Games in topology and their effective variants

Michał Skrzypczak

Colloquium Of MIM, 07.12.2017

FNP

UNIVERSITY
OF WARSAW

Part 1

Generic objects

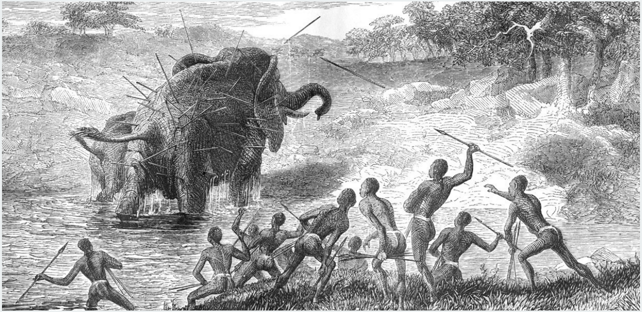
How to prove that **there exists a four-legged elephant**?

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

Option 1.: Find one.

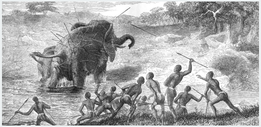
How to prove that **there exists a four-legged elephant**?

Option 1.: Find one.



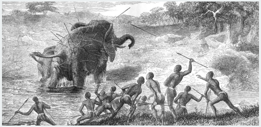
How to prove that **there exists a four-legged elephant**?

Option 1.: Find one.



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

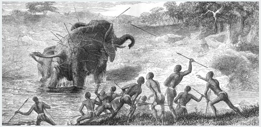
Option 1.: Find one.



Option 2.: Prove that a **generic** elephant has the property P.

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

Option 1.: Find one.

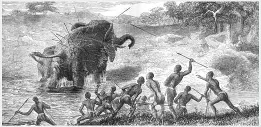


Option 2.: Prove that a **generic** elephant has the property P.

$$\mathbb{P}(P) > 1 - \epsilon$$

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

Option 1.: Find one.

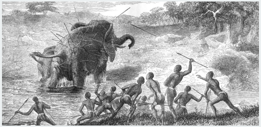


Option 2.: Prove that a **generic** elephant has the property P.

$$\mathbb{P}(P) > 1 - \epsilon$$

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

Option 1.: Find one.

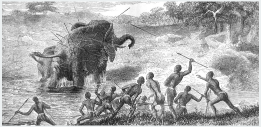


Option 2.: Prove that a **generic** elephant has the property P.

$$\mathbb{P}(P) > 1 - \epsilon$$

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

Option 1.: Find one.



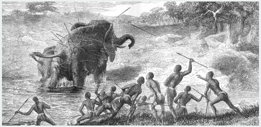
Option 2.: Prove that a **generic** elephant has the property P.

$$\mathbb{P}(P) > 1 - \epsilon$$

Option 3.: Go contrapositive, etc. . .

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

Option 1.: Find one.

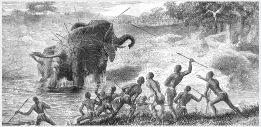


Option 2.: Prove that a **generic** elephant has the property P.

$$\underline{\mathbb{P}(P) > 1 - \epsilon}$$

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

Option 1.: Find one.



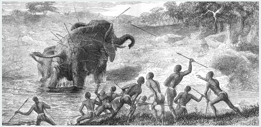
Option 2.: Prove that a **generic** elephant has the property P.

$$\underline{\mathbb{P}(P) > 1 - \epsilon}$$

$\rightsquigarrow$ strong arithmetical tools

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

Option 1.: Find one.



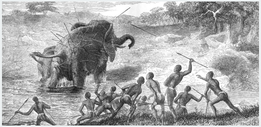
Option 2.: Prove that a **generic** elephant has the property P.

$$\underline{\mathbb{P}(P) > 1 - \epsilon}$$

- ⇝ strong arithmetical tools
- ⇝ effective computations

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

Option 1.: Find one.



Option 2.: Prove that a **generic** elephant has the property P.

$$\underline{\mathbb{P}(P) > 1 - \epsilon}$$

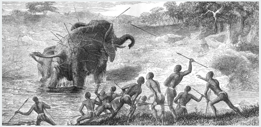
⇝ strong arithmetical tools

⇝ effective computations

⇝ infinitary properties:

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

Option 1.: Find one.



Option 2.: Prove that a **generic** elephant has the property P.

$$\underline{\mathbb{P}(P) > 1 - \epsilon}$$

$\leadsto$ strong arithmetical tools

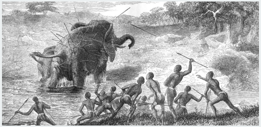
$\leadsto$ effective computations

$\leadsto$ infinitary properties:

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

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

Option 1.: Find one.



Option 2.: Prove that a **generic** elephant has the property P.

$$\underline{\mathbb{P}(P) > 1-\epsilon}$$

⇝ strong arithmetical tools

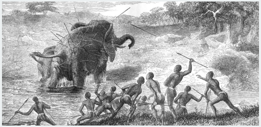
⇝ effective computations

⇝ infinitary properties:

$$\underline{\forall_{n\in\omega}\ \big(\mathbb{P}\left(P_n\right)=1\big) \quad\Longrightarrow\quad \mathbb{P}\left(\bigcap_{n\in\omega} P_n\right) = 1}$$

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

Option 1.: Find one.



Option 2.: Prove that a **generic** elephant has the property P.

$$\underline{\mathbb{P}(P) > 1 - \epsilon}$$

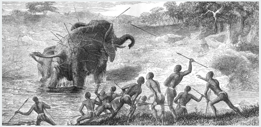
- ⇝ strong arithmetical tools
- ⇝ effective computations
- ⇝ infinitary properties:

but:

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

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

Option 1.: Find one.



Option 2.: Prove that a **generic** elephant has the property P.

$$\underline{\mathbb{P}(P) > 1 - \epsilon}$$

- ⇝ strong arithmetical tools
- ⇝ effective computations
- ⇝ infinitary properties:

but:

limitations of **quantitativity**

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

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

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.

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)

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 <u>dense</u> 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)

[thus non-empty]

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

⇝ 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)

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 <u>dense</u> 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)

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

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)

[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

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

Example

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

Theorem (Baire)

[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)

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 <u>dense</u> 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)

[thus non-empty]

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

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:

$$\begin{array}{rl} (\mathrm{I}): & 0, \\ (\mathrm{II}): & \end{array}$$

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

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

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

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:

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

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

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

$$\begin{array}{rlllll}
\text{(I):} & 0, & \underline{43226} & & \underline{13} & & \underline{8723466} \\
\text{(II):} & & & \underline{19743} & & \underline{\ } &
\end{array}$$

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

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

$$\begin{array}{rlllllll} \text{(I)}: & 0, & \underline{43226} & & \underline{13} & & \underline{8723466} & \\ \text{(II)}: & & & \underline{19743} & & \underline{\ } & & \underline{54326} \end{array}$$

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

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

$$\begin{array}{rllllllll}
\text{(I)}: & 0, & \underline{43226} & & \underline{13} & & \underline{8723466} & & \\
\text{(II)}: & & & \underline{19743} & & \underline{\ } & & \underline{54326} & \cdots
\end{array}$$

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

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

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

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

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

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

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

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

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

Theorem (Banach–Mazur)

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** $\pi \in W$

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

Theorem (Banach–Mazur)

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

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)

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** $\pi \in W$

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

Theorem (Banach–Mazur)

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** $\pi \in W$

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

Theorem (Banach–Mazur)

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** $\pi \in W$

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

Theorem (Banach–Mazur)

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).

($\Leftarrow$)

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

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

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

Theorem (Banach–Mazur)

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.

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

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

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

Theorem (Banach–Mazur)

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** $\pi \in W$

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

Theorem (Banach–Mazur)

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** $\pi \in W$

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

Theorem (Banach–Mazur)

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** $\pi \in W$

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

Theorem (Banach–Mazur)

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 $\mathrm{BM}(W)$ **iff**

$\big([0,1] - W\big)$ 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 $\underbrace{\text{finite duration}}_{\textbf{no}\ \text{infinite play} \ \equiv\ \text{well-founded game graph}}$ is determined.

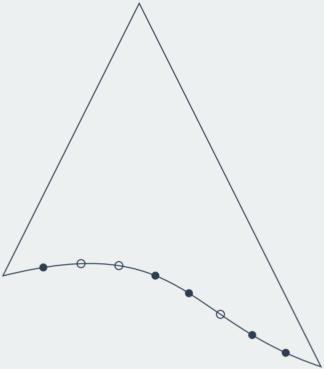
A game is **determined** if either (I) or (II) has a winning strategy.

- Every game of $\underbrace{\text{finite duration}}$ is determined.

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

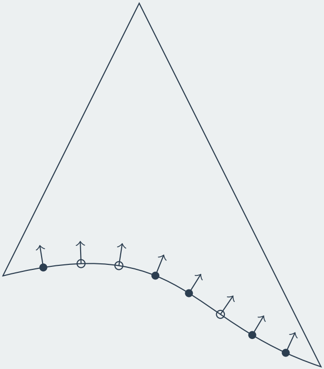
A game is **determined** if either (I) or (II) has a winning strategy.

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



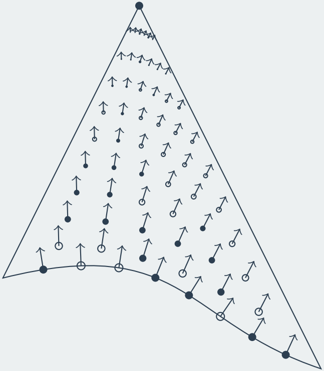
A game is **determined** if either (I) or (II) has a winning strategy.

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



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

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



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

- Every game of finite duration is determined.

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]; . . .))

$011001110101111011110101 \cdots \in \mathrm{XOR}$

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

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

iff

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

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

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

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 !**

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 !**

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

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 !**

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

(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. . .]

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

iff

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

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

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

(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. . .]

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

iff

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

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

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

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

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

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

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

iff

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

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

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

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

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

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

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

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

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

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

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

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

Proof: "strategy stealing"

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

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

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

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

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

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

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

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

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

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

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)

σ_{I}:

(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}:

(II):

(I):

σ_{II}:

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

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

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

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)

σ_{I}:

(II):

(I): $\underline{r_0}$

σ_{II}:

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

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

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

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)

$$\begin{array}{rl} \sigma_{\mathrm{I}}: & \underline{s_0} \\ \text{(II)}: & \end{array}$$

$$\begin{array}{rl} \text{(I)}: & \underline{r_0} \\ \sigma_{\mathrm{II}}: & \end{array}$$

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

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

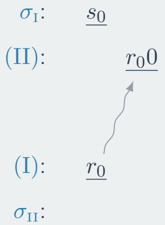
(II): $\quad\quad$ $\underline{11011}$ $\quad$ $\underline{1}$ $\quad\quad$ $\underline{00011}$ $\quad \cdots \rightsquigarrow \pi \in \{0,1\}^\omega$

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

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): $\underline{r_0 0}$

(I): $\underline{r_0}$

σ_{II}:

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

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

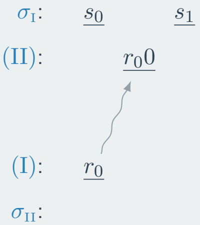
$$\begin{array}{lllllllll} \text{(I):} & \underline{01100} & & \underline{00} & & \underline{110010} & & & \\ \text{(II):} & & \underline{11011} & & \underline{1} & & \underline{00011} & \cdots & \rightsquigarrow \pi \in \{0,1\}^\omega \end{array}$$

$\big(\text{(I)}$ 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)



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

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

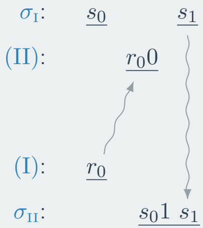
$$\begin{array}{rlllllll} \mathrm{(I)}: & \underline{01100} & & \underline{00} & & \underline{110010} & & \\ \mathrm{(II)}: & & \underline{11011} & & \underline{1} & & \underline{00011} & \cdots \rightsquigarrow \pi \in \{0,1\}^\omega \end{array}$$

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

Proof: "strategy stealing"

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

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



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

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

$\mathrm{(I)}$: $\underline{r_0}$

σ_{II}: $\underline{s_0 1\ s_1}$

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

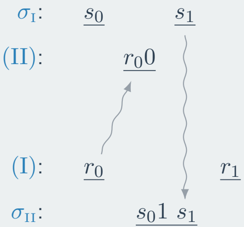
$$\begin{array}{rllllllll}(\mathrm{I}): & \underline{01100} & & \underline{00} & & \underline{110010} & & & \\ (\mathrm{II}): & & \underline{11011} & & \underline{1} & & \underline{00011} & \cdots & \leadsto \pi \in \{0,1\}^{\omega}\end{array}$$

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

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)



$$\begin{array}{rllll}\sigma_{\mathrm{I}}: & \underline{s_0} & & \underline{s_1} & \\ (\mathrm{II}): & & \underline{r_0 0} & & \\ (\mathrm{I}): & \underline{r_0} & & & \underline{r_1} \\ \sigma_{\mathrm{II}}: & & \underline{s_0 1 \ s_1} & & \end{array}$$

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

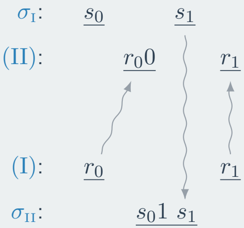
$$\begin{array}{rccccccll} (\mathrm{I}): & \underline{01100} & & \underline{00} & & \underline{110010} & & & \\ (\mathrm{II}): & & \underline{11011} & & \underline{1} & & \underline{00011} & \cdots & \rightsquigarrow \pi \in \{0,1\}^\omega \end{array}$$

