C\big)\big]$

$\sigma_{\mathrm{I}} \rightsquigarrow L$ is **not** WMSO def. (pumping)

$\sigma_{\mathrm{II}} \rightsquigarrow$ WMSO formula for L

Theorem (Colcombet et al. ['13]; S., Walukiewicz ['14])

It is decidable if a Büchi language of infinite trees is WMSO definable.

(Büchi language of infinite trees: **no** rigid representation; WMSO: weaker logic)

Proof

Take $L = \mathrm{L}(\mathcal{B})$ and construct a game $\mathcal{G}_{\mathcal{B}}$. $\big[W \equiv A \vee \big(B \wedge C\big)\big]$

$\sigma_{\mathrm{I}} \rightsquigarrow L$ is **not** WMSO def. (pumping)

$\sigma_{\mathrm{II}} \rightsquigarrow$ WMSO formula for L ■

Theorem (Colcombet et al. ['13]; S., Walukiewicz ['14])

It is decidable if a $\underbrace{\text{Büchi language of infinite trees}}_{\textbf{no}\text{ rigid representation}}$ is $\underbrace{\text{WMSO definable}}_{\text{weaker logic}}$.

Proof

Take $L = \mathrm{L}(\mathcal{B})$ and construct a game $\mathcal{G}_{\mathcal{B}}$. $\quad \big[W \equiv A \vee \big(B \wedge C\big)\big]$

$\sigma_{\mathrm{I}} \rightsquigarrow L$ is **not** WMSO def. (pumping)

$\sigma_{\mathrm{II}} \rightsquigarrow$ WMSO formula for L ∎

But it seemed that we can get more (ordinal ranks)!

Theorem (Colcombet et al. ['13]; S., Walukiewicz ['14])

It is decidable if a $\underbrace{\text{Büchi language of infinite trees}}_{\textbf{no}\text{ rigid representation}}$ is $\underbrace{\text{WMSO definable}}_{\text{weaker logic}}$.

Proof

Take $L = \mathrm{L}(\mathcal{B})$ and construct a game $\mathcal{G}_{\mathcal{B}}$. $\big[W \equiv A \vee \big(B \wedge C\big)\big]$

$\sigma_{\mathrm{I}} \rightsquigarrow L$ is **not** WMSO def. (pumping)

$\sigma_{\mathrm{II}} \rightsquigarrow$ WMSO formula for L ■

Theorem (Colcombet et al. ['13]; S., Walukiewicz ['14])

It is decidable if a Büchi language of infinite trees is WMSO definable.

(Büchi language of infinite trees: **no** rigid representation; WMSO: weaker logic)

Proof

Take $L = \mathrm{L}(\mathcal{B})$ and construct a game $\mathcal{G}_{\mathcal{B}}$. $\quad \big[W \equiv A \vee \big(B \wedge C\big)\big]$

$\sigma_{\mathrm{I}} \rightsquigarrow L$ is **not** WMSO def. (pumping) $\qquad \sigma_{\mathrm{II}} \rightsquigarrow$ WMSO formula for L ■

Theorem (S., Walukiewicz ['16])

A Büchi language is WMSO def. **iff** it is **Borel**; and it is decidable.

Theorem (Colcombet et al. ['13]; S., Walukiewicz ['14])

It is decidable if a Büchi language of infinite trees is WMSO definable.

(Büchi language of infinite trees: **no** rigid representation; WMSO: weaker logic)

Proof

Take $L = \mathrm{L}(\mathcal{B})$ and construct a game $\mathcal{G}_{\mathcal{B}}$. $\quad \big[W \equiv A \vee \big(B \wedge C\big)\big]$

$\sigma_{\mathrm{I}} \rightsquigarrow L$ is **not** WMSO def. (pumping) $\qquad \sigma_{\mathrm{II}} \rightsquigarrow$ WMSO formula for L ■

Theorem (S., Walukiewicz ['16])

A Büchi language is WMSO def. **iff** it is **Borel**; and it is decidable.

Proof

Take $L = \mathrm{L}(\mathcal{B})$ and construct a game $\mathcal{G}'_{\mathcal{B}}$.

Theorem (Colcombet et al. ['13]; S., Walukiewicz ['14])

It is decidable if a $\underbrace{\text{Büchi language of infinite trees}}_{\textbf{no}\text{ rigid representation}}$ is $\underbrace{\text{WMSO definable}}_{\text{weaker logic}}$.

Proof

Take $L = \mathrm{L}(\mathcal{B})$ and construct a game $\mathcal{G}_{\mathcal{B}}$. $\qquad \big[W \equiv A \vee \big(B \wedge C\big)\big]$

$\sigma_{\mathrm{I}} \rightsquigarrow L$ is **not** WMSO def. (pumping) $\qquad \sigma_{\mathrm{II}} \rightsquigarrow$ WMSO formula for L ■

Theorem (S., Walukiewicz ['16])

A Büchi language is WMSO def. **iff** it is **Borel**; and it is decidable.

