Infinite computations, logic, and topological complexity

Michał Skrzypczak

Workshop on Topology and Languages
Toulouse

Structures

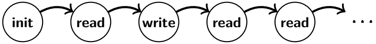
Structures

finite / **infinite**:

Structures

finite / **infinite**:

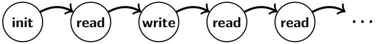
words



Structures

finite / **infinite**:

words

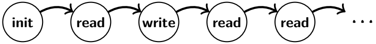


$\alpha\colon \omega \to A = \{\mathrm{init}, \mathrm{read}, \mathrm{write}\}$

Structures

finite / **infinite**:

words



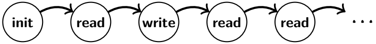
$$\alpha\colon \omega \to A = \{\text{init}, \text{read}, \text{write}\}$$

$$\alpha \in A^\omega$$

Structures

finite / **infinite**:

words



$$\alpha\colon \omega \to A = \{\mathrm{init}, \mathrm{read}, \mathrm{write}\}$$
$$\alpha \in A^\omega$$

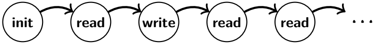
e.g.:

— trace of execution of system

Structures

finite / **infinite**:

words



$$\alpha \colon \omega \to A = \{\text{init}, \text{read}, \text{write}\}$$

$$\alpha \in A^\omega$$

e.g.:

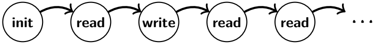
— trace of execution of system

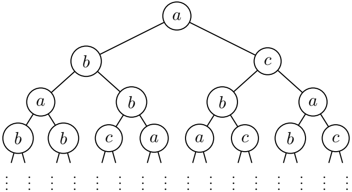
— log of communication

Structures

finite / **infinite**:

words and **trees**





$$\alpha\colon \omega \to A = \{\text{init}, \text{read}, \text{write}\}$$

$$\alpha \in A^\omega$$

e.g.:

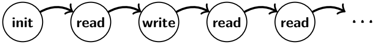
— trace of execution of system

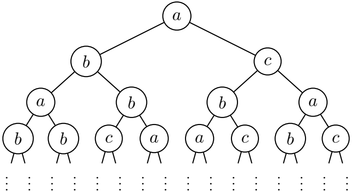
— log of communication

Structures

finite / **infinite**:

words and **trees**





$\alpha\colon \omega \to A = \{\text{init}, \text{read}, \text{write}\}$

$\alpha \in A^\omega$

$t\colon \{\texttt{L}, \texttt{R}\}^* \to A = \{a, b, c, d\}$

e.g.:

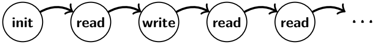
— trace of execution of system

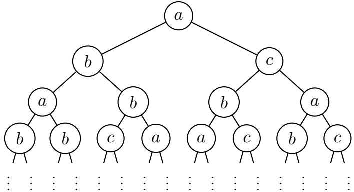
— log of communication

Structures

finite / **infinite**:

words and **trees**





$$\alpha\colon \omega \to A = \{\text{init}, \text{read}, \text{write}\}$$
$$\alpha \in A^\omega$$

$$t\colon \{\mathtt{L}, \mathtt{R}\}^* \to A = \{a, b, c, d\}$$
$$t \in A^{(\{\mathtt{L},\mathtt{R}\}^*)}$$

e.g.:

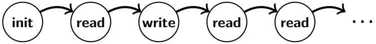
— trace of execution of system

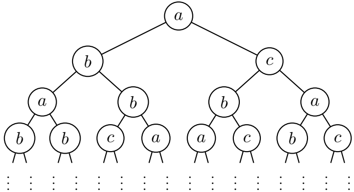
— log of communication

Structures

finite / **infinite**:

words and **trees**





$\alpha\colon \omega \to A = \{\text{init}, \text{read}, \text{write}\}$

$\alpha \in A^\omega$

$t\colon \{\mathtt{L}, \mathtt{R}\}^* \to A = \{a, b, c, d\}$

$t \in A^{(\{\mathtt{L},\mathtt{R}\}^*)}$

e.g.:

— trace of execution of system

— log of communication

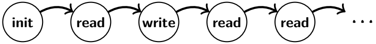
e.g.:

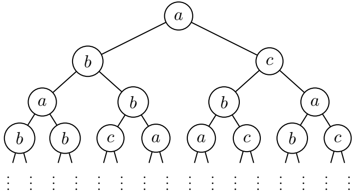
— all possible executions

Structures

finite / **infinite**:

words and **trees**





$\alpha\colon \omega \to A = \{\text{init}, \text{read}, \text{write}\}$

$\alpha \in A^\omega$

$t\colon \{\mathtt{L}, \mathtt{R}\}^* \to A = \{a, b, c, d\}$

$t \in A^{(\{\mathtt{L},\mathtt{R}\}^*)}$

e.g.:

— trace of execution of system

— log of communication

e.g.:

— all possible executions

— strategy against environment

Logics

Logics

Monadic Second-Order (MSO) logic:

Logics

Monadic Second-Order (MSO) logic:

- $\exists_x$, $\forall_x$ $\quad x$ — node

Logics

Monadic Second-Order (MSO) logic:

- $\exists_x$, $\forall_x$ x — node
- $\exists^{\mathsf{fin}}_X$, $\forall^{\mathsf{fin}}_X$ X — finite set of nodes

Logics

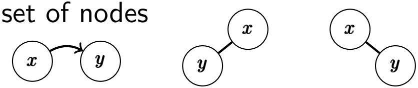
Monadic Second-Order (MSO) logic:

- $\exists_x$, $\forall_x$ x — node
- $\exists^{\mathsf{fin}}_X$, $\forall^{\mathsf{fin}}_X$ X — finite set of nodes
- $\exists_X$, $\forall_X$ X — any set of nodes

Logics

Monadic Second-Order (MSO) logic:

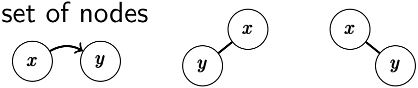
- $\exists_x$, $\forall_x$ $\quad x$ — node
- $\exists^{\mathsf{fin}}_X$, $\forall^{\mathsf{fin}}_X$ $\quad X$ — finite set of nodes
- $\exists_X$, $\forall_X$ $\quad X$ — any set of nodes
- successor predicates:



Logics

Monadic Second-Order (MSO) logic:

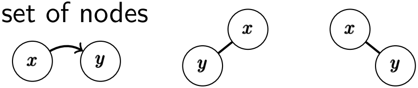
- $\exists_x$, $\forall_x$ $\quad x$ — node
- $\exists_X^{\text{fin}}$, $\forall_X^{\text{fin}}$ $\quad X$ — finite set of nodes
- $\exists_X$, $\forall_X$ $\quad X$ — any set of nodes
- successor predicates:



- label predicates: $a(x)$, $b(x)$, ...