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

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)


σ_{I}: $\underline{s_0}$ $\underline{s_1}$
(II): $\underline{r_0 0}$ $\underline{r_1}$
(I): $\underline{r_0}$ $\underline{r_1}$
σ_{II}: $\underline{s_0 1\ s_1}$

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

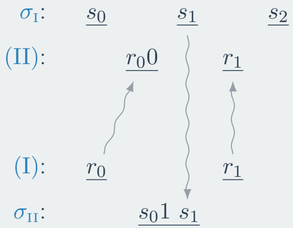
$$\begin{array}{rcccccccl}
(\mathrm{I}): & \underline{01100} & & \underline{00} & & \underline{110010} & & & \\
(\mathrm{II}): & & \underline{11011} & & \underline{1} & & \underline{00011} & \cdots & \rightsquigarrow \pi \in \{0,1\}^\omega
\end{array}$$

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

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)



σ_{I}: $\underline{s_0}$ $\underline{s_1}$ $\underline{s_2}$
(II): $\underline{r_0 0}$ $\underline{r_1}$
(I): $\underline{r_0}$ $\underline{r_1}$
σ_{II}: $\underline{s_0 1\ s_1}$

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

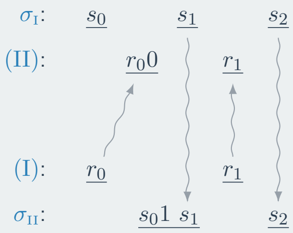
$$\begin{array}{rcccccccl} \text{(I):} & \underline{01100} & & \underline{00} & & \underline{110010} & & \cdots & \rightsquigarrow \pi\in\{0,1\}^\omega \\ \text{(II):} & & \underline{11011} & & \underline{1} & & \underline{00011} & & \end{array}$$

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

Proof: "strategy stealing"

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

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



$$\begin{array}{rccccc} \sigma_{\mathrm{I}}: & \underline{s_0} & & \underline{s_1} & & \underline{s_2} \\ \text{(II):} & & \underline{r_0 0} & & \underline{r_1} & \\ \text{(I):} & \underline{r_0} & & & \underline{r_1} & \\ \sigma_{\mathrm{II}}: & & \underline{s_0 1\ s_1} & & & \underline{s_2} \end{array}$$

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

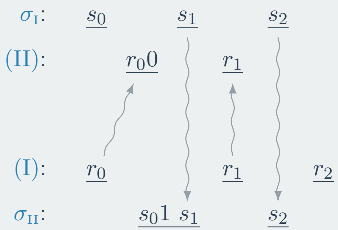
$$\begin{array}{rccccccccl} (\mathrm{I}): & \underline{01100} & & \underline{00} & & \underline{110010} & & & & \\ (\mathrm{II}): & & \underline{11011} & & \underline{1} & & \underline{00011} & \cdots & \rightsquigarrow & \pi \in \{0,1\}^\omega \end{array}$$

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

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)



$$\begin{array}{rccccccc} \sigma_{\mathrm{I}}: & \underline{s_0} & & \underline{s_1} & & \underline{s_2} & \\ (\mathrm{II}): & & \underline{r_0 0} & & \underline{r_1} & & \\ (\mathrm{I}): & \underline{r_0} & & & \underline{r_1} & & \underline{r_2} \\ \sigma_{\mathrm{II}}: & & \underline{s_0 1\ s_1} & & & \underline{s_2} & \end{array}$$

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

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

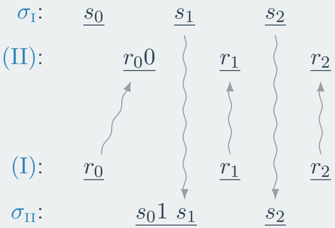
$$\begin{array}{rccccccl} \text{(I):} & \underline{01100} & & \underline{00} & & \underline{110010} & & \\ \text{(II):} & & \underline{11011} & & \underline{1} & & \underline{00011} & \cdots \rightsquigarrow \pi \in \{0,1\}^\omega \end{array}$$

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

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)



σ_{I}: $\underline{s_0}$ $\underline{s_1}$ $\underline{s_2}$
(II): $\underline{r_0 0}$ $\underline{r_1}$ $\underline{r_2}$
(I): $\underline{r_0}$ $\underline{r_1}$ $\underline{r_2}$
σ_{II}: $\underline{s_0 1\ s_1}$ $\underline{s_2}$

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

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

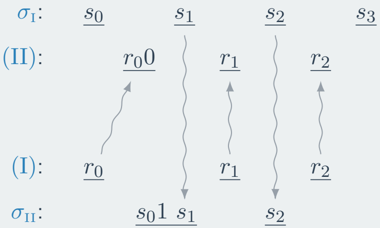
$$\begin{array}{lccccccll} \text{(I):} & \underline{01100} & & \underline{00} & & \underline{110010} & & & \\ \text{(II):} & & \underline{11011} & & \underline{1} & & \underline{00011} & \cdots & \rightsquigarrow \pi \in \{0,1\}^{\omega} \end{array}$$

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

Proof: **“**strategy stealing**”**

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)



$$\begin{array}{lcccccccc} \sigma_{\mathrm{I}}: & \underline{s_0} & & \underline{s_1} & & \underline{s_2} & & \underline{s_3} \\ \text{(II):} & & \underline{r_0 0} & & \underline{r_1} & & \underline{r_2} & \\ \text{(I):} & \underline{r_0} & & & \underline{r_1} & & \underline{r_2} & \\ \sigma_{\mathrm{II}}: & & \underline{s_0 1 \ s_1} & & & \underline{s_2} & & \end{array}$$

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

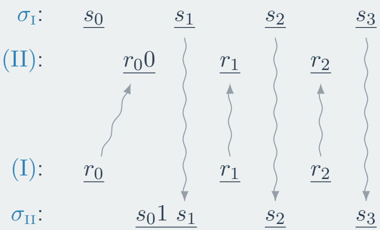
$$\begin{array}{rcccccccl} (\mathrm{I}): & \underline{01100} & & \underline{00} & & \underline{110010} & & & \\ (\mathrm{II}): & & \underline{11011} & & \underline{1} & & \underline{00011} & \cdots & \rightsquigarrow \pi \in \{0,1\}^\omega \end{array}$$

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

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)



$$\begin{array}{rccccccc} \sigma_{\mathrm{I}}: & \underline{s_0} & & \underline{s_1} & & \underline{s_2} & & \underline{s_3} \\ (\mathrm{II}): & & \underline{r_0 0} & & \underline{r_1} & & \underline{r_2} & \\ (\mathrm{I}): & \underline{r_0} & & & \underline{r_1} & & \underline{r_2} & \\ \sigma_{\mathrm{II}}: & & \underline{s_0 1\ s_1} & & & \underline{s_2} & & \underline{s_3} \end{array}$$

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

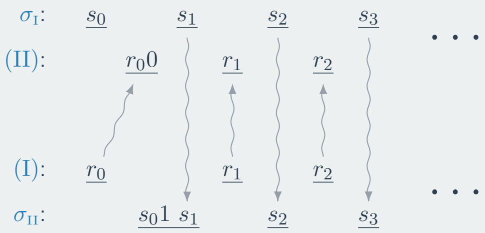
$$\begin{array}{rcccccccl}
(\mathrm{I}): & \underline{01100} & & \underline{00} & & \underline{110010} & & & \\
(\mathrm{II}): & & \underline{11011} & & \underline{1} & & \underline{00011} & \cdots & \rightsquigarrow \pi \in \{0,1\}^\omega
\end{array}$$

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

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)


σ_{I}: $\underline{s_0}$ $\underline{s_1}$ $\underline{s_2}$ $\underline{s_3}$ . . .
(II): $\underline{r_0 0}$ $\underline{r_1}$ $\underline{r_2}$
(I): $\underline{r_0}$ $\underline{r_1}$ $\underline{r_2}$
σ_{II}: $\underline{s_0 1\ s_1}$ $\underline{s_2}$ $\underline{s_3}$ . . .

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

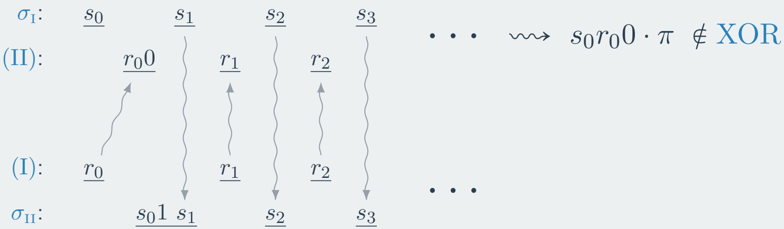
$$\begin{array}{rlllllllll} \text{(I):} & \underline{01100} & & \underline{00} & & \underline{110010} & & & & \\ \text{(II):} & & \underline{11011} & & \underline{1} & & \underline{00011} & \cdots & \rightsquigarrow & \pi \in \{0,1\}^{\omega} \end{array}$$

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

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)



$$\begin{array}{rllllllll} \sigma_{\mathrm{I}}: & \underline{s_0} & & \underline{s_1} & & \underline{s_2} & & \underline{s_3} & \cdots \rightsquigarrow s_0 r_0 0 \cdot \pi \notin \mathrm{XOR} \\ \text{(II):} & & \underline{r_0 0} & & \underline{r_1} & & \underline{r_2} & & \\ \text{(I):} & \underline{r_0} & & & \underline{r_1} & & \underline{r_2} & & \cdots \\ \sigma_{\mathrm{II}}: & & \underline{s_0 1\ s_1} & & & \underline{s_2} & & \underline{s_3} & \end{array}$$

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

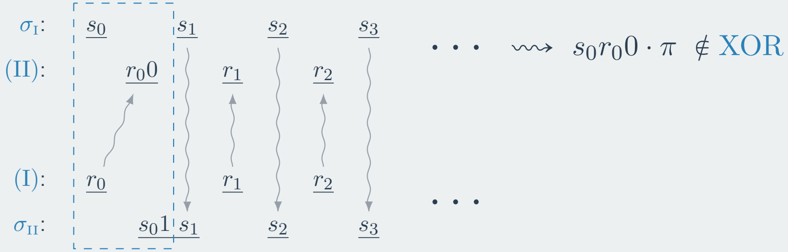
$$\begin{array}{rllllllll} (\mathrm{I}): & \underline{01100} & & \underline{00} & & \underline{110010} & & \cdots & \rightsquigarrow \pi \in \{0,1\}^\omega \\ (\mathrm{II}): & & \underline{11011} & & \underline{1} & & \underline{00011} & & \end{array}$$

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

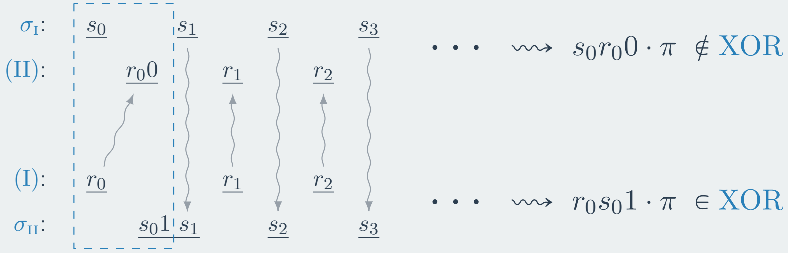
$$\begin{array}{lcccccccl}
\text{(I):} & \underline{01100} & & \underline{00} & & \underline{110010} & & & \\
\text{(II):} & & \underline{11011} & & \underline{1} & & \underline{00011} & \cdots & \rightsquigarrow \pi \in \{0,1\}^{\omega}
\end{array}$$

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

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

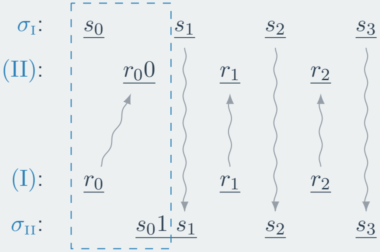
$$\begin{array}{rcccccccc}
(\mathrm{I}): & \underline{01100} & & \underline{00} & & \underline{110010} & & & \\
(\mathrm{II}): & & \underline{11011} & & \underline{1} & & \underline{00011} & \cdots & \rightsquigarrow \pi \in \{0,1\}^\omega
\end{array}$$

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

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)



$\cdots \rightsquigarrow s_0 r_0 0 \cdot \pi \notin \mathrm{XOR}$

$\cdots \rightsquigarrow r_0 s_0 1 \cdot \pi \in \mathrm{XOR}$

$\rightsquigarrow$ σ_{II} is a **winning strategy** 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): $\quad$ $\underline{11011}$ $\quad$ $\underline{1}$ $\quad$ $\underline{00011}$ $\quad \cdots \rightsquigarrow \pi \in \{0,1\}^\omega$

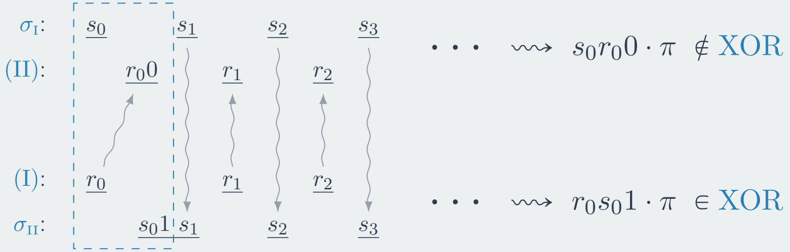
$$\big((\mathrm{I}) \text{ has a } \textbf{w.s.}\big) \implies \big((\mathrm{II}) \text{ has a } \textbf{w.s.}\big)$$

Proof: "strategy stealing"

Take σ_{I} — a **w.s.** of (I)

Construct σ_{II} — a **w.s.** of (II)

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



σ_{I}: $\underline{s_0}$ $\quad$ $\underline{s_1}$ $\quad$ $\underline{s_2}$ $\quad$ $\underline{s_3}$ $\quad \cdots \rightsquigarrow s_0 r_0 0 \cdot \pi \notin \mathrm{XOR}$

(II): $\underline{r_0 0}$ $\quad$ $\underline{r_1}$ $\quad$ $\underline{r_2}$

(I): $\underline{r_0}$ $\quad$ $\underline{r_1}$ $\quad$ $\underline{r_2}$

σ_{II}: $\underline{s_0 1}$ $\underline{s_1}$ $\quad$ $\underline{s_2}$ $\quad$ $\underline{s_3}$ $\quad \cdots \rightsquigarrow r_0 s_0 1 \cdot \pi \in \mathrm{XOR}$

$\rightsquigarrow$ 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,

when the winning condition is **Borel**.