Proof

Take $L = \mathrm{L}(\mathcal{B})$ and construct a game $\mathcal{G}'_{\mathcal{B}}$. $\qquad \big[W \equiv \big(A \vee B\big) \wedge C'\big]$

Theorem (Colcombet et al. ['13]; S., Walukiewicz ['14])

It is decidable if a $\underbrace{\text{Büchi language of infinite trees}}_{\textbf{no}\text{ rigid representation}}$ is $\underbrace{\text{WMSO definable}}_{\text{weaker logic}}$.

Proof

Take $L = \mathrm{L}(\mathcal{B})$ and construct a game $\mathcal{G}_{\mathcal{B}}$. $\qquad \big[W \equiv A \vee \big(B \wedge C\big)\big]$

$\sigma_{\mathrm{I}} \rightsquigarrow L$ is **not** WMSO def. (pumping) $\qquad \sigma_{\mathrm{II}} \rightsquigarrow$ WMSO formula for L ■

Theorem (S., Walukiewicz ['16])

A Büchi language is WMSO def. **iff** it is **Borel**; and it is decidable.

Proof

Take $L = \mathrm{L}(\mathcal{B})$ and construct a game $\mathcal{G}'_{\mathcal{B}}$. $\qquad \big[W \equiv \big(A \vee B\big) \wedge C'\big]$

$\sigma_{\mathrm{I}} \rightsquigarrow \boldsymbol{\Sigma}^1_1 \leqslant_{\mathrm{W}} L$

$\rightsquigarrow L$ is **not** WMSO def.

Theorem (Colcombet et al. ['13]; S., Walukiewicz ['14])

It is decidable if a Büchi language of infinite trees is WMSO definable.

(Büchi language of infinite trees: **no** rigid representation; WMSO: weaker logic)

Proof

Take $L = \mathrm{L}(\mathcal{B})$ and construct a game $\mathcal{G}_{\mathcal{B}}$. $\quad [W \equiv A \vee (B \wedge C)]$

(I) wins — (II) wins

$\sigma_{\mathrm{I}} \rightsquigarrow L$ is **not** WMSO def. (pumping)

$\sigma_{\mathrm{II}} \rightsquigarrow$ WMSO formula for L ∎

Theorem (S., Walukiewicz ['16])

A Büchi language is WMSO def. **iff** it is **Borel**; and it is decidable.

Proof

Take $L = \mathrm{L}(\mathcal{B})$ and construct a game $\mathcal{G}'_{\mathcal{B}}$. $\quad [W \equiv (A \vee B) \wedge C']$

(I) wins — (II) wins

$\sigma_{\mathrm{I}} \rightsquigarrow \mathbf{\Sigma}^1_1 \leqslant_{\mathrm{W}} L$

$\rightsquigarrow L$ is **not** WMSO def.

$\sigma_{\mathrm{II}} \rightsquigarrow$ WMSO formula for L

$\rightsquigarrow L$ is **Borel**

Theorem (Colcombet et al. ['13]; S., Walukiewicz ['14])

It is decidable if a Büchi language of infinite trees is WMSO definable.

(Büchi language of infinite trees: **no** rigid representation; WMSO: weaker logic)

Proof

Take $L = \mathrm{L}(\mathcal{B})$ and construct a game $\mathcal{G}_{\mathcal{B}}$. $\quad [W \equiv A \vee (B \wedge C)]$

(I) **wins** / (II) **wins**

$\sigma_{\mathrm{I}} \rightsquigarrow L$ is **not** WMSO def. (pumping)

$\sigma_{\mathrm{II}} \rightsquigarrow$ WMSO formula for L ∎

Theorem (S., Walukiewicz ['16])

A Büchi language is WMSO def. **iff** it is **Borel**; and it is decidable.

Proof

Take $L = \mathrm{L}(\mathcal{B})$ and construct a game $\mathcal{G}'_{\mathcal{B}}$. $\quad [W \equiv (A \vee B) \wedge C']$

(I) **wins** / (II) **wins**

$\sigma_{\mathrm{I}} \rightsquigarrow \mathbf{\Sigma}^1_1 \leqslant_{\mathrm{W}} L$
$\rightsquigarrow L$ is **not** WMSO def.