Logics

Monadic Second-Order (MSO) logic:

- $\exists_x$, $\forall_x$ $\quad$ x — node
- $\exists^{\mathsf{fin}}_X$, $\forall^{\mathsf{fin}}_X$ $\quad$ X — finite set of nodes
- $\exists_X$, $\forall_X$ $\quad$ X — any set of nodes
- successor predicates: 

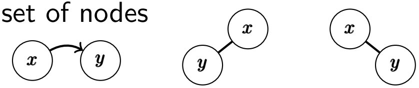
- label predicates: $a(x)$, $b(x)$, $\ldots$
- $x \in X$, $x = y$, $x \leqslant y$, $x \leqslant_{\mathrm{lex}} y$, $x \preceq y$

Logics

Monadic Second-Order (MSO) logic:

no arithmetic **!!!**

- $\exists_x$, $\forall_x$ $\quad x$ — node
- $\exists^{\text{fin}}_X$, $\forall^{\text{fin}}_X$ $\quad X$ — finite set of nodes
- $\exists_X$, $\forall_X$ $\quad X$ — any set of nodes
- successor predicates:

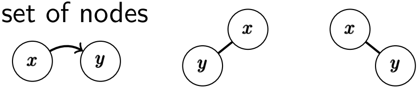


- label predicates: $a(x)$, $b(x)$, ...
- $x \in X$, $x = y$, $x \leqslant y$, $x \leqslant_{\text{lex}} y$, $x \preceq y$

Logics

Monadic Second-Order (MSO) logic:

- $\exists_x$, $\forall_x$ $\quad x$ — node
- $\exists^{\mathsf{fin}}_X$, $\forall^{\mathsf{fin}}_X$ $\quad X$ — finite set of nodes
- $\exists_X$, $\forall_X$ $\quad X$ — any set of nodes
- successor predicates:

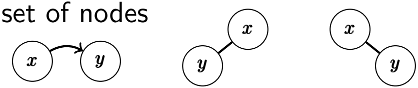


- label predicates: $a(x)$, $b(x)$, $\ldots$
- $x \in X$, $x = y$, $x \leqslant y$, $x \leqslant_{\mathrm{lex}} y$, $x \preceq y$

Logics

Monadic Second-Order (MSO) logic:

- $\exists_x$, $\forall_x$ $\quad x$ — node
- $\exists^{\mathsf{fin}}_X$, $\forall^{\mathsf{fin}}_X$ $\quad X$ — finite set of nodes
- $\exists_X$, $\forall_X$ $\quad X$ — any set of nodes
- successor predicates:



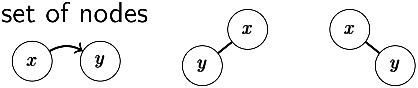
- label predicates: $a(x)$, $b(x)$, ...
- $x \in X$, $x = y$, $x \leqslant y$, $x \leqslant_{\mathrm{lex}} y$, $x \preceq y$

Weak Monadic Second-Order (WMSO) logic:

Logics

Monadic Second-Order (MSO) logic:

- $\exists_x$, $\forall_x$ x — node
- $\exists^{\text{fin}}_X$, $\forall^{\text{fin}}_X$ X — finite set of nodes
- $\exists_X$, $\forall_X$ X — any set of nodes
- successor predicates:



- label predicates: $a(x)$, $b(x)$, $\ldots$
- $x \in X$, $x = y$, $x \leqslant y$, $x \leqslant_{\text{lex}} y$, $x \preceq y$

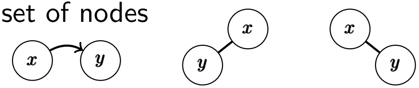
Weak Monadic Second-Order (WMSO) logic:

- only first-order $\exists_x$, $\forall_x$ and finite $\exists^{\text{fin}}_X$, $\forall^{\text{fin}}_X$ quantifiers

Logics

Monadic Second-Order (MSO) logic:

- $\exists_x$, $\forall_x$ $\quad x$ — node
- $\exists^{\text{fin}}_X$, $\forall^{\text{fin}}_X$ $\quad X$ — finite set of nodes
- $\exists_X$, $\forall_X$ $\quad X$ — any set of nodes
- successor predicates:



- label predicates: $a(x)$, $b(x)$, ...
- $x \in X$, $x = y$, $x \leqslant y$, $x \leqslant_{\text{lex}} y$, $x \preceq y$

Weak Monadic Second-Order (WMSO) logic:

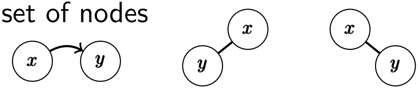
- only first-order $\exists_x$, $\forall_x$ and finite $\exists^{\text{fin}}_X$, $\forall^{\text{fin}}_X$ quantifiers

First-Order (FO) logic:

Logics

Monadic Second-Order (MSO) logic:

- $\exists_x$, $\forall_x$ $\quad x$ — node
- $\exists^{\text{fin}}_X$, $\forall^{\text{fin}}_X$ $\quad X$ — finite set of nodes
- $\exists_X$, $\forall_X$ $\quad X$ — any set of nodes
- successor predicates:



- label predicates: $a(x)$, $b(x)$, $\ldots$
- $x \in X$, $x = y$, $x \leqslant y$, $x \leqslant_{\text{lex}} y$, $x \preceq y$

Weak Monadic Second-Order (WMSO) logic:

- only first-order $\exists_x$, $\forall_x$ and finite $\exists^{\text{fin}}_X$, $\forall^{\text{fin}}_X$ quantifiers

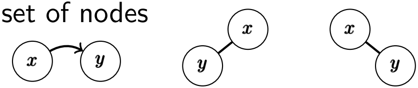
First-Order (FO) logic:

- only first-order $\exists_x$, $\forall_x$ quantifiers

Logics

Monadic Second-Order (MSO) logic:

- $\exists_x$, $\forall_x$ $\quad x$ — node
- $\exists^{\mathsf{fin}}_X$, $\forall^{\mathsf{fin}}_X$ $\quad X$ — finite set of nodes
- $\exists_X$, $\forall_X$ $\quad X$ — any set of nodes
- successor predicates:



- label predicates: $a(x)$, $b(x)$, ...
- $x \in X$, $x = y$, $x \leqslant y$, $x \leqslant_{\mathrm{lex}} y$, $x \preceq y$

Weak Monadic Second-Order (WMSO) logic:

- only first-order $\exists_x$, $\forall_x$ and finite $\exists^{\mathsf{fin}}_X$, $\forall^{\mathsf{fin}}_X$ quantifiers

First-Order (FO) logic:

- only first-order $\exists_x$, $\forall_x$ quantifiers

Other formalisms: LTL, CTL*, modal μ-calculus, ...