Many variants:

· Blackwell games

· Nash equilibria

· . . .

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

Fix a finite set $A = \{a, b, c, \ldots\}$.

Definition

Fix a finite set $A = \{a, b, c, \ldots\}$.

A set $L \subseteq A^\omega$ is regular if

Definition

Fix a finite set $A = \{a, b, c, \ldots\}$.

A set $L \subseteq A^\omega$ is regular if

L can be defined in Monadic Second-order logic:

Definition

Fix a finite set $A = \{a, b, c, \ldots\}$.

A set $L \subseteq A^\omega$ is regular if

L can be defined in Monadic Second-order logic:

— first-order ($\exists_{x \in \omega}$) and monadic second-order ($\exists_{X \subseteq \omega}$) quantifiers,

Definition

Fix a finite set $A = \{a, b, c, \ldots\}$.

A set $L \subseteq A^\omega$ is regular if

L can be defined in Monadic Second-order logic:

— first-order ($\exists_{x\in\omega}$) and monadic second-order ($\exists_{X\subseteq\omega}$) quantifiers,

— Boolean connectives ($\vee$, $\wedge$, $\neg$),

Definition

Fix a finite set $A = \{a, b, c, \ldots\}$.

A set $L \subseteq A^\omega$ is regular if

L can be defined in Monadic Second-order logic:

— first-order ($\exists_{x \in \omega}$) and monadic second-order ($\exists_{X \subseteq \omega}$) quantifiers,

— Boolean connectives ($\vee$, $\wedge$, $\neg$),

— atomic predicates: $a(x)$, $x \leqslant y$, $x \in X$.

Definition

Fix a finite set $A = \{a, b, c, \ldots\}$.

A set $L \subseteq A^\omega$ is regular if

L can be defined in Monadic Second-order logic:

— first-order ($\exists_{x \in \omega}$) and monadic second-order ($\exists_{X \subseteq \omega}$) quantifiers,

— Boolean connectives ($\vee$, $\wedge$, $\neg$),

— atomic predicates: $a(x)$, $x \leqslant y$, $x \in X$.

$$\forall_{x \in \omega}\ \exists_{y \in \omega}\ \big(x \leqslant y\ \wedge\ a(y)\big)$$

Definition

Fix a finite set $A = \{a, b, c, \ldots\}$.

A set $L \subseteq A^\omega$ is regular if

L can be defined in Monadic Second-order logic:

— first-order ($\exists_{x \in \omega}$) and monadic second-order ($\exists_{X \subseteq \omega}$) quantifiers,

— Boolean connectives ($\vee$, $\wedge$, $\neg$),

— atomic predicates: $a(x)$, $x \leqslant y$, $x \in X$.

$$\boxed{\forall_{x \in \omega}\ \exists_{y \in \omega}\ \big(x \leqslant y\ \wedge\ a(y)\big)\ \ \overset{\mathrm{L}(\varphi)}{\rightsquigarrow}}$$

Definition

Fix a finite set $A = \{a, b, c, \ldots\}$.

A set $L \subseteq A^\omega$ is regular if

L can be defined in Monadic Second-order logic:

— first-order ($\exists_{x\in\omega}$) and monadic second-order ($\exists_{X\subseteq\omega}$) quantifiers,

— Boolean connectives ($\vee$, $\wedge$, $\neg$),

— atomic predicates: $a(x)$, $x \leqslant y$, $x \in X$.

$$\forall_{x\in\omega}\ \exists_{y\in\omega}\ \big(x \leqslant y\ \wedge\ a(y)\big)\ \overset{\mathrm{L}(\varphi)}{\rightsquigarrow}\ \{\alpha \in A^\omega \mid$$

Definition

Fix a finite set $A = \{a, b, c, \ldots\}$.

A set $L \subseteq A^\omega$ is regular if

L can be defined in Monadic Second-order logic:

— first-order ($\exists_{x\in\omega}$) and monadic second-order ($\exists_{X\subseteq\omega}$) quantifiers,

— Boolean connectives ($\vee$, $\wedge$, $\neg$),

— atomic predicates: $a(x)$, $x \leqslant y$, $x \in X$.

$$\boxed{\forall_{x\in\omega}\ \exists_{y\in\omega}\ \big(x \leqslant y\ \wedge\ a(y)\big)\ \overset{\mathrm{L}(\varphi)}{\rightsquigarrow}\ \{\alpha \in A^\omega \mid \alpha \text{ has infinitely many } a\}}$$

Definition

Fix a finite set $A = \{a, b, c, \ldots\}$.

A set $L \subseteq A^\omega$ is regular if

L can be defined in Monadic Second-order logic:

— first-order ($\exists_{x\in\omega}$) and monadic second-order ($\exists_{X\subseteq\omega}$) quantifiers,

— Boolean connectives ($\vee$, $\wedge$, $\neg$),

— atomic predicates: $a(x)$, $x \leqslant y$, $x \in X$.

$$\forall_{x\in\omega}\ \exists_{y\in\omega}\ \big(x \leqslant y\ \wedge\ a(y)\big) \overset{\mathrm{L}(\varphi)}{\rightsquigarrow} \{\alpha \in A^\omega \mid \alpha \text{ has infinitely many } a\}$$

In other words

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

Definition

Fix a finite set $A = \{a, b, c, \ldots\}$.

A set $L \subseteq A^\omega$ is regular if

L can be defined in Monadic Second-order logic:

— first-order ($\exists_{x \in \omega}$) and monadic second-order ($\exists_{X \subseteq \omega}$) quantifiers,

— Boolean connectives ($\vee$, $\wedge$, $\neg$),

— atomic predicates: $a(x)$, $x \leqslant y$, $x \in X$.

$$\boxed{\forall_{x \in \omega} \, \exists_{y \in \omega} \, \big(x \leqslant y \;\wedge\; a(y)\big) \quad \overset{\mathrm{L}(\varphi)}{\rightsquigarrow} \quad \{\alpha \in A^\omega \mid \alpha \text{ has infinitely many } a\}}$$

In other words

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

— contains some basic languages, and

Definition

Fix a finite set $A = \{a, b, c, \ldots\}$.

A set $L \subseteq A^\omega$ is regular if

L can be defined in Monadic Second-order logic:

— first-order ($\exists_{x\in\omega}$) and monadic second-order ($\exists_{X\subseteq\omega}$) quantifiers,

— Boolean connectives ($\vee$, $\wedge$, $\neg$),

— atomic predicates: $a(x)$, $x \leqslant y$, $x \in X$.

$$\boxed{\forall_{x\in\omega}\ \exists_{y\in\omega}\ \big(x \leqslant y\ \wedge\ a(y)\big) \ \overset{\mathrm{L}(\varphi)}{\rightsquigarrow}\ \big\{\alpha \in A^\omega \mid \alpha \text{ has infinitely many } a\big\}}$$

In other words

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

— contains some basic languages, and

— is **closed** under Boolean operations and projection $\big(A \times B\big)^\omega \to A^\omega$.

Definition

Fix a finite set $A = \{a, b, c, \ldots\}$.

A set $L \subseteq A^\omega$ is regular if

L can be defined in Monadic Second-order logic:

— first-order ($\exists_{x\in\omega}$) and monadic second-order ($\exists_{X\subseteq\omega}$) quantifiers,

— Boolean connectives ($\vee$, $\wedge$, $\neg$),

— atomic predicates: $a(x)$, $x \leqslant y$, $x \in X$.

$$\boxed{\forall_{x\in\omega}\ \exists_{y\in\omega}\ \big(x \leqslant y\ \wedge\ a(y)\big)\ \overset{\mathrm{L}(\varphi)}{\rightsquigarrow}\ \{\alpha \in A^\omega \mid \alpha \text{ has infinitely many } a\}}$$

In other words

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

— contains some basic languages, and

— is **closed** under Boolean operations and projection $\big(A\times B\big)^\omega \to A^\omega$.

Facts:

Definition

Fix a finite set $A = \{a, b, c, \ldots\}$.

A set $L \subseteq A^\omega$ is regular if

L can be defined in Monadic Second-order logic:

— first-order ($\exists_{x\in\omega}$) and monadic second-order ($\exists_{X\subseteq\omega}$) quantifiers,

— Boolean connectives ($\vee$, $\wedge$, $\neg$),

— atomic predicates: $a(x)$, $x \leqslant y$, $x \in X$.

$$\boxed{\forall_{x\in\omega}\ \exists_{y\in\omega}\ \big(x \leqslant y\ \wedge\ a(y)\big)\ \overset{\mathrm{L}(\varphi)}{\rightsquigarrow}\ \{\alpha \in A^\omega \mid \alpha \text{ has infinitely many } a\}}$$

In other words

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

— contains some basic languages, and

— is **closed** under Boolean operations and projection $\big(A{\times}B\big)^\omega \to A^\omega$.

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

Definition

Fix a finite set $A = \{a, b, c, \ldots\}$.

A set $L \subseteq A^{\omega}$ is regular if

L can be defined in Monadic Second-order logic:

— first-order ($\exists_{x\in\omega}$) and monadic second-order ($\exists_{X\subseteq\omega}$) quantifiers,

— Boolean connectives ($\vee$, $\wedge$, $\neg$),

— atomic predicates: $a(x)$, $x \leqslant y$, $x \in X$.

$$\boxed{\forall_{x\in\omega}\ \exists_{y\in\omega}\ \big(x \leqslant y\ \wedge\ a(y)\big)\ \overset{\mathrm{L}(\varphi)}{\rightsquigarrow}\ \{\alpha \in A^{\omega} \mid \alpha \text{ has infinitely many } a\}}$$

In other words

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

— contains some basic languages, and

— is **closed** under Boolean operations and projection $\big(A\times B\big)^{\omega} \to A^{\omega}$.

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

Definition

Fix a finite set $A = \{a, b, c, \ldots\}$.

A set $L \subseteq A^\omega$ is regular if

L can be defined in Monadic Second-order logic:

— first-order ($\exists_{x\in\omega}$) and monadic second-order ($\exists_{X\subseteq\omega}$) quantifiers,

— Boolean connectives ($\vee$, $\wedge$, $\neg$),

— atomic predicates: $a(x)$, $x \leqslant y$, $x \in X$.

$$\boxed{\forall_{x\in\omega}\ \exists_{y\in\omega}\ \big(x \leqslant y \ \wedge\ a(y)\big) \ \overset{\mathrm{L}(\varphi)}{\rightsquigarrow}\ \{\alpha \in A^\omega \mid \alpha \text{ has infinitely many } a\}}$$

In other words

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

— contains some basic languages, and

— is **closed** under Boolean operations and projection $\big(A\times B\big)^\omega \to A^\omega$.

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

Definition

Fix a finite set $A = \{a, b, c, \ldots\}$.

A set $L \subseteq A^\omega$ is regular if

L can be defined in Monadic Second-order logic:

— first-order ($\exists_{x\in\omega}$) and monadic second-order ($\exists_{X\subseteq\omega}$) quantifiers,

— Boolean connectives ($\vee$, $\wedge$, $\neg$),

— atomic predicates: $a(x)$, $x \leqslant y$, $x \in X$.

$$\boxed{\forall_{x\in\omega}\ \exists_{y\in\omega}\ \big(x \leqslant y\ \wedge\ a(y)\big)\ \overset{\mathrm{L}(\varphi)}{\rightsquigarrow}\ \{\alpha \in A^\omega \mid \alpha \text{ has infinitely many } a\}}$$

In other words

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

— contains some basic languages, and

— is **closed** under Boolean operations and projection $\big(A\times B\big)^\omega \to A^\omega$.

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

Every $L \in \mathbf{REG}$ has a finite representation φ such that $\mathrm{L}(\varphi) = L$.

Theorem (Büchi ['62])

Given φ it is **decidable** if $\mathrm{L}(\varphi) \neq \emptyset$.

Theorem (Büchi ['62])

Given φ it is **decidable** if $\mathrm{L}(\varphi) \neq \emptyset$.

Proof

Using automata ($\varphi \mapsto \mathcal{A}_\varphi$) and Ramsey argument. ■

Theorem (Büchi ['62])

Given φ it is **decidable** if $\mathrm{L}(\varphi) \neq \emptyset$.

Proof

Using automata ($\varphi \mapsto \mathcal{A}_\varphi$) and Ramsey argument. ■

$\rightsquigarrow$ Decidability of:

Theorem (Büchi ['62])

Given φ it is **decidable** if $\mathrm{L}(\varphi) \neq \emptyset$.

Proof

Using automata ($\varphi \mapsto \mathcal{A}_\varphi$) and Ramsey argument. ■

$\leadsto$ Decidability of: $\mathrm{L}(\varphi) \overset{?}{=} A^\omega$,

Theorem (Büchi ['62])

Given φ it is **decidable** if $\mathrm{L}(\varphi) \neq \varnothing$.