$\sigma_{\mathrm{II}} \rightsquigarrow$ WMSO formula for L
$\rightsquigarrow L$ is **Borel** ∎

Theorem (Cavallari, Michalewski, S. [2017])

Let L be regular lang. of inf. trees. Then effectively either:

Theorem (Cavallari, Michalewski, S. [2017])

Let L be regular lang. of inf. trees. Then effectively either:

1. L is $\mathrm{weak{-}alt}(0,2)$-definable and $L \leqslant_{\mathrm{W}} \mathbf{\Pi}^0_2$

Theorem (Cavallari, Michalewski, S. [2017])

Let L be regular lang. of inf. trees. Then effectively either:

1. L is $\mathrm{weak{-}alt}(0,2)$-definable and $L \leqslant_{\mathrm{W}} \mathbf{\Pi}^0_2$

2. L isn't $\mathrm{weak{-}alt}(0,2)$-definable and $L \geqslant_{\mathrm{W}} \mathbf{\Sigma}^0_2$

Theorem (Cavallari, Michalewski, S. [2017])

Let L be regular lang. of inf. trees. Then effectively either:

1. L is $\mathrm{weak{-}alt}(0,2)$-definable and $L \leqslant_{\mathrm{W}} \mathbf{\Pi}^0_2$
2. L isn't $\underbrace{\mathrm{weak{-}alt}(0,2)}_{\text{weak index}}$-definable and $L \geqslant_{\mathrm{W}} \mathbf{\Sigma}^0_2$

Theorem (Cavallari, Michalewski, S. [2017])

Let L be regular lang. of inf. trees. Then effectively either:

1. L is $\mathrm{weak{-}alt}(0,2)$-definable and $L \leqslant_{\mathrm{W}} \boldsymbol{\Pi}^0_2$

2. L isn't $\underbrace{\mathrm{weak{-}alt}(0,2)}_{\text{weak index}}$-definable and $\underbrace{L \geqslant_{\mathrm{W}} \boldsymbol{\Sigma}^0_2}_{\text{topological complexity}}$

Theorem (Cavallari, Michalewski, S. [2017])

Let L be regular lang. of inf. trees. Then effectively either:

1. L is $\mathrm{weak{-}alt}(0,2)$-definable and $L \leqslant_{\mathrm{W}} \mathbf{\Pi}^0_2$

2. L isn't $\underbrace{\mathrm{weak{-}alt}(0,2)}_{\text{weak index}}$-definable and $\underbrace{L \geqslant_{\mathrm{W}} \mathbf{\Sigma}^0_2}_{\text{topological complexity}}$

Proof

Take two non-det. parity tree automata: $\mathcal{A}$ for L and $\mathcal{B}$ for L^{c}.

Theorem (Cavallari, Michalewski, S. [2017])

Let L be regular lang. of inf. trees. Then effectively either:

1. L is $\mathrm{weak{-}alt}(0,2)$-definable and $L \leqslant_{\mathrm{W}} \mathbf{\Pi}^0_2$
2. L isn't $\underbrace{\mathrm{weak{-}alt}(0,2)}_{\text{weak index}}$-definable and $\underbrace{L \geqslant_{\mathrm{W}} \mathbf{\Sigma}^0_2}_{\text{topological complexity}}$

Proof

Take two non-det. parity tree automata: $\mathcal{A}$ for L and $\mathcal{B}$ for L^{c}.

Consider a game $\mathcal{F}$ on $\mathcal{B} \times \mathcal{A} \times \mathcal{A}$

Theorem (Cavallari, Michalewski, S. [2017])

Let L be regular lang. of inf. trees. Then effectively either:

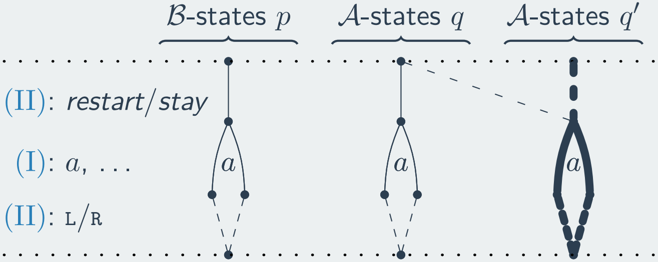
1. L is $\mathrm{weak{-}alt}(0,2)$-definable and $L \leqslant_{\mathrm{W}} \mathbf{\Pi}^0_2$

2. L isn't $\underbrace{\mathrm{weak{-}alt}(0,2)}_{\text{weak index}}$-definable and $\underbrace{L \geqslant_{\mathrm{W}} \mathbf{\Sigma}^0_2}_{\text{topological complexity}}$

Proof

Take two non-det. parity tree automata: $\mathcal{A}$ for L and $\mathcal{B}$ for L^{c}.