Examples

Examples

Infinite words

Examples

Infinite words

— safety: $\forall x.\ \neg a(x)$

Examples

Infinite words

— safety: $\forall x.\ \neg a(x)$

— liveness: $\forall x.\ \exists y.\quad x < y\ \wedge\ b(y)$

Examples

Infinite words

— safety: $\forall x.\ \neg a(x)$

— liveness: $\forall x.\ \exists y.\quad x < y\ \wedge\ b(y)$

Infinite trees

Examples

Infinite words

— safety: $\forall x.\ \neg a(x)$

— liveness: $\forall x.\ \exists y.\quad x < y \ \wedge\ b(y)$

Infinite trees

— two incomparable b:

$$\exists x.\ \exists y.\ b(x) \wedge b(y) \wedge \neg\big(x \preceq y \vee y \preceq x\big)$$

Examples

Infinite words

— safety: $\forall x.\ \neg a(x)$

— liveness: $\forall x.\ \exists y.\quad x < y\ \wedge\ b(y)$

Infinite trees

— two incomparable b:

$$\exists x.\ \exists y.\ b(x) \wedge b(y) \wedge \neg\big(x \preceq y \vee y \preceq x\big)$$

— infinite chain of a:

$$\exists X.\ \big(\exists x.\ x \in X\big) \wedge$$

Examples

Infinite words

— safety: $\forall x.\ \neg a(x)$

— liveness: $\forall x.\ \exists y.\quad x < y\ \wedge\ b(y)$

Infinite trees

— two incomparable b:

$$\exists x.\ \exists y.\ b(x) \wedge b(y) \wedge \neg\big(x \preceq y \vee y \preceq x\big)$$

— infinite chain of a:

$$\exists X.\ \big(\exists x.\ x \in X\big) \wedge$$
$$\big(\forall x.\ x \in X \Rightarrow a(x)\big) \wedge$$

Examples

Infinite words

— safety: $\forall x.\ \neg a(x)$

— liveness: $\forall x.\ \exists y.\quad x < y \ \wedge\ b(y)$

Infinite trees

— two incomparable b:

$$\exists x.\ \exists y.\ b(x) \wedge b(y) \wedge \neg\big(x \preceq y \vee y \preceq x\big)$$

— infinite chain of a:

$$\begin{aligned}\exists X.\ &\big(\exists x.\ x \in X\big) \wedge \\ &\big(\forall x.\ x \in X \Rightarrow a(x)\big) \wedge \\ &\forall x.\ x \in X \Rightarrow\ \exists y.\ x \prec y \ \wedge\ y \in X\end{aligned}$$

Examples

Infinite words

— safety: $\forall x.\ \neg a(x)$

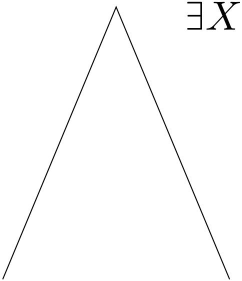
— liveness: $\forall x.\ \exists y.\quad x < y\ \wedge\ b(y)$

Infinite trees

— two incomparable b:

$$\exists x.\ \exists y.\ b(x) \wedge b(y) \wedge \neg\big(x \preceq y \vee y \preceq x\big)$$

— infinite chain of a:



$$\begin{aligned}\exists X.\ &\big(\exists x.\ x \in X\big) \wedge \\ &\big(\forall x.\ x \in X \Rightarrow a(x)\big) \wedge \\ &\forall x.\ x \in X \Rightarrow\ \exists y.\ x \prec y\ \wedge\ y \in X\end{aligned}$$

Examples

Infinite words

— safety: $\forall x.\ \neg a(x)$

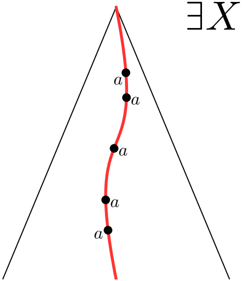
— liveness: $\forall x.\ \exists y.\quad x < y\ \wedge\ b(y)$

Infinite trees

— two incomparable b:

$$\exists x.\ \exists y.\ b(x) \wedge b(y) \wedge \neg\big(x \preceq y \vee y \preceq x\big)$$

— infinite chain of a:



$$\exists X.\ \big(\exists x.\ x \in X\big) \wedge$$
$$\big(\forall x.\ x \in X \Rightarrow a(x)\big) \wedge$$
$$\forall x.\ x \in X \Rightarrow\ \exists y.\ x \prec y\ \wedge\ y \in X$$

Examples

Infinite words

— safety: $\forall x.\ \neg a(x)$

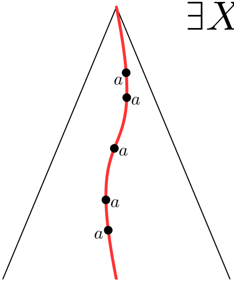
— liveness: $\forall x.\ \exists y.\quad x < y\ \wedge\ b(y)$

Infinite trees

— two incomparable b:

$$\exists x.\ \exists y.\ b(x) \wedge b(y) \wedge \neg\big(x \preceq y \vee y \preceq x\big)$$

— infinite chain of a:



$$\begin{aligned}\exists X.\ &\big(\exists x.\ x \in X\big) \wedge \\ &\big(\forall x.\ x \in X \Rightarrow a(x)\big) \wedge \\ &\forall x.\ x \in X \Rightarrow\ \exists y.\ x \prec y\ \wedge\ y \in X\end{aligned}$$

— b at even depth:

...

Decidability

Decidability

Theorem (Büchi 1962)

The MSO theory of $(\omega, \leqslant)$ is decidable.

Decidability

Theorem (Büchi 1962)

The MSO theory of $(\omega, \leqslant)$ is decidable.

Theorem (Rabin 1969)

The MSO theory of $(\{\mathtt{L}, \mathtt{R}\}^*, \leqslant_{\mathrm{lex}}, \preceq)$ is decidable.

Decidability

Theorem (Büchi 1962)

The MSO theory of $(\omega, \leqslant)$ is decidable.

Theorem (Rabin 1969)

The MSO theory of $(\{\mathtt{L}, \mathtt{R}\}^*, \leqslant_{\text{lex}}, \preceq)$ is decidable.

$\rightsquigarrow$ For an MSO formula φ it is decidable if:

Decidability

Theorem (Büchi 1962)

The MSO theory of $(\omega, \leqslant)$ is decidable.

Theorem (Rabin 1969)

The MSO theory of $(\{\mathtt{L}, \mathtt{R}\}^*, \leqslant_{\mathrm{lex}}, \preceq)$ is decidable.

$\rightsquigarrow$ For an MSO formula φ it is decidable if:

- there is a word / tree satisfying φ?

Decidability

Theorem (Büchi 1962)

The MSO theory of $(\omega, \leqslant)$ is decidable.

Theorem (Rabin 1969)

The MSO theory of $(\{\mathtt{L}, \mathtt{R}\}^*, \leqslant_{\mathrm{lex}}, \preceq)$ is decidable.

$\rightsquigarrow$ For an MSO formula φ it is decidable if:

- there is a word / tree satisfying φ?
- all words / trees satisfy φ?

Decidability

Theorem (Büchi 1962)

The MSO theory of $(\omega, \leqslant)$ is decidable.

Theorem (Rabin 1969)

The MSO theory of $(\{\mathtt{L}, \mathtt{R}\}^*, \leqslant_{\mathrm{lex}}, \preceq)$ is decidable.

$\rightsquigarrow$ For an MSO formula φ it is decidable if:

- there is a word / tree satisfying φ?
- all words / trees satisfy φ?
- φ implies ψ?

Decidability

Theorem (Büchi 1962)

The MSO theory of $(\omega, \leqslant)$ is decidable.

Theorem (Rabin 1969)

The MSO theory of $(\{\mathtt{L}, \mathtt{R}\}^*, \leqslant_{\text{lex}}, \preceq)$ is decidable.

$\rightsquigarrow$ For an MSO formula φ it is decidable if:

- there is a word / tree satisfying φ?
- all words / trees satisfy φ?
- φ implies ψ?
- ...

Decidability

Theorem (Büchi 1962)

The MSO theory of $(\omega, \leqslant)$ is decidable.