Proof

Using automata ($\varphi \mapsto \mathcal{A}_{\varphi}$) and Ramsey argument. ■

⇝ Decidability of:

$$\begin{tikzcd} \mathrm{L}(\varphi) \overset{?}{=} A^{\omega}, \arrow[d, squiggly] \\ \mathrm{L}(\neg\varphi) \overset{?}{=} \varnothing \end{tikzcd}$$

Theorem (Büchi ['62])

Given φ it is **decidable** if $\mathrm{L}(\varphi) \neq \emptyset$.

Proof

Using automata ($\varphi \mapsto \mathcal{A}_\varphi$) and Ramsey argument. ■

$\rightsquigarrow$ Decidability of:

$$\begin{tikzcd}
\mathrm{L}(\varphi) \overset{?}{=} A^\omega, \quad \mathrm{L}(\psi) \overset{?}{\subseteq} \mathrm{L}(\varphi), \arrow[d, squiggly] \\
\mathrm{L}(\neg\varphi) \overset{?}{=} \emptyset
\end{tikzcd}$$

Theorem (Büchi ['62])

Given φ it is **decidable** if $\mathrm{L}(\varphi) \neq \varnothing$.

Proof

Using automata ($\varphi \mapsto \mathcal{A}_\varphi$) and Ramsey argument. ■

$\rightsquigarrow$ Decidability of:

$$\begin{tikzcd}
\mathrm{L}(\varphi) \overset{?}{=} A^\omega, \arrow[d, squiggly] & \mathrm{L}(\psi) \overset{?}{\subseteq} \mathrm{L}(\varphi), \arrow[d, squiggly] \\
\mathrm{L}(\neg\varphi) \overset{?}{=} \varnothing & \mathrm{L}(\psi \wedge \neg\varphi) \overset{?}{=} \varnothing
\end{tikzcd}$$

Theorem (Büchi ['62])

Given φ it is **decidable** if $\mathrm{L}(\varphi) \neq \emptyset$.

Proof

Using automata ($\varphi \mapsto \mathcal{A}_\varphi$) and Ramsey argument. ∎

$\rightsquigarrow$ Decidability of:

$$\begin{tikzcd} \mathrm{L}(\varphi)\overset{?}{=}A^\omega, \arrow[d, squiggly] & \mathrm{L}(\psi)\overset{?}{\subseteq}\mathrm{L}(\varphi), \arrow[d, squiggly] & \mathrm{L}(\psi)\overset{?}{=}\mathrm{L}(\varphi), & \dots \\ \mathrm{L}(\neg\varphi)\overset{?}{=}\emptyset & \mathrm{L}(\psi\wedge\neg\varphi)\overset{?}{=}\emptyset & & \end{tikzcd}$$

Theorem (Büchi ['62])

Given φ it is **decidable** if $\mathrm{L}(\varphi) \neq \emptyset$.

Proof

Using automata ($\varphi \mapsto \mathcal{A}_\varphi$) and Ramsey argument. ■

$\rightsquigarrow$ Decidability of:

$$\begin{tikzcd}
\mathrm{L}(\varphi) \overset{?}{=} A^\omega, \arrow[d, rightsquigarrow] & \mathrm{L}(\psi) \overset{?}{\subseteq} \mathrm{L}(\varphi), \arrow[d, rightsquigarrow] & \mathrm{L}(\psi) \overset{?}{=} \mathrm{L}(\varphi), & \dots \\
\mathrm{L}(\neg\varphi) \overset{?}{=} \emptyset & \mathrm{L}(\psi \wedge \neg\varphi) \overset{?}{=} \emptyset & &
\end{tikzcd}$$

$\rightsquigarrow$ Model-checking:

Theorem (Büchi ['62])

Given φ it is **decidable** if $\mathrm{L}(\varphi) \neq \emptyset$.

Proof

Using automata ($\varphi \mapsto \mathcal{A}_\varphi$) and Ramsey argument. ■

⇝ Decidability of:

$$\begin{tikzcd}[column sep=small]
\mathrm{L}(\varphi) \overset{?}{=} A^\omega, \arrow[d, squiggly] & \mathrm{L}(\psi) \overset{?}{\subseteq} \mathrm{L}(\varphi), \arrow[d, squiggly] & \mathrm{L}(\psi) \overset{?}{=} \mathrm{L}(\varphi), & \ldots \\
\mathrm{L}(\neg\varphi) \overset{?}{=} \emptyset & \mathrm{L}(\psi \wedge \neg\varphi) \overset{?}{=} \emptyset & &
\end{tikzcd}$$

⇝ Model-checking: given a machine M and a specification φ,

Theorem (Büchi ['62])

Given φ it is **decidable** if $\mathrm{L}(\varphi) \neq \emptyset$.

Proof

Using automata ($\varphi \mapsto \mathcal{A}_\varphi$) and Ramsey argument. ■

$\rightsquigarrow$ Decidability of:

$$\begin{tikzcd}
\mathrm{L}(\varphi) \overset{?}{=} A^\omega, \arrow[d, squiggly] & \mathrm{L}(\psi) \overset{?}{\subseteq} \mathrm{L}(\varphi), \arrow[d, squiggly] & \mathrm{L}(\psi) \overset{?}{=} \mathrm{L}(\varphi), & \ldots \\
\mathrm{L}(\neg\varphi) \overset{?}{=} \emptyset & \mathrm{L}(\psi \wedge \neg\varphi) \overset{?}{=} \emptyset & &
\end{tikzcd}$$

$\rightsquigarrow$ Model-checking: given a machine M and a specification φ,

decide if $M \models \varphi$.

Theorem (Büchi ['62])

Given φ it is **decidable** if $\mathrm{L}(\varphi) \neq \emptyset$.

Proof

Using automata ($\varphi \mapsto \mathcal{A}_\varphi$) and Ramsey argument. ■

⇝ Decidability of:

$$\begin{tikzcd} \mathrm{L}(\varphi)\overset{?}{=}A^{\omega}, \arrow[d, squiggly] & \mathrm{L}(\psi)\overset{?}{\subseteq}\mathrm{L}(\varphi), \arrow[d, squiggly] & \mathrm{L}(\psi)\overset{?}{=}\mathrm{L}(\varphi), & \ldots \\ \mathrm{L}(\neg\varphi)\overset{?}{=}\emptyset & \mathrm{L}(\psi\wedge\neg\varphi)\overset{?}{=}\emptyset & & \end{tikzcd}$$

⇝ Model-checking: given a machine M and a specification φ, decide if $M \models \varphi$.

1. Express behaviour of M as ψ_M.

Theorem (Büchi ['62])

Given φ it is **decidable** if $\mathrm{L}(\varphi) \neq \emptyset$.

Proof

Using automata ($\varphi \mapsto \mathcal{A}_\varphi$) and Ramsey argument. ■

⇝ Decidability of:

$$\begin{tikzcd} \mathrm{L}(\varphi) \overset{?}{=} A^\omega, \arrow[d, squiggly] & \mathrm{L}(\psi) \overset{?}{\subseteq} \mathrm{L}(\varphi), \arrow[d, squiggly] & \mathrm{L}(\psi) \overset{?}{=} \mathrm{L}(\varphi), & \ldots \\ \mathrm{L}(\neg\varphi) \overset{?}{=} \emptyset & \mathrm{L}(\psi \wedge \neg\varphi) \overset{?}{=} \emptyset & & \end{tikzcd}$$

⇝ Model-checking: given a machine M and a specification φ,

decide if $M \models \varphi$.

1. Express behaviour of M as ψ_M.
2. Verify if $\psi_M \Rightarrow \varphi$.

Theorem (Büchi ['62])

Given φ it is **decidable** if $\mathrm{L}(\varphi) \neq \varnothing$.

Proof

Using automata ($\varphi \mapsto \mathcal{A}_\varphi$) and Ramsey argument. ■

⇝ Decidability of:

$$\begin{tikzcd} \mathrm{L}(\varphi)\overset{?}{=}A^\omega \arrow[d, squiggly] & \mathrm{L}(\psi)\overset{?}{\subseteq}\mathrm{L}(\varphi) \arrow[d, squiggly] & \mathrm{L}(\psi)\overset{?}{=}\mathrm{L}(\varphi) & \dots \\ \mathrm{L}(\neg\varphi)\overset{?}{=}\varnothing & \mathrm{L}(\psi\wedge\neg\varphi)\overset{?}{=}\varnothing & & \end{tikzcd}$$

⇝ Model-checking: given a machine M and a specification φ,

decide if $M \models \varphi$.

1. Express behaviour of M as ψ_M.
2. Verify if $\psi_M \Rightarrow \varphi$.

[In fact: translate $\neg\varphi$ into $\mathcal{A}_{\neg\varphi}$ and check $M \times \mathcal{A}_{\neg\varphi}$ for emptiness]

Theorem (Büchi ['62])

Given φ it is **decidable** if $\mathrm{L}(\varphi) \neq \emptyset$.

Proof

Using automata ($\varphi \mapsto \mathcal{A}_\varphi$) and Ramsey argument. ■

⇝ Decidability of:

$$\begin{tikzcd}
\mathrm{L}(\varphi) \overset{?}{=} A^\omega, \arrow[d, squiggly] & \mathrm{L}(\psi) \overset{?}{\subseteq} \mathrm{L}(\varphi), \arrow[d, squiggly] & \mathrm{L}(\psi) \overset{?}{=} \mathrm{L}(\varphi), & \ldots \\
\mathrm{L}(\neg\varphi) \overset{?}{=} \emptyset & \mathrm{L}(\psi \wedge \neg\varphi) \overset{?}{=} \emptyset & &
\end{tikzcd}$$

⇝ Model-checking: given a machine M and a specification φ,

decide if $M \models \varphi$.

1. Express behaviour of M as ψ_M.

2. Verify if $\psi_M \Rightarrow \varphi$.

[In fact: translate $\neg\varphi$ into $\mathcal{A}_{\neg\varphi}$ and check $M \times \mathcal{A}_{\neg\varphi}$ for emptiness]

⇝ Working implementations (e.g. **MONA** from Aarhus)

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 $\mathcal{G}(W)$:

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

Theorem (Büchi, Landweber ['69])

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

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

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

(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 $\mathcal{G}(W)$:

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

(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 $\mathcal{G}(W)$:

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

(II) wins π **iff** $\pi \in W$

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 $\mathcal{G}(W)$:

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

(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 $\mathcal{G}(W)$:

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

(II) wins π **iff** $\pi \in W$

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 $\mathcal{G}(W)$:

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

(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,

Theorem (Büchi, Landweber ['69])

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

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

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

(II) wins π **iff** $\pi \in W$

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 $\mathcal{G}(W)$:

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

(II) wins π **iff** $\pi \in W$

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 $m_{i+1} \overset{\text{def}}{=} \delta(m_i, a_i)$,

Theorem (Büchi, Landweber ['69])

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

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

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

(II) wins π **iff** $\pi \in W$

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 $m_{i+1} \overset{\text{def}}{=} \delta(m_i, a_i)$,

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

Part 4

Applications

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 $G \subseteq A^\omega$. $\left[G = \mathrm{L}(\varphi_G)\right]$

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

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

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

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

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

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

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

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

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

(I):
(II):

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

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

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

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

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

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

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

$$\begin{array}{rll} \text{(I)}: & \underline{a_0} & \\ \text{(II)}: & & \underline{\flat} \end{array}$$

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

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

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

$$\begin{array}{rccc} \text{(I):} & \underline{a_0} & & \underline{a_1} \\ \text{(II):} & & \underline{\flat} & \end{array}$$

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

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

Construct a regular $W_G \subseteq \big(A \sqcup \{\flat\}\big)^\omega$: $\Big[\varphi_G \mapsto \varphi_{W_G} \text{ s.t. } \mathrm{L}(\varphi_{W_G}) = W_G\Big]$

$$\begin{array}{rcccc} \text{(I)}: & \underline{a_0} & & \underline{a_1} & \\ \text{(II)}: & & \underline{\flat} & & \underline{\flat} \end{array}$$

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

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

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

$$\begin{array}{rcccccc}
\text{(I):} & \underline{a_0} & & \underline{a_1} & & \underline{\flat} \\
\text{(II):} & & \underline{\flat} & & \underline{\flat} &
\end{array}$$

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

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

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

$$\begin{array}{lcccccc}
\text{(I):} & \underline{a_0} & & \underline{a_1} & & \underline{\flat} & \\
\text{(II):} & & \underline{\flat} & & \underline{\flat} & & \underline{a_2}
\end{array}$$

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

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

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

$$\begin{array}{lcccccccc} \text{(I)}: & \underline{a_0} & & \underline{a_1} & & \underline{\flat} & & \underline{\flat} \\ \text{(II)}: & & \underline{\flat} & & \underline{\flat} & & \underline{a_2} & \end{array}$$

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

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

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

$$\begin{array}{rcccccccc}
\text{(I)}: & \underline{a_0} & & \underline{a_1} & & \underline{\flat} & & \underline{\flat} & \\
\text{(II)}: & & \underline{\flat} & & \underline{\flat} & & \underline{a_2} & & \underline{a_3}
\end{array}$$

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

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

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

$$\begin{array}{rccccccccc} \text{(I)}: & \underline{a_0} & & \underline{a_1} & & \underline{\flat} & & \underline{\flat} & & \underline{\flat} \\ \text{(II)}: & & \underline{\flat} & & \underline{\flat} & & \underline{a_2} & & \underline{a_3} & \end{array}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Similarly with other game-characterised properties for regular sets:

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

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

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

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

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

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

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

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

Similarly with other game-characterised properties for regular sets:

— countability,

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

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

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

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

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

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

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

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

Similarly with other game-characterised properties for regular sets:

— countability,

— measure 0,

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

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

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

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

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

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

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

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

Similarly with other game-characterised properties for regular sets:

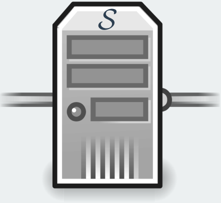
— countability,

— measure 0,

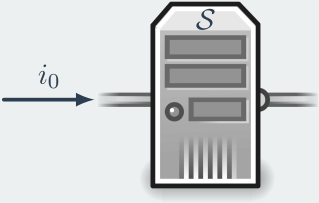
— Wadge reductions, . . .