Consider a game $\mathcal{F}$ on $\mathcal{B} \times \mathcal{A} \times \mathcal{A}$

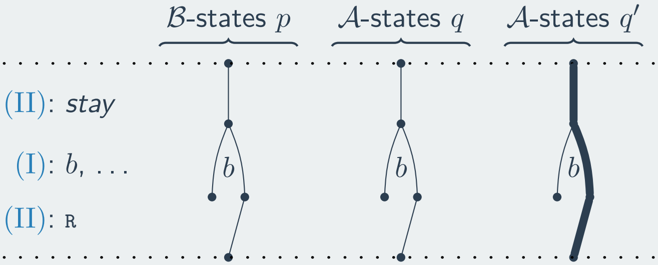


Gameplay

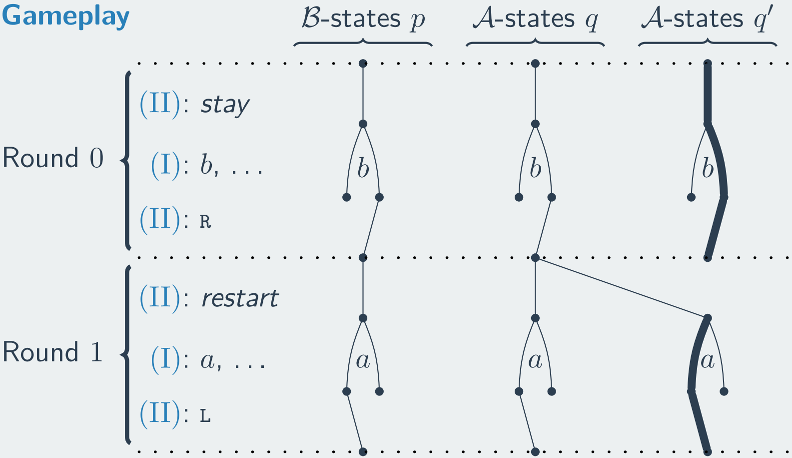
Gameplay

Gameplay

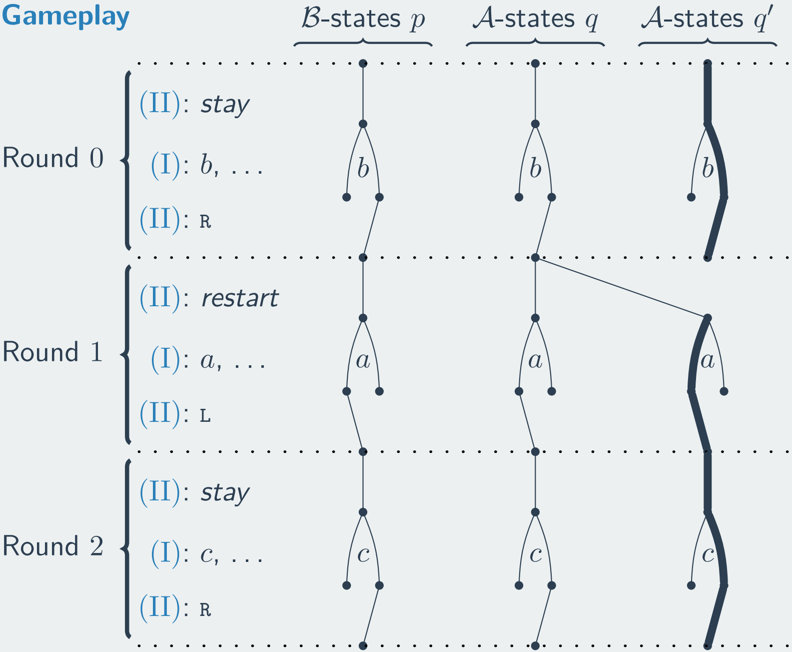
Round 0
- (II): *stay*
- (I): b, ...
- (II): R



Gameplay


$\mathcal{B}$-states p
$\mathcal{A}$-states q
$\mathcal{A}$-states q'
Round 0
(II): *stay*
(I): b, . . .
(II): R
Round 1
(II): *restart*
(I): a, . . .
(II): L
b
b
b
a
a
a

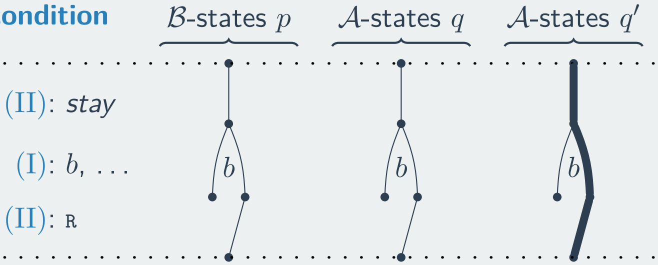
Gameplay



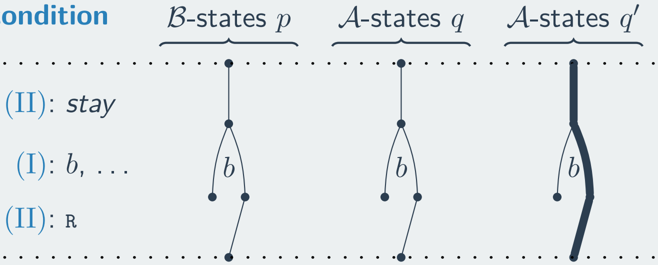
$\mathcal{B}$-states p
$\mathcal{A}$-states q
$\mathcal{A}$-states q'
Round 0
(II): *stay*
(I): b, . . .
(II): R
Round 1
(II): *restart*
(I): a, . . .
(II): L
Round 2
(II): *stay*
(I): c, . . .
(II): R
b
b
b
a
a
a
c
c
c