Theorem (Rabin 1969)

The MSO theory of $\left(\{\mathtt{L}, \mathtt{R}\}^*, \leqslant_{\mathrm{lex}}, \preceq\right)$ is decidable.

$\rightsquigarrow$ For an MSO formula φ it is decidable if:

- there is a word / tree satisfying φ?
- all words / trees satisfy φ?
- φ implies ψ?
- ...

(the same for all weaker logics)

Decidability

Theorem (Büchi 1962)

The MSO theory of $(\omega, \leqslant)$ is decidable.

Theorem (Rabin 1969)

The MSO theory of $(\{\mathtt{L}, \mathtt{R}\}^*, \leqslant_{\mathrm{lex}}, \preceq)$ is decidable.

$\rightsquigarrow$ For an MSO formula φ it is decidable if:

- there is a word / tree satisfying φ?
- all words / trees satisfy φ?
- φ implies ψ?
- ...

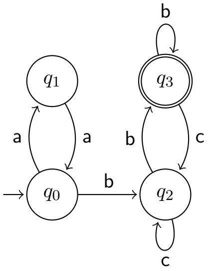
(the same for all weaker logics)

"the mother of all decidability results"

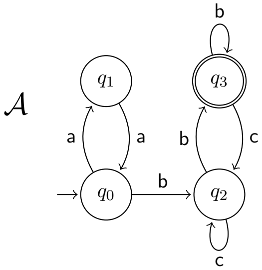
Automata

Automata

$\mathcal{A}$



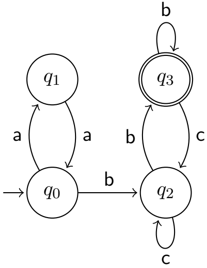
Automata



a run of $\mathcal{A}$ over α

Automata

$\mathcal{A}$

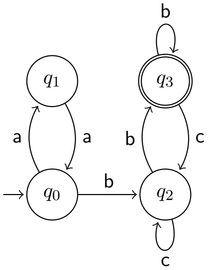


a run of $\mathcal{A}$ over α

$$\alpha = \quad a \quad a \quad b \quad c \quad b \quad b \quad c \quad \cdots$$

Automata

$\mathcal{A}$

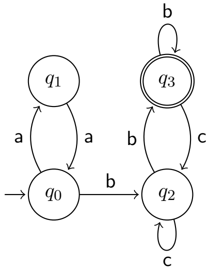


a run of $\mathcal{A}$ over α

$$\begin{array}{rcccccccc} \alpha = & a & a & b & c & b & b & c & \cdots \\ \rho = & q_0 \rightarrow & q_1 \rightarrow & q_0 \rightarrow & q_2 \rightarrow & q_2 \rightarrow & q_3 \rightarrow & q_3 \rightarrow & q_2 \cdots \end{array}$$

Automata

$\mathcal{A}$

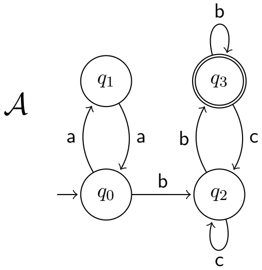


a run of $\mathcal{A}$ over α

$$\begin{array}{llllllllll} \alpha = & a & a & b & c & b & b & c & \cdots \\ \rho = & q_0 \to q_1 \to & q_0 \to & q_2 \to & q_2 \to & q_3 \to & q_3 \to & q_2 & \cdots \end{array}$$

(similarly for tree automata)

Automata



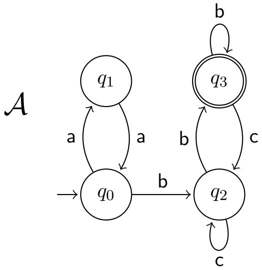
a run of $\mathcal{A}$ over α

$$\alpha = \quad a \quad a \quad b \quad c \quad b \quad b \quad c \quad \cdots$$
$$\rho = \quad q_0 \to q_1 \to q_0 \to q_2 \to q_2 \to q_3 \to q_3 \to q_2 \quad \cdots$$

(similarly for tree automata)

$\mathcal{A}$ accepts α if it has a run over α satisfying the acceptance condition

Automata



a run of $\mathcal{A}$ over α

$$\alpha = \quad a \quad a \quad b \quad c \quad b \quad b \quad c \quad \cdots$$
$$\rho = \quad q_0 \to q_1 \to q_0 \to q_2 \to q_2 \to q_3 \to q_3 \to q_2 \cdots$$

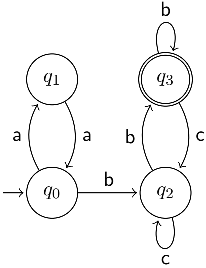
(similarly for tree automata)

$\mathcal{A}$ accepts α if it has a run over α satisfying the acceptance condition

- Büchi: infinitely many accepting states

Automata

$\mathcal{A}$



a run of $\mathcal{A}$ over α

$$\alpha = \quad a \quad a \quad b \quad c \quad b \quad b \quad c \quad \cdots$$
$$\rho = \quad q_0 \to q_1 \to q_0 \to q_2 \to q_2 \to q_3 \to q_3 \to q_2 \quad \cdots$$

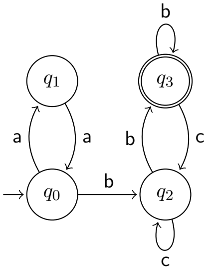
(similarly for tree automata)

$\mathcal{A}$ accepts α if it has a run over α satisfying the acceptance condition

- Büchi: infinitely many accepting states
- parity: dominating priority is even

Automata

$\mathcal{A}$



a run of $\mathcal{A}$ over α

$$\begin{array}{llllllllll} \alpha = & a & a & b & c & b & b & c & \cdots \\ \rho = & q_0 \to & q_1 \to & q_0 \to & q_2 \to & q_2 \to & q_3 \to & q_3 \to & q_2 & \cdots \end{array}$$

(similarly for tree automata)

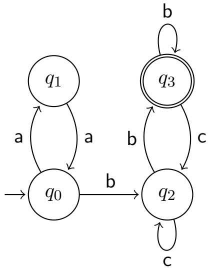
$\mathcal{A}$ accepts α if it has a run over α satisfying the acceptance condition

- Büchi: infinitely many accepting states
- parity: dominating priority is even
- Muller: Boolean combination of

"q appears infinitely often"

Automata

$\mathcal{A}$



a run of $\mathcal{A}$ over α

$$\alpha = \quad a \quad a \quad b \quad c \quad b \quad b \quad c \quad \cdots$$
$$\rho = \quad q_0 \to q_1 \to q_0 \to q_2 \to q_2 \to q_3 \to q_3 \to q_2 \cdots$$

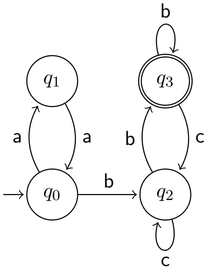
(similarly for tree automata)

$\mathcal{A}$ accepts α if it has a run over α satisfying the acceptance condition

- Büchi: infinitely many accepting states
- parity: dominating priority is even
- Muller: Boolean combination of "q appears infinitely often"
- ...