Synthesis

Synthesis

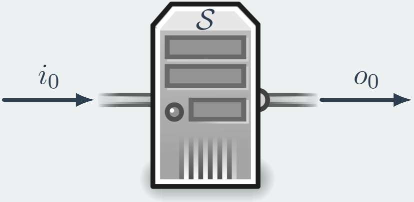


Synthesis



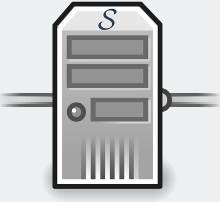
Trace $\tau = (i_0$

Synthesis



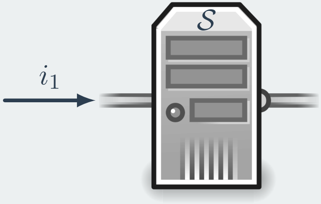
Trace $\tau = (i_0\ o_0$

Synthesis



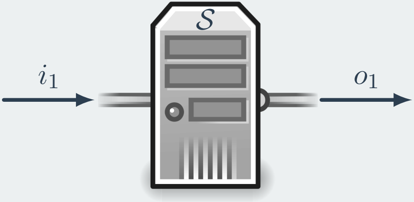
Trace $\tau = (i_0\ o_0$

Synthesis



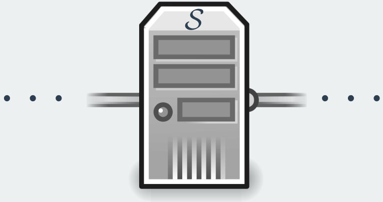
Trace $\tau = (i_0 \; o_0 \; i_1$

Synthesis



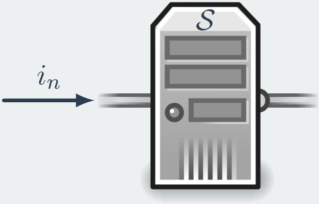
Trace $\tau = (i_0\ o_0\ i_1\ o_1$

Synthesis



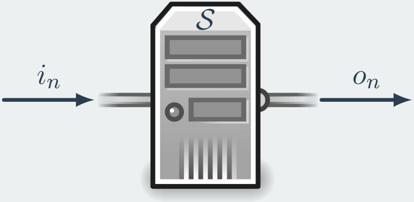
Trace $\tau = (i_0 \; o_0 \; i_1 \; o_1 \; \cdots$

Synthesis



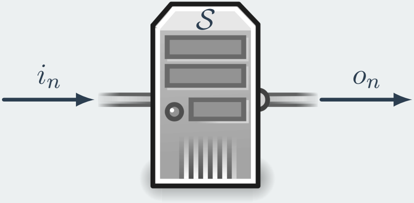
Trace $\tau = (i_0\ o_0\ i_1\ o_1\ \cdots\ i_n$

Synthesis



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

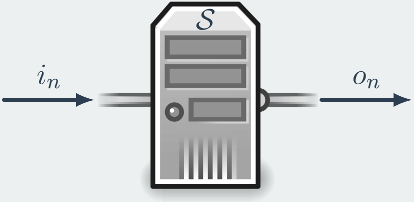
Synthesis



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

Specification
φ over $I \sqcup O$

Synthesis

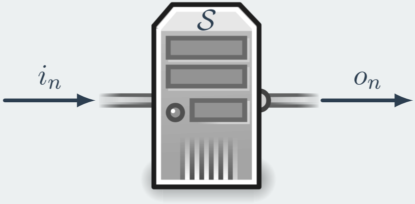


Trace $\tau = (i_0\ o_0\ i_1\ o_1\ \cdots\ i_n\ o_n\ \cdots) \in \big(I \sqcup O\big)^\omega$

Specification
φ over $I \sqcup O$

Synthesis $\longmapsto$

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

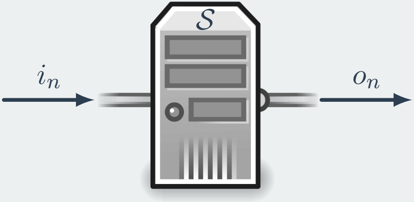


Trace $\tau = (i_0\ o_0\ i_1\ o_1\ \cdots\ i_n\ o_n\ \cdots) \in \left(I \sqcup O\right)^{\omega}$

Specification
φ over $I \sqcup O$

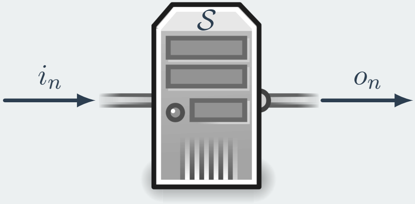
Synthesis $\longmapsto$

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



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

Specification φ over $I \sqcup O$ $\xrightarrow{\textbf{Synthesis}}$ Implementation $\mathcal{S}\colon I \rightsquigarrow O$ $\left[\underbrace{\text{whenever possible}}_{\varphi \equiv \text{“}o_0 = i_1\text{”}}\right]$



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

Specification

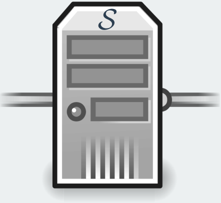
φ over $I \sqcup O$

$$\xrightarrow{\textbf{Synthesis}}$$

Implementation

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

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



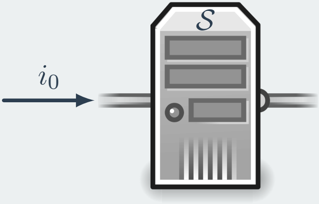
Env.:

Impl.:

Synthesis

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

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



Env.: $\underline{i_0}$

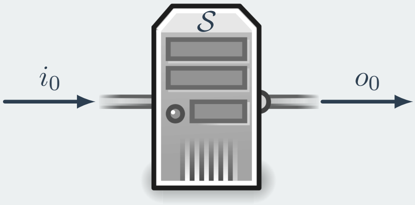
Impl.:

Specification
φ over $I \sqcup O$

Synthesis

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

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

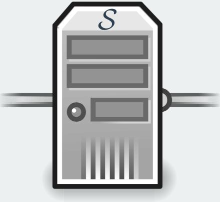


Env.: $\underline{i_0}$

Impl.: $\underline{o_0}$

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

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



Env.: $\underline{i_0}$

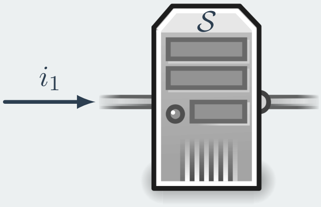
Impl.: $\underline{o_0}$

Specification
φ over $I \sqcup O$

$$\xrightarrow{\textbf{Synthesis}}$$

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

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



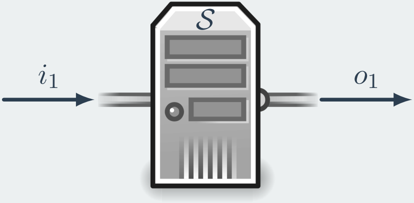
Env.:	$\underline{i_0}$		$\underline{i_1}$
Impl.:		$\underline{o_0}$	

Specification
φ over $I \sqcup O$

Synthesis $\longmapsto$

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

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



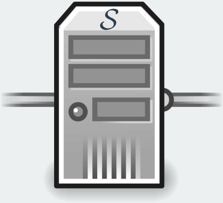
Env.:	$\underline{i_0}$		$\underline{i_1}$	
Impl.:		$\underline{o_0}$		$\underline{o_1}$

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

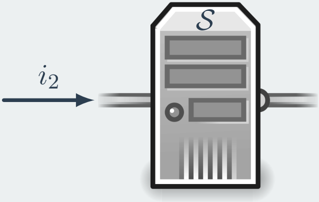
Impl.: $\underline{o_0}$ $\underline{o_1}$

Specification
φ over $I \sqcup O$

Synthesis

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

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



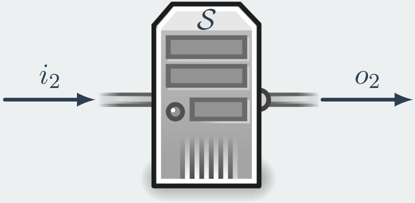
Env.: $\underline{i_0}$ $\underline{i_1}$ $\underline{i_2}$

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

Specification
φ over $I \sqcup O$

$\longmapsto$ **Synthesis**

Implementation
$\mathcal{S}\colon I \rightsquigarrow O$
$\underbrace{\left[\text{whenever possible}\right]}_{\varphi \equiv \text{“}o_0 = i_1\text{”}}$



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

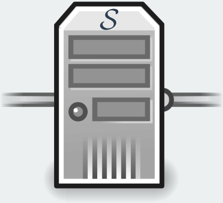
Impl.: $\underline{o_0}$ $\underline{o_1}$ $\underline{o_2}$

Specification
φ over $I \sqcup O$

Synthesis $\longmapsto$

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

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



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

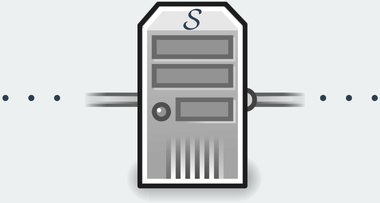
Impl.: $\underline{o_0}$ $\underline{o_1}$ $\underline{o_2}$

Specification
φ over $I \sqcup O$

$\xmapsto{\textbf{Synthesis}}$

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

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



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

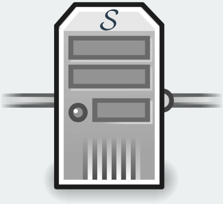
Impl.: $\underline{o_0}$ $\underline{o_1}$ $\underline{o_2}$ $\cdots$

Specification
φ over $I \sqcup O$

$$\longmapsto^{\textbf{Synthesis}}$$

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

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



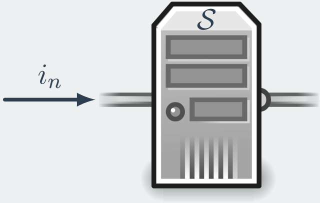
Env.:	$\underline{i_0}$		$\underline{i_1}$		$\underline{i_2}$		$\cdots$
Impl.:		$\underline{o_0}$		$\underline{o_1}$		$\underline{o_2}$	$\cdots$

Specification
φ over $I \sqcup O$

Synthesis

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

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



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

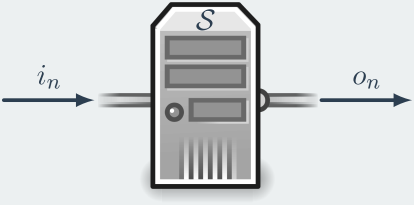
Impl.: $\underline{o_0}$ $\underline{o_1}$ $\underline{o_2}$ $\cdots$

Specification
φ over $I \sqcup O$

Synthesis $\longmapsto$

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

$\underbrace{\left[\text{whenever possible}\right]}_{\varphi \,\equiv\, \text{``}o_0 = i_1\text{''}}$



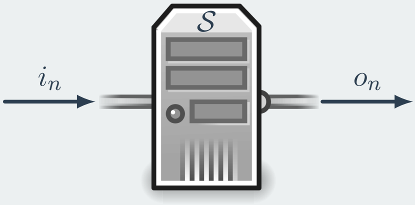
Env.:	$\underline{i_0}$		$\underline{i_1}$		$\underline{i_2}$		$\cdots$	$\underline{i_n}$	
Impl.:		$\underline{o_0}$		$\underline{o_1}$		$\underline{o_2}$	$\cdots$		$\underline{o_n}$

Specification
φ over $I \sqcup O$

Synthesis ⟶

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

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

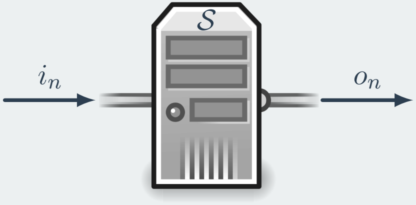


Env.: $\underline{i_0}$ $\underline{i_1}$ $\underline{i_2}$ $\cdots$ $\underline{i_n}$

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

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

$$\begin{array}{c} \text{Specification} \\ \varphi \text{ over } I \sqcup O \end{array} \xrightarrow{\textbf{Synthesis}} \begin{array}{c} \text{Implementation} \\ \mathcal{S}\colon I \rightsquigarrow O \\ \underbrace{\left[\text{whenever possible}\right]}_{\varphi \equiv \text{“}o_0 = i_1\text{”}} \end{array}$$



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

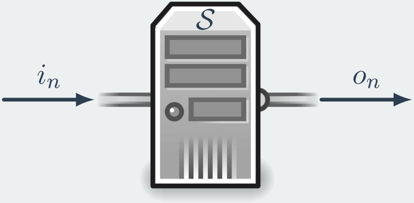
Specification
φ over $I \sqcup O$

$$\longmapsto$$

Synthesis

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

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



$$\begin{array}{lccccccccccc}
\textbf{Env.:} & \underline{i_0} & & \underline{i_1} & & \underline{i_2} & & \cdots & \underline{i_n} & & & \\
\textbf{Impl.:} & & \underline{o_0} & & \underline{o_1} & & \underline{o_2} & \cdots & & \underline{o_n} & \cdots \rightsquigarrow & \tau \overset{?}{\models} \varphi
\end{array}$$

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

$\swarrow$

(I) wins

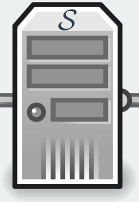
Specification
φ over $I \sqcup O$

Synthesis $\longmapsto$

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

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

$i_n \rightarrow$



$o_n \rightarrow$

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

$\swarrow$

(I) wins

$\Downarrow$

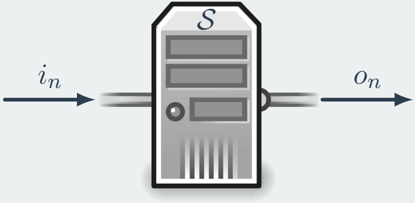
φ is unrealisable

Specification
φ over $I \sqcup O$

Synthesis $\longmapsto$

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

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



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

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

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

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

(I) wins

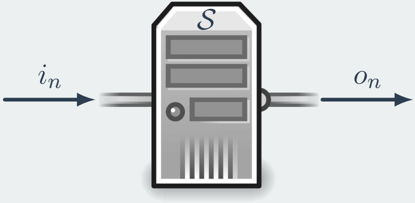
$\Downarrow$

φ is unrealisable

(II) wins

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

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



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

Impl.: $\underline{o_0}$ $\underline{o_1}$ $\underline{o_2}$ $\cdots$ $\underline{o_n}$ $\cdots \rightsquigarrow \tau \overset{?}{\models} \varphi$

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

(I) wins

$\Downarrow$

φ is unrealisable

(II) wins

$\Downarrow$

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

Part 5

Effective characterisations

Task: **understand** which $L \in \mathbf{REG}$ are **simple**.