Winning condition

Winning condition

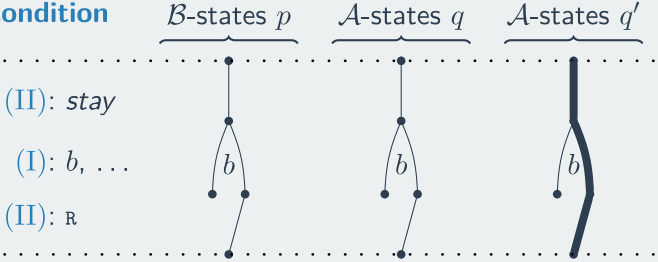


Winning condition



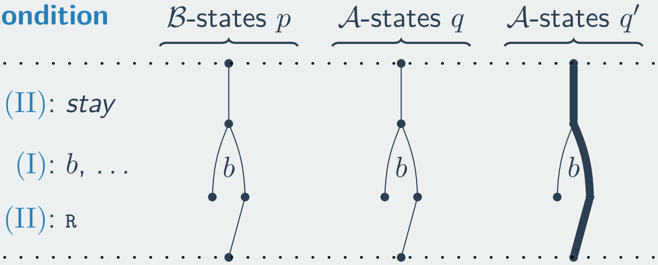
(WR) (II) restarted infinitely many times

Winning condition



(WR) (II) restarted infinitely many times $\rightsquigarrow$ $\mathbf{\Pi}^0_2$-complete set

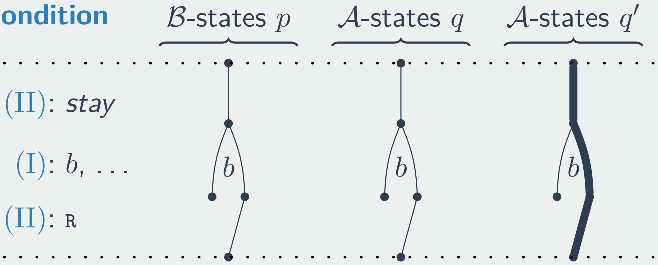
Winning condition



(WR) (II) restarted infinitely many times $\rightsquigarrow$ $\mathbf{\Pi}^0_2$-complete set

(WB) $\mathcal{B}$-states p are accepting

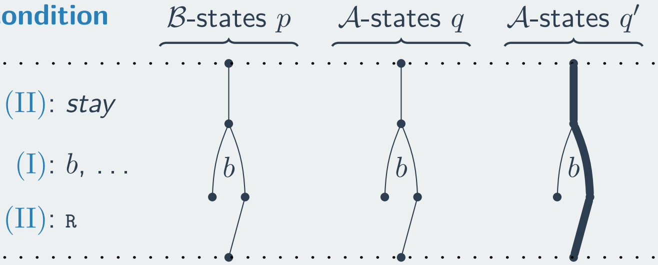
Winning condition



(WR) (II) restarted infinitely many times $\rightsquigarrow$ $\mathbf{\Pi}^0_2$-complete set

(WB) $\mathcal{B}$-states p are accepting $\rightsquigarrow$ seems that $t \notin L$