Automata

$\mathcal{A}$



a run of $\mathcal{A}$ over α

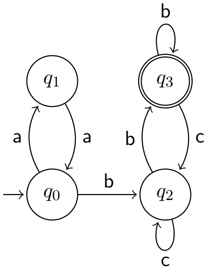
$$\begin{array}{lcccccccc} \alpha = & a & a & b & c & b & b & c & \cdots \\ \rho = & q_0 \to & q_1 \to & q_0 \to & q_2 \to & q_2 \to & q_3 \to & q_3 \to & q_2 \cdots \end{array}$$

(similarly for tree automata)

$\mathcal{A}$ accepts α if it has a run over α satisfying the acceptance condition

Automata

$\mathcal{A}$



a run of $\mathcal{A}$ over α

$$\begin{array}{llllllllll} \alpha = & a & a & b & c & b & b & c & \cdots \\ \rho = & q_0 \to q_1 \to q_0 \to q_2 \to q_2 \to q_3 \to q_3 \to q_2 & & & & & & & \cdots \end{array}$$

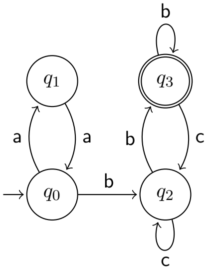
(similarly for tree automata)

$\mathcal{A}$ accepts α if it has a run over α satisfying the acceptance condition

Equivalence

Automata

$\mathcal{A}$



a run of $\mathcal{A}$ over α

$$\alpha = \quad a \quad a \quad b \quad c \quad b \quad b \quad c \quad \cdots$$
$$\rho = \quad q_0 \to q_1 \to q_0 \to q_2 \to q_2 \to q_3 \to q_3 \to q_2 \quad \cdots$$

(similarly for tree automata)

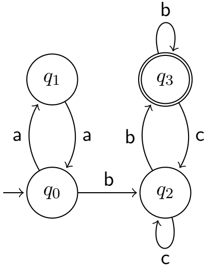
$\mathcal{A}$ accepts α if it has a run over α satisfying the acceptance condition

Equivalence

formula φ $\leftrightsquigarrow$ $\mathcal{A}$ automaton

Automata

$\mathcal{A}$



a run of $\mathcal{A}$ over α

$$\alpha = \quad a \quad a \quad b \quad c \quad b \quad b \quad c \quad \cdots$$
$$\rho = \quad q_0 \to q_1 \to q_0 \to q_2 \to q_2 \to q_3 \to q_3 \to q_2 \quad \cdots$$

(similarly for tree automata)

$\mathcal{A}$ accepts α if it has a run over α satisfying the acceptance condition

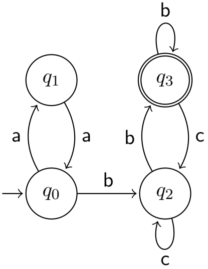
Equivalence

formula $\varphi \leftrightsquigarrow \mathcal{A}$ automaton

α satisfies φ if and only if $\mathcal{A}$ accepts α

Automata

$\mathcal{A}$



a run of $\mathcal{A}$ over α

$$\alpha = \quad a \quad a \quad b \quad c \quad b \quad b \quad c \quad \cdots$$
$$\rho = \quad q_0 \to q_1 \to q_0 \to q_2 \to q_2 \to q_3 \to q_3 \to q_2 \quad \cdots$$

(similarly for tree automata)

$\mathcal{A}$ accepts α if it has a run over α satisfying the acceptance condition

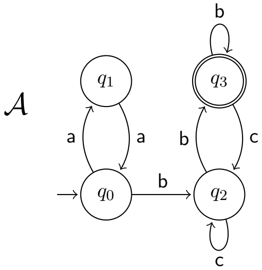
Equivalence

formula $\varphi \leftrightsquigarrow \mathcal{A}$ automaton

α satisfies φ if and only if $\mathcal{A}$ accepts α

Question: is φ satisfiable?

Automata



a run of $\mathcal{A}$ over α

$$\alpha = \quad a \quad a \quad b \quad c \quad b \quad b \quad c \quad \cdots$$
$$\rho = \quad q_0 \to q_1 \to q_0 \to q_2 \to q_2 \to q_3 \to q_3 \to q_2 \quad \cdots$$

(similarly for tree automata)

$\mathcal{A}$ accepts α if it has a run over α satisfying the acceptance condition

Equivalence

formula $\varphi \leftrightsquigarrow \mathcal{A}$ automaton

α satisfies φ if and only if $\mathcal{A}$ accepts α

Question: is φ satisfiable?

Solution: construct $\mathcal{A}$ and look for accepting loops

Model-checking

Model-checking

Given a machine

$$M$$

Model-checking

Given a machine

M

and a specification

φ

Model-checking

Given a machine and a specification

M φ

does M satisfy φ ?

Model-checking

Given a machine

M

and a specification

φ

does M satisfy φ ?

$\rightsquigarrow$

automaton for $\neg\varphi$

$\mathcal{A}$

Model-checking

Given a machine M

and a specification φ

does M satisfy φ ?

automaton for $\neg\varphi$

$\mathcal{A}$

$M \times \mathcal{A}$

Model-checking

Given a machine M

and a specification φ

does M satisfy φ ?

automaton for $\neg\varphi$

$\mathcal{A}$

$M \times \mathcal{A}$

look for accepting loops in the product graph

Expressive power

Expressive power

Logics:

Expressive power

Logics:

FO

Expressive power

Logics: Quantifiers:

FO $\exists_x$

$+$ dual

Expressive power

Logics:	Quantifiers:
FO	$\exists_x$ + dual
WMSO	$\exists_x$, $\exists^{\mathsf{fin}}_X$ + dual

Expressive power

Logics:	Quantifiers:
FO	$\exists_x$ + dual
WMSO	$\exists_x$, $\exists_X^{\text{fin}}$ + dual
$\exists$MSO	$\exists_{X_1} \ldots \exists_{X_n}\ \psi$ for $\psi \in$WMSO

Expressive power

Logics:	Quantifiers:
FO	$\exists_x$ + dual
WMSO	$\exists_x$, $\exists_X^{\mathsf{fin}}$ + dual
$\exists$MSO	$\exists_{X_1} \ldots \exists_{X_n}\ \psi$ for $\psi \in$WMSO
MSO	$\exists_x$, $\exists_X^{\mathsf{fin}}$, $\exists_X$ + dual

Expressive power

Logics:	Quantifiers:	Automata:
FO	$\exists_x$ + dual	
WMSO	$\exists_x,\ \exists_X^{\mathsf{fin}}$ + dual	
$\exists$MSO	$\exists_{X_1} \ldots \exists_{X_n}\ \psi$ for $\psi \in$WMSO	
MSO	$\exists_x,\ \exists_X^{\mathsf{fin}},\ \exists_X$ + dual	

Expressive power

Logics:	Quantifiers:	Automata:
FO	$\exists_x$ + dual	(no natural class)
WMSO	$\exists_x$, $\exists^{\text{fin}}_X$ + dual	
$\exists$MSO	$\exists_{X_1} \ldots \exists_{X_n}\ \psi$ for $\psi \in$WMSO	
MSO	$\exists_x$, $\exists^{\text{fin}}_X$, $\exists_X$ + dual	

Expressive power

Logics:	Quantifiers:	Automata:
FO	$\exists_x$ + dual	(no natural class)
WMSO	$\exists_x$, $\exists_X^{\mathsf{fin}}$ + dual	weak alternating
$\exists$MSO	$\exists_{X_1} \ldots \exists_{X_n}\ \psi$ for $\psi \in$WMSO	
MSO	$\exists_x$, $\exists_X^{\mathsf{fin}}$, $\exists_X$ + dual	