Task: **understand** which $L \in \mathbf{REG}$ are **simple**.

Procedure:

Input: φ

Output: is $\mathrm{L}(\varphi)$ simple?

Task: **understand** which $L \in \mathbf{REG}$ are **simple**.

definable in a weaker logic (e.g. $\textsc{fo}$)

Procedure:

Input: φ

Output: is $\mathrm{L}(\varphi)$ simple?

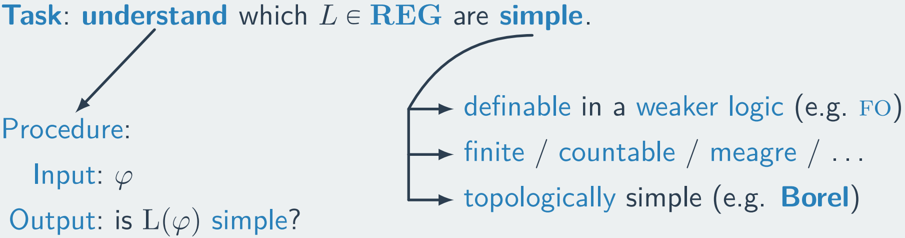
Task: **understand** which $L \in \mathbf{REG}$ are **simple**.

Procedure:

Input: φ

Output: is $\mathrm{L}(\varphi)$ simple?

- definable in a weaker logic (e.g. FO)
- finite / countable / meagre / . . .



Task: **understand** which $L \in \mathbf{REG}$ are **simple**.

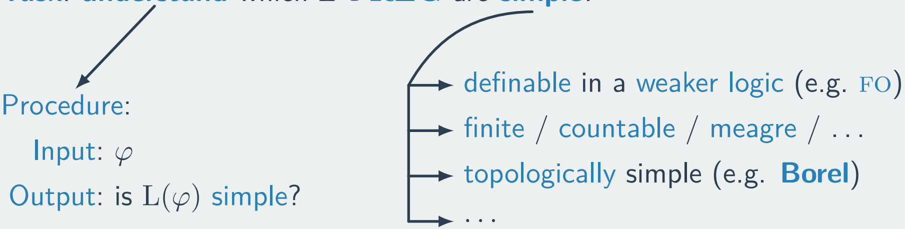
Procedure:

Input: φ

Output: is $\mathrm{L}(\varphi)$ simple?

- definable in a weaker logic (e.g. FO)
- finite / countable / meagre / ...
- topologically simple (e.g. **Borel**)

Task: **understand** which $L \in \mathbf{REG}$ are **simple**.



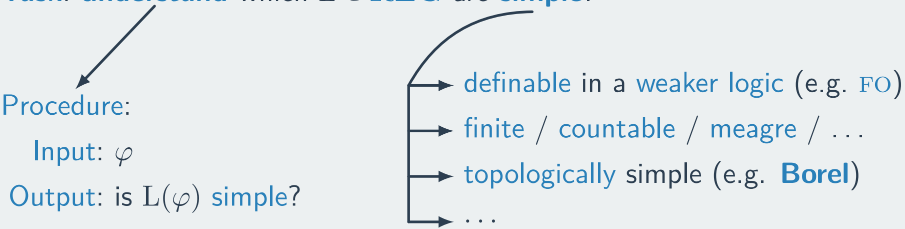
Procedure:

Input: φ

Output: is $\mathrm{L}(\varphi)$ 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: φ

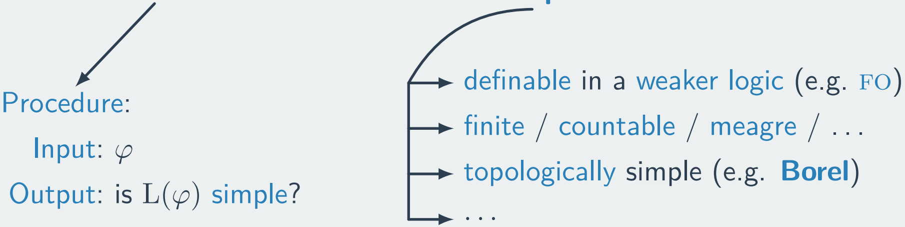
Output: is $\mathrm{L}(\varphi)$ simple?

- 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: φ

Output: is $\mathrm{L}(\varphi)$ 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: φ

Output: is $\mathrm{L}(\varphi)$ 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.

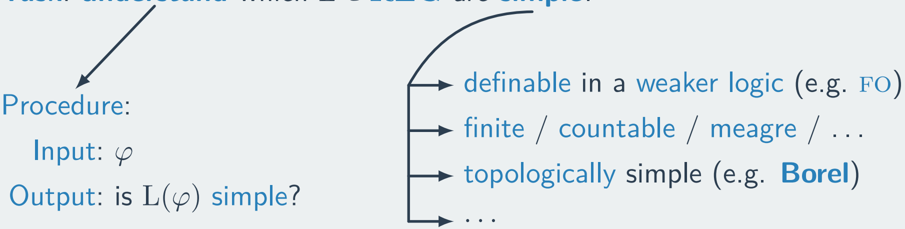
Theorem (Bojańczyk, Walukiewicz ['04])

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: φ

Output: is $\mathrm{L}(\varphi)$ 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, Facchini, Idziaszek, Kuperberg, Michalewski, Murlak, Niwiński, Place, Sreejith, Walukiewicz, . . .]

Pattern method for rigid representations

Pattern method for rigid representations

1. Input $L = \mathrm{L}(\varphi)$

Pattern method for **rigid representations**

1. Input $L = \mathrm{L}(\varphi)$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_0)$

Pattern method for **rigid representations**

1. Input $L = \mathrm{L}(\varphi)$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_0)$ $\left[\begin{array}{c}\text{Properties of } \mathcal{A}_0 \\ \text{are properties of } L\end{array}\right]$

Pattern method for **rigid representations**

1. Input $L = \mathrm{L}(\varphi)$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_0)$ $\left[\begin{array}{c}\text{Properties of } \mathcal{A}_0 \\ \text{are properties of } L\end{array}\right]$

$$\varphi \equiv \big(\varphi \wedge \Psi\big) \vee \big(\varphi \wedge \neg\Psi\big)$$

Pattern method for **rigid representations**

1. Input $L = \mathrm{L}(\varphi)$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_0)$ $\left[\begin{array}{c}\text{Properties of } \mathcal{A}_0 \\ \text{are properties of } L\end{array}\right]$

Pattern method for **rigid representations**

1. Input $L = \mathrm{L}(\varphi)$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_0)$ $\left[\begin{array}{c}\text{Properties of } \mathcal{A}_0 \\ \text{are properties of } L\end{array}\right]$

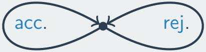
3. Search in $\mathcal{A}_0$ for a complicated **pattern**

Pattern method for rigid representations

1. Input $L = \mathrm{L}(\varphi)$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_0)$ [Properties of $\mathcal{A}_0$ are properties of L]

3. Search in $\mathcal{A}_0$ for a complicated **pattern**

[e.g. 

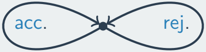
]

Pattern method for rigid representations

1. Input $L = \mathrm{L}(\varphi)$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_0)$ $\left[\begin{array}{c}\text{Properties of } \mathcal{A}_0 \\ \text{are properties of } L\end{array}\right]$

3. Search in $\mathcal{A}_0$ for a complicated **pattern**

[e.g. 

or $x^M \neq x^M \cdot x$]

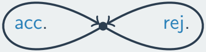
Pattern method for **rigid representations**

1. Input $L = \mathrm{L}(\varphi)$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_0)$ [Properties of $\mathcal{A}_0$ are properties of L]

3. Search in $\mathcal{A}_0$ for a complicated **pattern**

[e.g.



or $x^M \neq x^M \cdot x$]

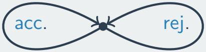
3.a Prove that L is **simple**

Pattern method for **rigid representations**

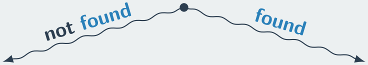
1. Input $L = \mathrm{L}(\varphi)$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_0)$ $\left[\begin{array}{c}\text{Properties of } \mathcal{A}_0 \\ \text{are properties of } L\end{array}\right]$

3. Search in $\mathcal{A}_0$ for a complicated **pattern**

[e.g. 

or $x^M \neq x^M \cdot x$]



3.a Prove that L is **simple**

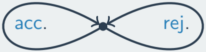
3.b Use it to show that L is hard

Pattern method for **rigid representations**

1. Input $L = \mathrm{L}(\varphi)$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_0)$ [Properties of $\mathcal{A}_0$ are properties of L]

3. Search in $\mathcal{A}_0$ for a complicated **pattern**

[e.g. 

or $x^M \neq x^M \cdot x$]

not found
found

3.a Prove that L is **simple**

3.b Use it to show that L is hard

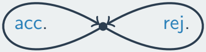
Limitations:

Pattern method for **rigid representations**

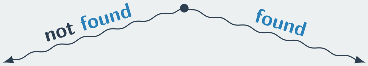
1. Input $L = \mathrm{L}(\varphi)$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_0)$ [Properties of $\mathcal{A}_0$ are properties of L]

3. Search in $\mathcal{A}_0$ for a complicated **pattern**

[e.g. 

or $x^M \neq x^M \cdot x$]



3.a Prove that L is **simple**

3.b Use it to show that L is hard

Limitations:

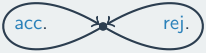
- **3.a** works under the assumption of lack of obstruction

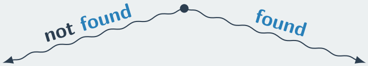
Pattern method for **rigid representations**

1. Input $L = \mathrm{L}(\varphi)$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_0)$ $\left[\begin{array}{c}\text{Properties of } \mathcal{A}_0 \\ \text{are properties of } L\end{array}\right]$

3. Search in $\mathcal{A}_0$ for a complicated **pattern**

[e.g.  or $x^M \neq x^M \cdot x$]



3.a Prove that L is **simple**

3.b Use it to show that L is hard

Limitations:

- **3.a** works under the assumption of lack of obstruction

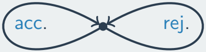
  $\rightsquigarrow$ difficult proofs

Pattern method for **rigid representations**

1. Input $L = \mathrm{L}(\varphi)$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_0)$ $\Big[$ Properties of $\mathcal{A}_0$ are properties of L $\Big]$

3. Search in $\mathcal{A}_0$ for a complicated **pattern**

$\Big[$e.g. 

or $x^M \neq x^M \cdot x$ $\Big]$

not found
found

3.a Prove that L is **simple**

3.b Use it to show that L is hard

Limitations:

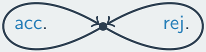
- **3.a** works under the assumption of lack of obstruction
  ⇝ difficult proofs
- **3.b** uses complexity in $\mathcal{A}_0$ to prove complexity of L

Pattern method for rigid representations

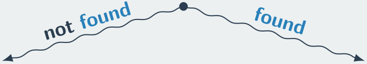
1. Input $L = \mathrm{L}(\varphi)$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_0)$ $\left[\begin{array}{c}\text{Properties of } \mathcal{A}_0 \\ \text{are properties of } L\end{array}\right]$

3. Search in $\mathcal{A}_0$ for a complicated **pattern**

[e.g. 

or $x^M \neq x^M \cdot x$]



3.a Prove that L is **simple**

3.b Use it to show that L is hard

Limitations:

- **3.a** works under the assumption of lack of obstruction

  ⇝ difficult proofs

- **3.b** uses complexity in $\mathcal{A}_0$ to prove complexity of L

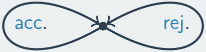
  ⇝ requires **rigid representations**

Pattern method for **rigid representations**

1. Input $L = \mathrm{L}(\varphi)$

2. Compute a **rigid representation** $L = \mathrm{L}(\mathcal{A}_0)$ [Properties of $\mathcal{A}_0$ are properties of L]

3. Search in $\mathcal{A}_0$ for a complicated **pattern**

[e.g. 

or $x^M \neq x^M \cdot x$]

not found
found

3.a Prove that L is **simple**

3.b Use it to show that L is hard

Limitations:

- **3.a** works under the assumption of lack of obstruction
  $\rightsquigarrow$ difficult proofs
- **3.b** uses complexity in $\mathcal{A}_0$ to prove complexity of L
  $\rightsquigarrow$ requires **rigid representations**

No such for infinite trees**!**

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 $\mathcal{G}_\varphi$

(I) **wins**

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 $\mathcal{G}_\varphi$

(I) **wins**

3.a Take his **w.s.** σ_{I}

Use σ_{I} to prove that L is simple

Game method

1. Input $L = \mathrm{L}(\varphi)$ **2.** Construct a game $\mathcal{G}_\varphi$ **3.** Solve $\mathcal{G}_\varphi$

(I) **wins**

(II) **wins**

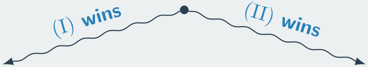
3.a Take his **w.s.** σ_{I}

3.b Take his **w.s.** σ_{II}

Use σ_{I} 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 simple

3.b Take his **w.s.** σ_{II}

Use σ_{II} to prove that L is hard

Game method

1. Input $L = \mathrm{L}(\varphi)$ **2.** Construct a game $\mathcal{G}_\varphi$ **3.** Solve $\mathcal{G}_\varphi$

(I) **wins**
(II) **wins**

3.a Take his **w.s.** σ_{I}

Use σ_{I} to prove that L is simple

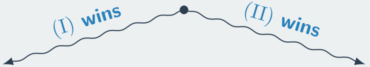
3.b Take his **w.s.** σ_{II}

Use σ_{II} to prove that L is hard

$\rightsquigarrow$ 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 $\mathcal{G}_\varphi$



(I) **wins**

(II) **wins**

3.a Take his **w.s.** σ_{I}

Use σ_{I} to prove that L is simple

3.b Take his **w.s.** σ_{II}

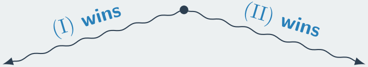
Use σ_{II} to prove that L is hard

$\rightsquigarrow$ 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**.