Winning condition

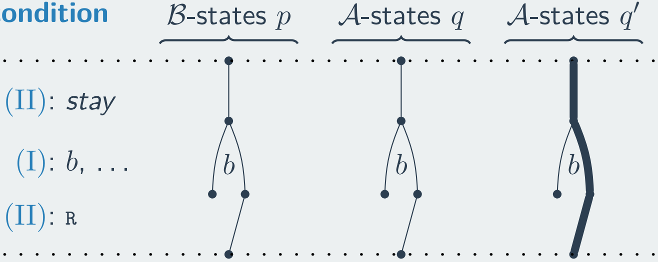


(WR) (II) restarted infinitely many times ⇝ $\mathbf{\Pi}^0_2$-complete set

(WB) $\mathcal{B}$-states p are accepting ⇝ seems that $t \notin L$

(WA) $\mathcal{A}$-states q' are accepting

Winning condition

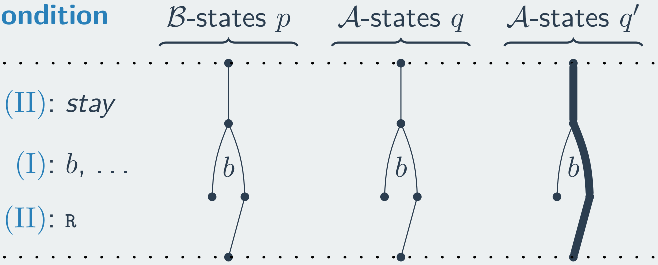


(WR) (II) restarted infinitely many times ⇝ $\mathbf{\Pi}^0_2$-complete set

(WB) $\mathcal{B}$-states p are accepting ⇝ seems that $t \notin L$

(WA) $\mathcal{A}$-states q' are accepting ⇝ seems that $t \in L$

Winning condition



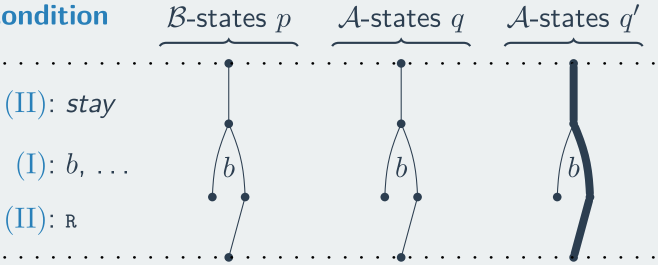
(WR) (II) restarted infinitely many times ⇝ $\mathbf{\Pi}^0_2$-complete set

(WB) $\mathcal{B}$-states p are accepting ⇝ seems that $t \notin L$

(WA) $\mathcal{A}$-states q' are accepting ⇝ seems that $t \in L$

$$W \equiv \big((\mathrm{WR}) \wedge (\mathrm{WB})\big) \vee \big(\neg(\mathrm{WR}) \wedge (\mathrm{WA})\big)$$

Winning condition



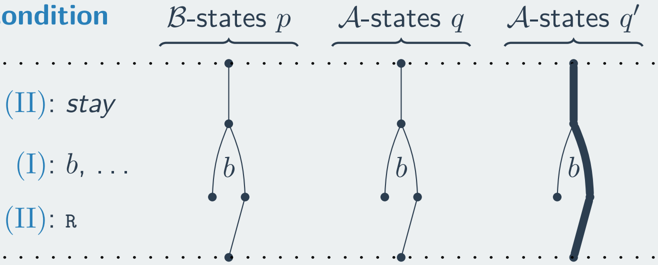
(WR) (II) restarted infinitely many times $\rightsquigarrow$ $\mathbf{\Pi}^0_2$-complete set

(WB) $\mathcal{B}$-states p are accepting $\rightsquigarrow$ seems that $t \notin L$

(WA) $\mathcal{A}$-states q' are accepting $\rightsquigarrow$ seems that $t \in L$

$$W \equiv \underbrace{\big((\mathrm{WR}) \wedge (\mathrm{WB})\big) \vee \big(\neg(\mathrm{WR}) \wedge (\mathrm{WA})\big)}_{\text{Wadge-like condition for } (\mathrm{WR}) \leqslant_{\mathrm{W}} L^{\mathrm{c}}}$$

Winning condition



(WR) (II) restarted infinitely many times ⇝ $\mathbf{\Pi}^0_2$-complete set

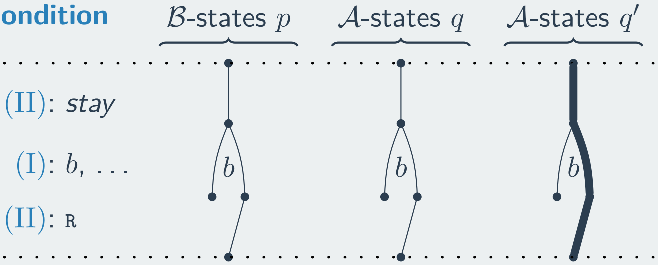
(WB) $\mathcal{B}$-states p are accepting ⇝ seems that $t \notin L$

(WA) $\mathcal{A}$-states q' are accepting ⇝ seems that $t \in L$

$$W \equiv \underbrace{\big((\mathrm{WR}) \wedge (\mathrm{WB})\big) \vee \big(\neg(\mathrm{WR}) \wedge (\mathrm{WA})\big)}_{\text{Wadge-like condition for } (\mathrm{WR}) \leqslant_{\mathrm{W}} L^{\mathrm{c}}}$$

⇝ regular condition over infinite words

Winning condition



(WR) (II) restarted infinitely many times ⇝ $\mathbf{\Pi}^0_2$-complete set

(WB) $\mathcal{B}$-states p are accepting ⇝ seems that $t \notin L$

(WA) $\mathcal{A}$-states q' are accepting ⇝ seems that $t \in L$

$$W \equiv \underbrace{\big((\mathrm{WR}) \land (\mathrm{WB})\big) \lor \big(\neg(\mathrm{WR}) \land (\mathrm{WA})\big)}$$

Wadge-like condition

for $(\mathrm{WR}) \leqslant_{\mathrm{W}} L^{\mathrm{c}}$

⇝ regular condition over infinite words ⇝ we can solve $\mathcal{F}$

Two lemmata:

Two lemmata:

1. If (I) wins $\mathcal{F}$ then L is **not** $\mathbf{\Pi}^0_2$

Two lemmata:

1. If (I) wins $\mathcal{F}$ then L is **not** $\mathbf{\Pi}^0_2$

Proof

Take a strategy of (I) in $\mathcal{F}$

Two lemmata:

1. If (I) wins $\mathcal{F}$ then L is **not** $\mathbf{\Pi}^0_2$

Proof

Take a strategy of (I) in $\mathcal{F}$

Confront it with multiple strategies of (II)

Two lemmata:

1. If (I) wins $\mathcal{F}$ then L is **not** $\mathbf{\Pi}^0_2$

Proof

Take a strategy of (I) in $\mathcal{F}$

Confront it with multiple strategies of (II)

$\rightsquigarrow$ a reduction proving that $(\mathrm{WR}) \leqslant_{\mathrm{W}} L^{\mathrm{c}}$

Two lemmata:

1. If (I) wins $\mathcal{F}$ then L is **not** $\mathbf{\Pi}^0_2$

Proof

Take a strategy of (I) in $\mathcal{F}$

Confront it with multiple strategies of (II)

$\rightsquigarrow$ a reduction proving that $(\mathrm{WR}) \leqslant_{\mathrm{W}} L^{\mathrm{c}}$

$\rightsquigarrow$ L is **not** $\mathrm{weak{-}alt}(0, 2)$-definable

Two lemmata:

1. If (I) wins $\mathcal{F}$ then L is **not** $\mathbf{\Pi}^0_2$

Proof

Take a strategy of (I) in $\mathcal{F}$

Confront it with multiple strategies of (II)

$\rightsquigarrow$ a reduction proving that $(\mathrm{WR}) \leqslant_{\mathrm{W}} L^{\mathrm{c}}$

$\rightsquigarrow$ L is **not** $\mathrm{weak{-}alt}(0,2)$-definable

1. If (II) wins $\mathcal{F}$ then L is $\mathrm{weak{-}alt}(0,2)$-definable

Two lemmata:

1. If (I) wins $\mathcal{F}$ then L is **not** $\mathbf{\Pi}^0_2$

Proof

Take a strategy of (I) in $\mathcal{F}$

Confront it with multiple strategies of (II)

$\rightsquigarrow$ a reduction proving that $(\mathrm{WR}) \leqslant_{\mathrm{W}} L^{\mathrm{c}}$

$\rightsquigarrow$ L is **not** $\mathrm{weak{-}alt}(0, 2)$-definable

1. If (II) wins $\mathcal{F}$ then L is $\mathrm{weak{-}alt}(0, 2)$-definable

Proof

Take a finite memory strategy of (II) in $\mathcal{F}$

Two lemmata:

1. If (I) wins $\mathcal{F}$ then L is **not** $\mathbf{\Pi}^0_2$

Proof

Take a strategy of (I) in $\mathcal{F}$

Confront it with multiple strategies of (II)

$\rightsquigarrow$ a reduction proving that $(\mathrm{WR}) \leqslant_{\mathrm{W}} L^{\mathrm{c}}$

$\rightsquigarrow$ L is **not** $\mathrm{weak}{-}\mathrm{alt}(0,2)$-definable

1. If (II) wins $\mathcal{F}$ then L is $\mathrm{weak}{-}\mathrm{alt}(0,2)$-definable

Proof

Take a finite memory strategy of (II) in $\mathcal{F}$

Add some pumping

Two lemmata:

1. If (I) wins $\mathcal{F}$ then L is **not** $\mathbf{\Pi}^0_2$

Proof

Take a strategy of (I) in $\mathcal{F}$

Confront it with multiple strategies of (II)

$\rightsquigarrow$ a reduction proving that $(\mathrm{WR}) \leqslant_{\mathrm{W}} L^{\mathrm{c}}$

$\rightsquigarrow$ L is **not** $\mathrm{weak{-}alt}(0,2)$-definable

1. If (II) wins $\mathcal{F}$ then L is $\mathrm{weak{-}alt}(0,2)$-definable

Proof

Take a finite memory strategy of (II) in $\mathcal{F}$

Add some pumping

$\rightsquigarrow$ a weak-alternating $(0,2)$ automaton for L

Two lemmata:

1. If (I) wins $\mathcal{F}$ then L is **not** $\boldsymbol{\Pi}^0_2$

Proof

Take a strategy of (I) in $\mathcal{F}$

Confront it with multiple strategies of (II)

$\rightsquigarrow$ a reduction proving that $(\mathrm{WR}) \leqslant_{\mathrm{W}} L^{\mathrm{c}}$

$\rightsquigarrow$ L is **not** $\mathrm{weak{-}alt}(0,2)$-definable

1. If (II) wins $\mathcal{F}$ then L is $\mathrm{weak{-}alt}(0,2)$-definable

Proof

Take a finite memory strategy of (II) in $\mathcal{F}$

Add some pumping

$\rightsquigarrow$ a weak-alternating $(0,2)$ automaton for L

$\rightsquigarrow$ $L \leqslant_{\mathrm{W}} \boldsymbol{\Pi}^0_2$

Two lemmata:

1. If (I) wins $\mathcal{F}$ then L is **not** $\mathbf{\Pi}^0_2$

Proof

Take a strategy of (I) in $\mathcal{F}$

Confront it with multiple strategies of (II)

$\rightsquigarrow$ a reduction proving that $(\mathrm{WR}) \leqslant_{\mathrm{W}} L^{\mathrm{c}}$

$\rightsquigarrow$ L is **not** $\mathrm{weak{-}alt}(0,2)$-definable

1. If (II) wins $\mathcal{F}$ then L is $\mathrm{weak{-}alt}(0,2)$-definable

Proof

Take a finite memory strategy of (II) in $\mathcal{F}$

Add some pumping

$\rightsquigarrow$ a weak-alternating $(0,2)$ automaton for L

$\rightsquigarrow$ $L \leqslant_{\mathrm{W}} \mathbf{\Pi}^0_2$

} A complete proof

Two lemmata:

1. If (I) wins $\mathcal{F}$ then L is **not** $\mathbf{\Pi}^0_2$

Proof

Take a strategy of (I) in $\mathcal{F}$

Confront it with multiple strategies of (II)

$\rightsquigarrow$ a reduction proving that $(\mathrm{WR}) \leqslant_{\mathrm{W}} L^{\mathrm{c}}$

$\rightsquigarrow$ L is **not** $\mathrm{weak{-}alt}(0,2)$-definable

1. If (II) wins $\mathcal{F}$ then L is $\mathrm{weak{-}alt}(0,2)$-definable

Proof

Take a finite memory strategy of (II) in $\mathcal{F}$

Add some pumping

$\rightsquigarrow$ a weak-alternating $(0,2)$ automaton for L

$\rightsquigarrow$ $L \leqslant_{\mathrm{W}} \mathbf{\Pi}^0_2$

} A complete proof **not** using properties on which the game $\mathcal{F}$ is based

Two lemmata:

1. If (I) wins $\mathcal{F}$ then L is **not** $\mathbf{\Pi}^0_2$

Proof

Take a strategy of (I) in $\mathcal{F}$

Confront it with multiple strategies of (II)

$\leadsto$ a reduction proving that $(\mathrm{WR}) \leqslant_{\mathrm{W}} L^{\mathrm{c}}$

$\leadsto$ L is **not** $\mathrm{weak{-}alt}(0,2)$-definable

1. If (II) wins $\mathcal{F}$ then L is $\mathrm{weak{-}alt}(0,2)$-definable

Proof

Take a finite memory strategy of (II) in $\mathcal{F}$

Add some pumping

$\leadsto$ a weak-alternating $(0,2)$ automaton for L

$\leadsto$ $L \leqslant_{\mathrm{W}} \mathbf{\Pi}^0_2$

A complete proof **not** using properties on which the game $\mathcal{F}$ is based

[dealternation]

Summary

Summary

→ characterising which languages are simple

Summary

→ characterising which languages are simple

→ pattern method (rigid representatons: det. aut. / algebra)

Summary

→ characterising which languages are simple

→ pattern method (rigid representatons: det. aut. / algebra)

↙

pattern found

⇝ L is hard

Summary

→ characterising which languages are simple

→ pattern method (rigid representatons: det. aut. / algebra)

pattern found

⇝ L is hard

pattern missing

⇝ L is simple

Summary

→ characterising which languages are simple

→ pattern method (rigid representatons: det. aut. / algebra)

pattern found ⇝ L is hard

pattern missing ⇝ L is simple

→ games (may deal with non-determinism)

Summary

→ characterising which languages are simple

→ pattern method (rigid representatons: det. aut. / algebra)

pattern found

$\rightsquigarrow$ L is hard

pattern missing

$\rightsquigarrow$ L is simple

→ games (may deal with non-determinism)

strategy of (I)

$\rightsquigarrow$ L is hard

Summary

→ characterising which languages are simple

→ pattern method (rigid representatons: det. aut. / algebra)

pattern found

⇝ L is hard

pattern missing

⇝ L is simple

→ games (may deal with non-determinism)

strategy of (I)

⇝ L is hard

strategy of (II)

⇝ L is simple

Summary

→ characterising which languages are simple

→ pattern method (rigid representatons: det. aut. / algebra)

- pattern found ⇝ L is hard
- pattern missing ⇝ L is simple

→ games (may deal with non-determinism)

- strategy of (I) ⇝ L is hard
- strategy of (II) ⇝ L is simple

→ **no** general recipe for design

Summary

⟶ characterising which languages are simple

⟶ pattern method (rigid representatons: det. aut. / algebra)

pattern found
⇝ L is hard

pattern missing
⇝ L is simple

⟶ games (may deal with non-determinism)

strategy of (I)
⇝ L is hard

strategy of (II)
⇝ L is simple

⟶ **no** general recipe for design

Conjecture: Every class of languages has a game characterisation