Expressive power

Logics:	Quantifiers:	Automata:
FO	$\exists_x$ + dual	(no natural class)
WMSO	$\exists_x$, $\exists_X^{\text{fin}}$ + dual	weak alternating
$\exists$MSO	$\exists_{X_1} \ldots \exists_{X_n}\ \psi$ for $\psi \in$WMSO	non-deterministic Büchi alternating Büchi
MSO	$\exists_x$, $\exists_X^{\text{fin}}$, $\exists_X$ + dual	

Expressive power

Logics:	Quantifiers:	Automata:
FO	$\exists_x$ + dual	(no natural class)
WMSO	$\exists_x$, $\exists_X^{\text{fin}}$ + dual	weak alternating
$\exists$MSO	$\exists_{X_1} \ldots \exists_{X_n}\ \psi$ for $\psi \in$WMSO	non-deterministic Büchi alternating Büchi
MSO	$\exists_x$, $\exists_X^{\text{fin}}$, $\exists_X$ + dual	non-deterministic parity alternating parity

Expressive power

Logics:	Quantifiers:	Automata:
FO	$\exists_x$ + dual	(no natural class)
$\cap\!\mid$		
WMSO	$\exists_x$, $\exists_X^{\text{fin}}$ + dual	weak alternating
$\cap\!\mid$		
$\exists$MSO	$\exists_{X_1} \ldots \exists_{X_n}\ \psi$ for $\psi \in$WMSO	non-deterministic Büchi alternating Büchi
$\cap\!\mid$		
MSO	$\exists_x$, $\exists_X^{\text{fin}}$, $\exists_X$ + dual	non-deterministic parity alternating parity

Expressive power

Logics:	Quantifiers:	Automata:
FO	$\exists_x$ + dual	(no natural class)
$\cap\vert$		
WMSO	$\exists_x$, $\exists_X^{\mathsf{fin}}$ + dual	weak alternating
$\cap\vert$		
$\exists$MSO	$\exists_{X_1} \ldots \exists_{X_n}\ \psi$ for $\psi \in$WMSO	non-deterministic Büchi alternating Büchi
$\cap\vert$		
MSO	$\exists_x$, $\exists_X^{\mathsf{fin}}$, $\exists_X$ + dual	non-deterministic parity alternating parity

Problem: characterise these logics

Expressive power

Logics:	Quantifiers:	Automata:
FO	$\exists_x$ + dual	(no natural class)
$\cap\vert$		
WMSO	$\exists_x$, $\exists^{\mathsf{fin}}_X$ + dual	weak alternating
$\cap\vert$		
$\exists$MSO	$\exists_{X_1} \ldots \exists_{X_n}\ \psi$ for $\psi \in$WMSO	non-deterministic Büchi alternating Büchi
$\cap\vert$		
MSO	$\exists_x$, $\exists^{\mathsf{fin}}_X$, $\exists_X$ + dual	non-deterministic parity alternating parity

Problem: characterise these logics

Input: φ from a stronger logic $\mathcal{L}$

Expressive power

Logics:	Quantifiers:	Automata:
FO	$\exists_x$ + dual	(no natural class)
$\cap\vert$		
WMSO	$\exists_x$, $\exists^{\text{fin}}_X$ + dual	weak alternating
$\cap\vert$		
$\exists$MSO	$\exists_{X_1} \ldots \exists_{X_n}\ \psi$ for $\psi \in$WMSO	non-deterministic Büchi alternating Büchi
$\cap\vert$		
MSO	$\exists_x$, $\exists^{\text{fin}}_X$, $\exists_X$ + dual	non-deterministic parity alternating parity

Problem: characterise these logics

Input: φ from a stronger logic $\mathcal{L}$

Output: is there $\psi \equiv \varphi$ in a weaker logic $\mathcal{L}'$?

Expressive power: (finite) and infinite words

Expressive power: (finite) and infinite words

$$\mathrm{FO} \subsetneq \mathrm{WMSO} \equiv \exists\mathrm{MSO} \equiv \mathrm{MSO}$$

Expressive power: (finite) and infinite words

$$\mathrm{FO} \subsetneq \mathrm{WMSO} \equiv \exists\mathrm{MSO} \equiv \mathrm{MSO}$$

Theorem (McNaughton, Kupferman, Vardi)

$\mathrm{WMSO} \equiv \mathrm{MSO}$ over infinite words

Expressive power: (finite) and infinite words

$$\mathrm{FO} \subsetneq \mathrm{WMSO} \equiv \exists\mathrm{MSO} \equiv \mathrm{MSO}$$

Theorem (McNaughton, Kupferman, Vardi)

$\mathrm{WMSO} \equiv \mathrm{MSO}$ over infinite words

Theorem (Schutzenberger, McNaughton, Papert, Thomas)

It is decidable if a given MSO formula is equivalent to an FO formula.

Expressive power: (finite) and infinite words

$$\mathrm{FO} \subsetneq \mathrm{WMSO} \equiv \exists\mathrm{MSO} \equiv \mathrm{MSO}$$

Theorem (McNaughton, Kupferman, Vardi)

$\mathrm{WMSO} \equiv \mathrm{MSO}$ over infinite words

Theorem (Schutzenberger, McNaughton, Papert, Thomas)

It is decidable if a given MSO formula is equivalent to an FO formula.

$\rightsquigarrow$ aperiodic monoids, counter-free automata, star-free regular expressions, ...

Expressive power: (finite) and infinite words

$$\mathrm{FO} \subsetneq \mathrm{WMSO} \equiv \exists\mathrm{MSO} \equiv \mathrm{MSO}$$

Theorem (McNaughton, Kupferman, Vardi)

$\mathrm{WMSO} \equiv \mathrm{MSO}$ over infinite words

Theorem (Schutzenberger, McNaughton, Papert, Thomas)

It is decidable if a given MSO formula is equivalent to an FO formula.

$\rightsquigarrow$ aperiodic monoids, counter-free automata, star-free regular expressions, ...

Expressive power: infinite trees

Expressive power: (finite) and infinite words

$$\text{FO} \subsetneq \text{WMSO} \equiv \exists\text{MSO} \equiv \text{MSO}$$

Theorem (McNaughton, Kupferman, Vardi)

WMSO $\equiv$ MSO over infinite words

Theorem (Schutzenberger, McNaughton, Papert, Thomas)

It is decidable if a given MSO formula is equivalent to an FO formula.

$\rightsquigarrow$ aperiodic monoids, counter-free automata, star-free regular expressions, ...

Expressive power: infinite trees

$$\text{FO} \subsetneq \text{WMSO} \subsetneq \exists\text{MSO} \subsetneq \text{MSO}$$

Expressive power: (finite) and infinite words

$$\mathrm{FO} \subsetneq \mathrm{WMSO} \equiv \exists\mathrm{MSO} \equiv \mathrm{MSO}$$

Theorem (McNaughton, Kupferman, Vardi)
WMSO ≡ MSO over infinite words

Theorem (Schutzenberger, McNaughton, Papert, Thomas)
It is decidable if a given MSO formula is equivalent to an FO formula.