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 simple

3.b Take his **w.s.** σ_{II}

Use σ_{II} to prove that L is hard

$\rightsquigarrow$ 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 $\mathcal{G}_\varphi$

(I) **wins**

(II) **wins**

3.a Take his **w.s.** σ_{I}

Use σ_{I} to prove that L is simple

3.b Take his **w.s.** σ_{II}

Use σ_{II} to prove that L is hard

$\rightsquigarrow$ 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).

Game method

1. Input $L = \mathrm{L}(\varphi)$ **2.** Construct a game $\mathcal{G}_\varphi$ **3.** Solve $\mathcal{G}_\varphi$

(I) **wins**

(II) **wins**

3.a Take his **w.s.** σ_{I}

Use σ_{I} to prove that L is simple

3.b Take his **w.s.** σ_{II}

Use σ_{II} to prove that L is hard

⇝ 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 $\mathcal{G}_\varphi$

(I) **wins**

(II) **wins**

3.a Take his **w.s.** σ_{I}

Use σ_{I} to prove that L is simple

3.b Take his **w.s.** σ_{II}

Use σ_{II} to prove that L is hard

$\rightsquigarrow$ 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 $\mathcal{G}_\varphi$

(I) **wins**

(II) **wins**

3.a Take his **w.s.** σ_{I}

Use σ_{I} to prove that L is simple

3.b Take his **w.s.** σ_{II}

Use σ_{II} to prove that L is hard

$\rightsquigarrow$ 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

Theorem (S., Walukiewicz ['14])

It is decidable if a Büchi language of infinite trees is WMSO definable.

Theorem (S., Walukiewicz ['14])

It is decidable if a $\underbrace{\text{Büchi language of infinite trees}}_{\textbf{no}\text{ rigid representation}}$ is WMSO definable.

Theorem (S., Walukiewicz ['14])

It is decidable if a $\underbrace{\text{Büchi language of infinite trees}}_{\textbf{no}\ \text{rigid representation}}$ is $\underbrace{\textsc{wmso}}_{\text{weaker logic}}$ definable.

Theorem (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 (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

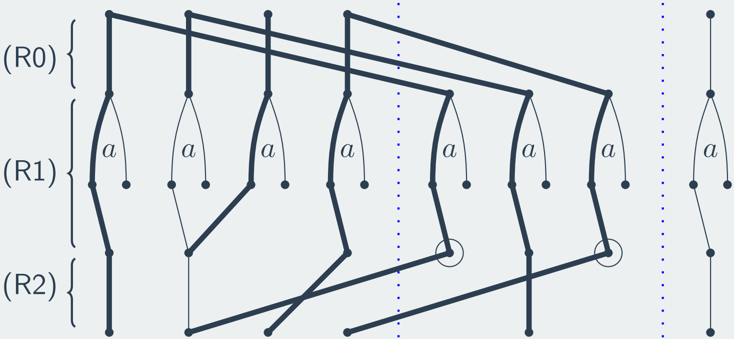
Take $L = \mathrm{L}(\mathcal{B})$ and construct a game $\mathcal{G}_{\mathcal{B}}$. $\big[W \equiv A \vee \big(B \wedge C\big)\big]$

Theorem (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 $\mathcal{G}_{\mathcal{B}}$. $\qquad \left[W \equiv A \vee \left(B \wedge C\right)\right]$



Theorem (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 (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 $\mathcal{G}_{\mathcal{B}}$. $\big[W \equiv A \vee \big(B \wedge C\big)\big]$

$\sigma_{\mathrm{I}} \rightsquigarrow$ a WMSO formula for L

Theorem (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 $\mathcal{G}_{\mathcal{B}}$. $\big[W \equiv A \vee \big(B \wedge C\big)\big]$

(I) **wins**

(II) **wins**

$\sigma_{\mathrm{I}} \rightsquigarrow$ a WMSO formula for L

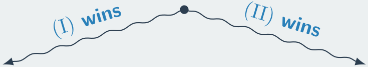
$\sigma_{\mathrm{II}} \rightsquigarrow$ L is **not** WMSO def.

Theorem (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 $\mathcal{G}_{\mathcal{B}}$. $\left[W \equiv A \vee \left(B \wedge C\right)\right]$



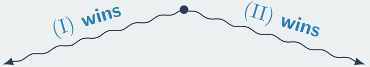
$\sigma_{\mathrm{I}} \rightsquigarrow$ a WMSO formula for L $\sigma_{\mathrm{II}} \rightsquigarrow$ L is **not** WMSO def. ■

Theorem (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 $\mathcal{G}_{\mathcal{B}}$. $\big[W \equiv A \vee \big(B \wedge C\big)\big]$



$\sigma_{\mathrm{I}} \rightsquigarrow$ a WMSO formula for L $\sigma_{\mathrm{II}} \rightsquigarrow$ L is **not** WMSO def. ■

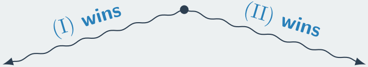
But it seemed that we can get more (ordinal ranks)!

Theorem (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 $\mathcal{G}_{\mathcal{B}}$. $\big[W \equiv A \vee \big(B \wedge C\big)\big]$



$\sigma_{\mathrm{I}} \rightsquigarrow$ a WMSO formula for L

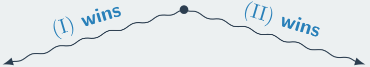
$\sigma_{\mathrm{II}} \rightsquigarrow L$ is **not** WMSO def. ■

Theorem (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 $\mathcal{G}_{\mathcal{B}}$. $\big[W \equiv A \vee \big(B \wedge C\big)\big]$



(I) **wins**

(II) **wins**

$\sigma_{\mathrm{I}} \rightsquigarrow$ a WMSO formula for L

$\sigma_{\mathrm{II}} \rightsquigarrow$ L is **not** WMSO def. ■

Theorem (S., Walukiewicz ['16])

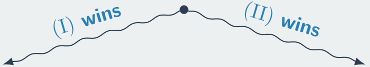
A Büchi language is WMSO def. **iff** it is **Borel**; and it is decidable.

Theorem (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 $\mathcal{G}_{\mathcal{B}}$. $\big[W \equiv A \lor \big(B \land C\big)\big]$



$\sigma_{\mathrm{I}} \rightsquigarrow$ a WMSO formula for L $\sigma_{\mathrm{II}} \rightsquigarrow$ L is **not** WMSO def. ■

Theorem (S., Walukiewicz ['16])

A Büchi language is WMSO def. **iff** it is **Borel**; and it is decidable.

Proof

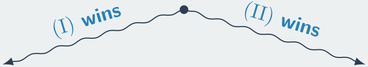
Take $L = \mathrm{L}(\mathcal{B})$ and construct a game $\mathcal{G}'_{\mathcal{B}}$.

Theorem (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 $\mathcal{G}_{\mathcal{B}}$. $\big[W \equiv A \vee \big(B \wedge C\big)\big]$



$\sigma_{\mathrm{I}} \rightsquigarrow$ a WMSO formula for L

$\sigma_{\mathrm{II}} \rightsquigarrow L$ is **not** WMSO def. ■

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 $\mathcal{G}'_{\mathcal{B}}$. $\big[W \equiv \big(A \vee B\big) \wedge C'\big]$

Theorem (S., Walukiewicz ['14])

It is decidable if a $\underbrace{\text{Büchi language of infinite trees}}_{\text{no rigid representation}}$ is $\underbrace{\text{WMSO definable}}_{\text{weaker logic}}$.

Proof

Take $L = \mathrm{L}(\mathcal{B})$ and construct a game $\mathcal{G}_{\mathcal{B}}$. $\big[W \equiv A \vee \big(B \wedge C\big)\big]$

(I) wins
(II) wins

$\sigma_{\mathrm{I}} \rightsquigarrow$ a WMSO formula for L $\sigma_{\mathrm{II}} \rightsquigarrow$ L is **not** WMSO def. ■

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 $\mathcal{G}'_{\mathcal{B}}$. $\big[W \equiv \big(A \vee B\big) \wedge C'\big]$

$\sigma_{\mathrm{I}} \rightsquigarrow$ a WMSO formula for L

$\rightsquigarrow$ L is **Borel**

Theorem (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 $\mathcal{G}_{\mathcal{B}}$. $\big[W \equiv A \vee \big(B \wedge C\big)\big]$

(I) wins
(II) wins

$\sigma_{\mathrm{I}} \rightsquigarrow$ a WMSO formula for L $\sigma_{\mathrm{II}} \rightsquigarrow$ L is **not** WMSO def. ■

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 $\mathcal{G}'_{\mathcal{B}}$. $\big[W \equiv \big(A \vee B\big) \wedge C'\big]$

(I) wins
(II) wins

$\sigma_{\mathrm{I}} \rightsquigarrow$ a WMSO formula for L $\sigma_{\mathrm{II}} \rightsquigarrow$ L is **not Borel**

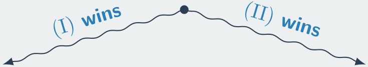
$\rightsquigarrow$ L is **Borel** $\rightsquigarrow$ L is **not** WMSO def.

Theorem (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 $\mathcal{G}_{\mathcal{B}}$. $\quad \big[W \equiv A \vee \big(B \wedge C\big)\big]$



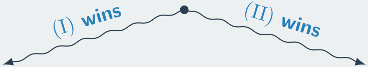
$\sigma_{\mathrm{I}} \rightsquigarrow$ a WMSO formula for L $\qquad$ $\sigma_{\mathrm{II}} \rightsquigarrow$ L is **not** WMSO def. ■

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 $\mathcal{G}'_{\mathcal{B}}$. $\quad \big[W \equiv \big(A \vee B\big) \wedge C'\big]$



$\sigma_{\mathrm{I}} \rightsquigarrow$ a WMSO formula for L $\qquad$ $\sigma_{\mathrm{II}} \rightsquigarrow$ L is **not** **Borel**

$\rightsquigarrow$ L is **Borel** $\qquad$ $\rightsquigarrow$ L is **not** WMSO def. ■

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

Let L be regular lang. of inf. trees. Then effectively either:

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

Let L be regular lang. of inf. trees. Then effectively either:

1. L is $\quad\mathrm{weak{-}alt}(0,2)$-definable and $L \in \mathbf{\Pi}^0_2$

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

Let L be regular lang. of inf. trees. Then effectively either:

1. L is $\mathrm{weak{-}alt}(0,2)$-definable and $L \in \mathbf{\Pi}^0_2$

2. L isn't $\mathrm{weak{-}alt}(0,2)$-definable and $L \notin \mathbf{\Pi}^0_2$

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

Let L be regular lang. of inf. trees. Then effectively either:

1. L is $\mathrm{weak{-}alt}(0,2)$-definable and $L \in \mathbf{\Pi}^0_2$

2. L isn't $\underbrace{\mathrm{weak{-}alt}(0,2)}_{\text{weak index}}$-definable and $L \notin \mathbf{\Pi}^0_2$

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

Let L be regular lang. of inf. trees. Then effectively either:

1. L is $\mathrm{weak}{-}\mathrm{alt}(0,2)$-definable and $L \in \mathbf{\Pi}^0_2$

2. L isn't $\underbrace{\mathrm{weak}{-}\mathrm{alt}(0,2)}_{\text{weak index}}$-definable and $\underbrace{L \notin \mathbf{\Pi}^0_2}_{\text{topological complexity}}$

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

Let L be regular lang. of inf. trees. Then effectively either:

1. L is $\mathrm{weak}{-}\mathrm{alt}(0,2)$-definable and $L \in \mathbf{\Pi}^0_2$

2. L isn't $\underbrace{\mathrm{weak}{-}\mathrm{alt}(0,2)}_{\text{weak index}}$-definable and $\underbrace{L \notin \mathbf{\Pi}^0_2}_{\text{topological complexity}}$

Proof

Consider a game $\mathcal{F}$

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

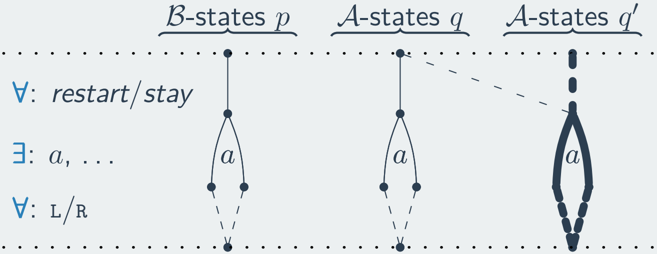
Let L be regular lang. of inf. trees. Then effectively either:

1. L is $\mathrm{weak{-}alt}(0,2)$-definable and $L \in \mathbf{\Pi}^0_2$

2. L isn't $\underbrace{\mathrm{weak{-}alt}(0,2)}_{\text{weak index}}$-definable and $\underbrace{L \notin \mathbf{\Pi}^0_2}_{\text{topological complexity}}$

Proof

Consider a game $\mathcal{F}$



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

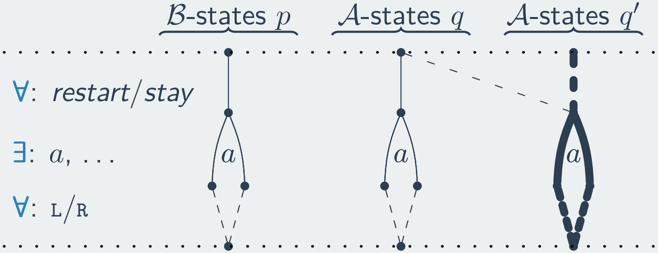
Let L be regular lang. of inf. trees. Then effectively either:

1. L is $\text{weak-alt}(0,2)$-definable and $L \in \mathbf{\Pi}^0_2$

2. L isn't $\underbrace{\text{weak-alt}(0,2)}_{\text{weak index}}$-definable and $\underbrace{L \notin \mathbf{\Pi}^0_2}_{\text{topological complexity}}$

Proof

Consider a game $\mathcal{F}$



$\mathcal{B}$-states p

$\mathcal{A}$-states q

$\mathcal{A}$-states q'

$\forall$: *restart/stay*

$\exists$: a, . . .

$\forall$: L/R

$$W \equiv \big((\mathrm{WR}) \wedge (\mathrm{WB})\big) \vee \big(\neg(\mathrm{WR}) \wedge (\mathrm{WA})\big)$$

Two lemmata:

Two lemmata:

1. If (I) wins $\mathcal{F}$ then L is $\mathrm{weak}{-}\mathrm{alt}(0,2)$-definable

Two lemmata:

1. If (I) wins $\mathcal{F}$ then L is $\mathrm{weak{-}alt}(0,2)$-definable

Proof

Take a finite memory strategy of (I) in $\mathcal{F}$

Two lemmata:

1. If $\mathrm{(I)}$ wins $\mathcal{F}$ then L is $\mathrm{weak{-}alt}(0,2)$-definable

Proof

Take a finite memory strategy of $\mathrm{(I)}$ in $\mathcal{F}$

Add some pumping

Two lemmata:

1. If (I) wins $\mathcal{F}$ then L is $\mathrm{weak{-}alt}(0,2)$-definable

Proof

Take a finite memory strategy of (I) 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 $\mathrm{weak{-}alt}(0,2)$-definable

Proof

Take a finite memory strategy of (I) in $\mathcal{F}$

Add some pumping

$\rightsquigarrow$ a weak alternating $(0,2)$ automaton for L ■

$\rightsquigarrow$ $L \in \mathbf{\Pi}^0_2$

Two lemmata:

1. If (I) wins $\mathcal{F}$ then L is $\mathrm{weak{-}alt}(0,2)$-definable

Proof

Take a finite memory strategy of (I) in $\mathcal{F}$

Add some pumping

$\rightsquigarrow$ a weak alternating $(0,2)$ automaton for L ■

$\rightsquigarrow L \in \mathbf{\Pi}^0_2$

2. If (II) wins $\mathcal{F}$ then L is **not** $\mathbf{\Pi}^0_2$

Two lemmata:

1. If (I) wins $\mathcal{F}$ then L is $\mathrm{weak{-}alt}(0,2)$-definable

Proof

Take a finite memory strategy of (I) in $\mathcal{F}$

Add some pumping

$\rightsquigarrow$ a weak alternating $(0,2)$ automaton for L ■

$\rightsquigarrow$ $L \in \mathbf{\Pi}^0_2$

2. If (II) wins $\mathcal{F}$ then L is **not** $\mathbf{\Pi}^0_2$

Proof

Take a strategy of (II) in $\mathcal{F}$

Two lemmata:

1. If (I) wins $\mathcal{F}$ then L is $\mathrm{weak}{-}\mathrm{alt}(0,2)$-definable

Proof

Take a finite memory strategy of (I) in $\mathcal{F}$

Add some pumping

$\rightsquigarrow$ a weak alternating $(0,2)$ automaton for L ■

$\rightsquigarrow$ $L \in \mathbf{\Pi}^0_2$

2. If (II) wins $\mathcal{F}$ then L is **not** $\mathbf{\Pi}^0_2$

Proof

Take a strategy of (II) in $\mathcal{F}$

Confront it with a family of quasi-strategies of (I)

Two lemmata:

1. If (I) wins $\mathcal{F}$ then L is $\mathrm{weak{-}alt}(0, 2)$-definable

Proof

Take a finite memory strategy of (I) in $\mathcal{F}$

Add some pumping

$\leadsto$ a weak alternating $(0, 2)$ automaton for L ■

$\leadsto L \in \mathbf{\Pi}^0_2$

2. If (II) wins $\mathcal{F}$ then L is **not** $\mathbf{\Pi}^0_2$

Proof

Take a strategy of (II) in $\mathcal{F}$

Confront it with a family of quasi-strategies of (I)

$\leadsto$ a reduction proving that $L \notin \mathbf{\Pi}^0_2$ ■

Two lemmata:

1. If (I) wins $\mathcal{F}$ then L is $\mathrm{weak{-}alt}(0,2)$-definable

Proof

Take a finite memory strategy of (I) in $\mathcal{F}$

Add some pumping

$\rightsquigarrow$ a weak alternating $(0,2)$ automaton for L ■

$\rightsquigarrow$ $L \in \mathbf{\Pi}^0_2$

2. If (II) wins $\mathcal{F}$ then L is **not** $\mathbf{\Pi}^0_2$

Proof

Take a strategy of (II) in $\mathcal{F}$

Confront it with a family of quasi-strategies of (I)

$\rightsquigarrow$ a reduction proving that $L \notin \mathbf{\Pi}^0_2$ ■

$\rightsquigarrow$ L is **not** $\mathrm{weak{-}alt}(0,2)$-definable

Two lemmata:

1. If (I) wins $\mathcal{F}$ then L is $\mathrm{weak{-}alt}(0,2)$-definable

Proof

Take a finite memory strategy of (I) in $\mathcal{F}$

Add some pumping

$\rightsquigarrow$ a weak alternating $(0,2)$ automaton for L $\blacksquare$

$\rightsquigarrow$ $L \in \mathbf{\Pi}^0_2$

2. If (II) wins $\mathcal{F}$ then L is **not** $\mathbf{\Pi}^0_2$

Proof

Take a strategy of (II) in $\mathcal{F}$

Confront it with a family of quasi-strategies of (I)

$\rightsquigarrow$ a reduction proving that $L \notin \mathbf{\Pi}^0_2$ $\blacksquare$

$\rightsquigarrow$ L is **not** $\mathrm{weak{-}alt}(0,2)$-definable

A complete proof

Two lemmata:

1. If (I) wins $\mathcal{F}$ then L is $\mathrm{weak{-}alt}(0,2)$-definable

Proof

Take a finite memory strategy of (I) in $\mathcal{F}$

Add some pumping

$\rightsquigarrow$ a weak alternating $(0,2)$ automaton for L ■

$\rightsquigarrow$ $L \in \mathbf{\Pi}^0_2$

2. If (II) wins $\mathcal{F}$ then L is **not** $\mathbf{\Pi}^0_2$

Proof

Take a strategy of (II) in $\mathcal{F}$

Confront it with a family of quasi-strategies of (I)

$\rightsquigarrow$ a reduction proving that $L \notin \mathbf{\Pi}^0_2$ ■

$\rightsquigarrow$ L is **not** $\mathrm{weak{-}alt}(0,2)$-definable

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 $\mathrm{weak{-}alt}(0,2)$-definable

Proof

Take a finite memory strategy of (I) in $\mathcal{F}$

Add some pumping

$\rightsquigarrow$ a weak alternating $(0,2)$ automaton for L ■

$\rightsquigarrow$ $L \in \mathbf{\Pi}^0_2$

2. If (II) wins $\mathcal{F}$ then L is **not** $\mathbf{\Pi}^0_2$

Proof

Take a strategy of (II) in $\mathcal{F}$

Confront it with a family of quasi-strategies of (I)

$\rightsquigarrow$ a reduction proving that $L \notin \mathbf{\Pi}^0_2$ ■

$\rightsquigarrow$ L is **not** $\mathrm{weak{-}alt}(0,2)$-definable

A complete proof **not** using properties on which the game $\mathcal{F}$ is based

[dealternation]

Summary

Summary

→ characterising properties of sets

Summary

⟶ characterising properties of sets

⟶ games in general + **determinacy**

Summary

⟶ characterising properties of sets

⟶ games in general + **determinacy**

⟶ effectiveness for regular winning conditions

Summary

- ⟶ characterising properties of sets
- ⟶ games in general + **determinacy**
- ⟶ effectiveness for regular winning conditions
- ⟶ pattern method (rigid representatons: determinism / algebra)

Summary

→ characterising properties of sets

→ games in general + **determinacy**

→ effectiveness for regular winning conditions

→ pattern method (rigid representatons: determinism / algebra)

pattern missing
$\rightsquigarrow$ L is simple

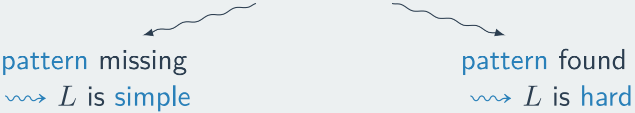
Summary

⟶ characterising properties of sets

⟶ games in general + **determinacy**

⟶ effectiveness for regular winning conditions

⟶ pattern method (rigid representatons: determinism / algebra)

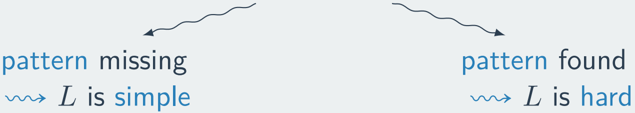


pattern missing
$\rightsquigarrow L$ is simple

pattern found
$\rightsquigarrow L$ is hard

Summary

- $\longrightarrow$ characterising properties of sets
- $\longrightarrow$ games in general + **determinacy**
- $\longrightarrow$ effectiveness for regular winning conditions
- $\longrightarrow$ pattern method (rigid representatons: determinism / algebra)



pattern missing
$\rightsquigarrow L$ is simple

pattern found
$\rightsquigarrow L$ is hard

- $\longrightarrow$ games (may deal with non-determinism)

Summary

- → characterising properties of sets
- → games in general + **determinacy**
- → effectiveness for regular winning conditions
- → pattern method (rigid representatons: determinism / algebra)

pattern missing
$\rightsquigarrow$ L is simple

pattern found
$\rightsquigarrow$ L is hard

- → games (may deal with non-determinism)

strategy of (I)
$\rightsquigarrow$ L is simple

Summary

- $\longrightarrow$ characterising properties of sets
- $\longrightarrow$ games in general + **determinacy**
- $\longrightarrow$ effectiveness for regular winning conditions
- $\longrightarrow$ pattern method (rigid representatons: determinism / algebra)

pattern missing
$\rightsquigarrow L$ is simple

pattern found
$\rightsquigarrow L$ is hard

- $\longrightarrow$ games (may deal with non-determinism)

strategy of (I)
$\rightsquigarrow L$ is simple

strategy of (II)
$\rightsquigarrow L$ is hard

Summary

⟶ characterising properties of sets

⟶ games in general + **determinacy**

⟶ effectiveness for regular winning conditions

⟶ pattern method (rigid representatons: determinism / algebra)

pattern missing
$\rightsquigarrow$ L is simple

pattern found
$\rightsquigarrow$ L is hard

⟶ games (may deal with non-determinism)

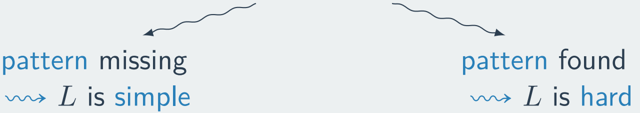
strategy of (I)
$\rightsquigarrow$ L is simple

strategy of (II)
$\rightsquigarrow$ L is hard

⟶ **no** general recipe for design

Summary

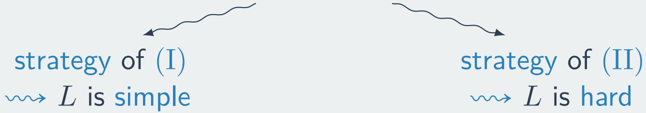
- → characterising properties of sets
- → games in general + **determinacy**
- → effectiveness for regular winning conditions
- → pattern method (rigid representatons: determinism / algebra)



pattern missing
⇝ L is simple

pattern found
⇝ L is hard

- → games (may deal with non-determinism)



strategy of (I)
⇝ L is simple

strategy of (II)
⇝ L is hard

- → **no** general recipe for design

Conjecture: Every class of languages has a game characterisation.