⇝ aperiodic monoids, counter-free automata, star-free regular expressions, . . .

Expressive power: infinite trees

$$\mathrm{FO} \subsetneq \mathrm{WMSO} \subsetneq \exists\mathrm{MSO} \subsetneq \mathrm{MSO}$$

Theorem (Colcombet, Kuperberg, Löding, Vanden Boom [2013])
It is decidable if a given ∃MSO formula is equivalent to a WMSO formula.

Expressive power: (finite) and infinite words

$$\mathrm{FO} \subsetneq \mathrm{WMSO} \equiv \exists\mathrm{MSO} \equiv \mathrm{MSO}$$

Theorem (McNaughton, Kupferman, Vardi)

WMSO ≡ MSO over infinite words

Theorem (Schutzenberger, McNaughton, Papert, Thomas)

It is decidable if a given MSO formula is equivalent to an FO formula.

⇝ aperiodic monoids, counter-free automata, star-free regular expressions, ...

Expressive power: infinite trees

$$\mathrm{FO} \subsetneq \mathrm{WMSO} \subsetneq \exists\mathrm{MSO} \subsetneq \mathrm{MSO}$$

Theorem (Colcombet, Kuperberg, Löding, Vanden Boom [2013])

It is decidable if a given ∃MSO formula is equivalent to a WMSO formula.

Expressive power: (finite) and infinite words

$$\mathrm{FO} \subsetneq \mathrm{WMSO} \equiv \exists\mathrm{MSO} \equiv \mathrm{MSO}$$

Theorem (McNaughton, Kupferman, Vardi)
WMSO $\equiv$ MSO over infinite words

Theorem (Schutzenberger, McNaughton, Papert, Thomas)
It is decidable if a given MSO formula is equivalent to an FO formula.

$\rightsquigarrow$ aperiodic monoids, counter-free automata, star-free regular expressions, . . .

Expressive power: infinite trees

$$\mathrm{FO} \subsetneq \mathrm{WMSO} \subsetneq \exists\mathrm{MSO} \subsetneq \mathrm{MSO}$$

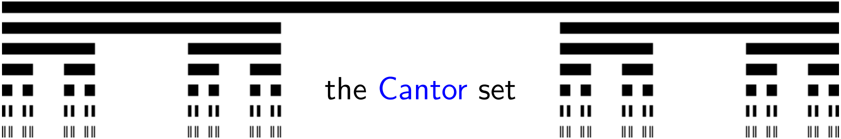
Theorem (Colcombet, Kuperberg, Löding, Vanden Boom [2013])
It is decidable if a given $\exists$MSO formula is equivalent to a WMSO formula.

(theory of regular cost functions)

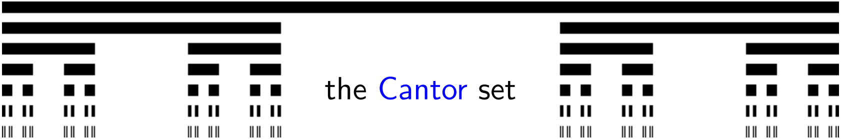
Topology

Topology

Topology



Topology

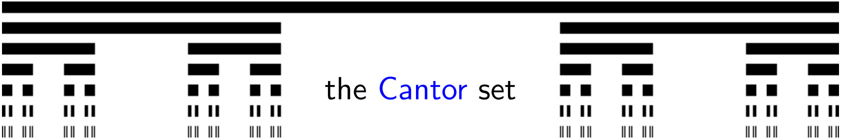


the Cantor set

$\cong$

A^ω with the product topology

Topology



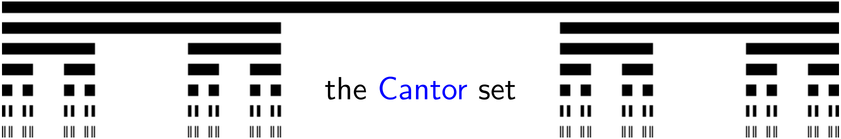
the Cantor set

$\cong$

A^ω with the product topology

(infinite words)

Topology



the Cantor set

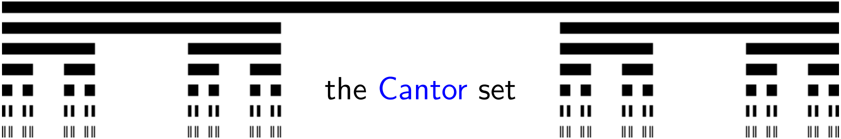
$$\cong$$

A^ω with the product topology

(infinite words)

$$\{0,2\}^\omega \ni (2,0,2,2,0,0,0,2,\ldots) \longmapsto 0.20220002_3 \in [0,1]$$

Topology



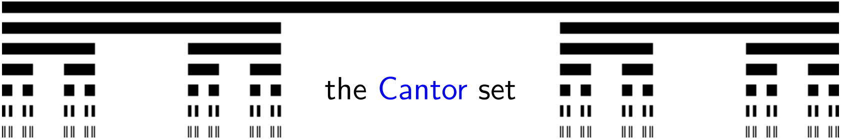
the Cantor set

$\cong$

A^ω with the product topology

(infinite words)

Topology



the Cantor set

$\cong$

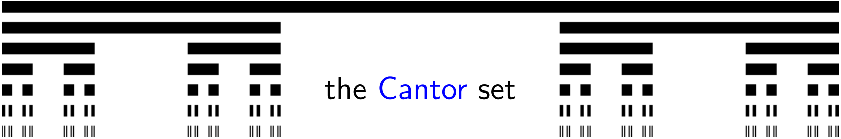
A^ω with the product topology

(infinite words)

$\cong$

$A^{(\{\mathtt{L},\mathtt{R}\}^*)}$ with the product topology

Topology



the Cantor set

$\cong$

A^{ω} with the product topology

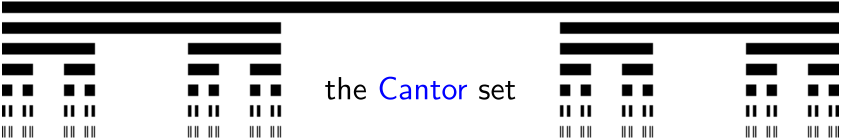
(infinite words)

$\cong$

$A^{(\{\mathtt{L},\mathtt{R}\}^*)}$ with the product topology

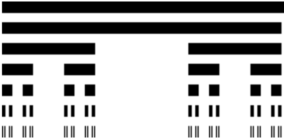
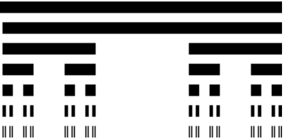
(infinite trees)

Topology



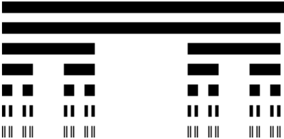
the Cantor set

Topology

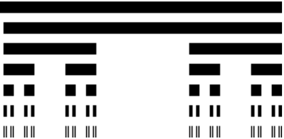

the Cantor set


Pointwise convergence

Topology



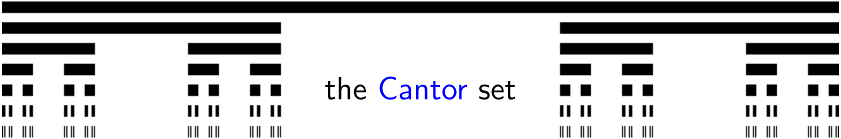
the Cantor set



Pointwise convergence

$a \quad a \quad b \quad c \quad b \quad b \quad c \quad a \quad c \quad b \quad a \quad c \quad a \quad b \quad c \quad b \quad c \quad \cdots$

Topology

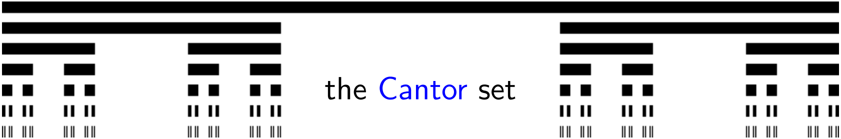


Pointwise convergence

$$\color{red}{a} \quad \color{red}{a} \quad b \quad c \quad b \quad b \quad c \quad a \quad c \quad b \quad a \quad c \quad a \quad b \quad c \quad b \quad c \quad \cdots$$

$$\color{red}{a} \quad \color{red}{a} \quad \color{red}{a} \quad b \quad c \quad a \quad b \quad a \quad b \quad c \quad c \quad a \quad c \quad b \quad c \quad c \quad a \quad \cdots$$

Topology



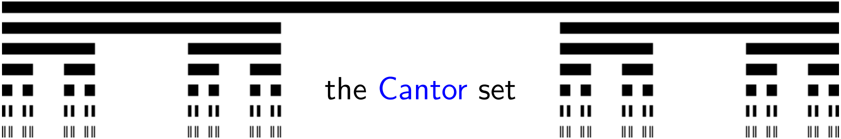
Pointwise convergence

$$a \quad a \quad b \quad c \quad b \quad b \quad c \quad a \quad c \quad b \quad a \quad c \quad a \quad b \quad c \quad b \quad c \quad \cdots$$

$$a \quad a \quad a \quad b \quad c \quad a \quad b \quad a \quad b \quad c \quad c \quad a \quad c \quad b \quad c \quad c \quad a \quad \cdots$$

$$a \quad a \quad a \quad a \quad a \quad a \quad a \quad c \quad c \quad b \quad c \quad a \quad b \quad c \quad a \quad c \quad c \quad \cdots$$

Topology



Pointwise convergence

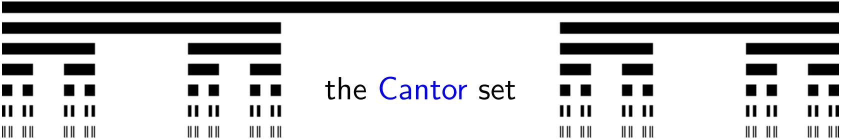
$$a\ a\ b\ c\ b\ b\ c\ a\ c\ b\ a\ c\ a\ b\ c\ b\ c\ \cdots$$

$$a\ a\ a\ b\ c\ a\ b\ a\ b\ c\ c\ a\ c\ b\ c\ c\ a\ \cdots$$

$$a\ a\ a\ a\ a\ a\ a\ c\ c\ b\ c\ a\ b\ c\ a\ c\ c\ \cdots$$

$$a\ a\ a\ a\ a\ c\ b\ a\ c\ b\ a\ c\ b\ b\ b\ a\ c\ \cdots$$

Topology



Pointwise convergence

$$a\ a\ b\ c\ b\ b\ c\ a\ c\ b\ a\ c\ a\ b\ c\ b\ c\ \cdots$$

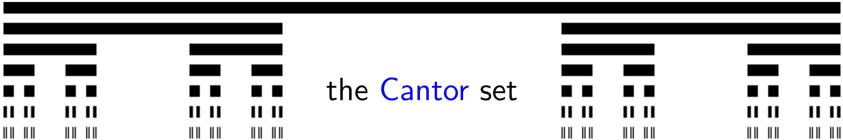
$$a\ a\ a\ b\ c\ a\ b\ a\ b\ c\ c\ a\ c\ b\ c\ c\ a\ \cdots$$

$$a\ a\ a\ a\ a\ a\ a\ c\ c\ b\ c\ a\ b\ c\ a\ c\ c\ \cdots$$

$$a\ a\ a\ a\ a\ c\ b\ a\ c\ b\ a\ c\ b\ b\ b\ a\ c\ \cdots$$

$$a\ a\ a\ a\ a\ a\ a\ c\ b\ a\ c\ a\ b\ b\ c\ b\ c\ \cdots$$

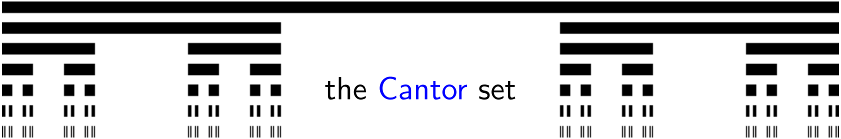
Topology



Pointwise convergence

a	a	b	c	b	b	c	a	c	b	a	c	a	b	c	b	c	$\cdots$
a	a	a	b	c	a	b	a	b	c	c	a	c	b	c	c	a	$\cdots$
a	a	a	a	a	a	a	c	c	b	c	a	b	c	a	c	c	$\cdots$
a	a	a	a	a	c	b	a	c	b	a	c	b	b	b	a	c	$\cdots$
a	a	a	a	a	a	a	c	b	a	c	a	b	b	c	b	c	$\cdots$
a	a	a	a	a	a	a	a	a	a	c	b	a	c	a	b	b	$\cdots$

Topology



Pointwise convergence

a	a	b	c	b	b	c	a	c	b	a	c	a	b	c	b	c	$\cdots$
a	a	a	b	c	a	b	a	b	c	c	a	c	b	c	c	a	$\cdots$
a	a	a	a	a	a	a	c	c	b	c	a	b	c	a	c	c	$\cdots$
a	a	a	a	a	c	b	a	c	b	a	c	b	b	b	a	c	$\cdots$
a	a	a	a	a	a	a	c	b	a	c	a	b	b	c	b	c	$\cdots$
a	a	a	a	a	a	a	a	a	a	c	b	a	c	a	b	b	$\cdots$
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	

Topology

Pointwise convergence

a	a	b	c	b	b	c	a	c	b	a	c	a	b	c	b	c	$\cdots$
a	a	a	b	c	a	b	a	b	c	c	a	c	b	c	c	a	$\cdots$
a	a	a	a	a	a	a	c	c	b	c	a	b	c	a	c	c	$\cdots$
a	a	a	a	a	c	b	a	c	b	a	c	b	b	b	a	c	$\cdots$
a	a	a	a	a	a	a	c	b	a	c	a	b	b	c	b	c	$\cdots$
a	a	a	a	a	a	a	a	a	a	c	b	a	c	a	b	b	$\cdots$
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	
a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	$\cdots$

Descriptive set theory

Descriptive set theory

basic open sets in A^ω

Descriptive set theory

basic open sets in A^ω = finite Boolean combinations of: "the i-th letter is a_i"

Descriptive set theory

basic open sets in A^ω $=$ finite Boolean combinations of: "the i-th letter is a_i"

$$\{\alpha \in A^\omega \mid \alpha(7) = b \vee (\alpha(3) = a \wedge \alpha(4) \neq c)\}$$

Descriptive set theory

basic open sets in A^ω $=$ finite Boolean combinations of: "the i-th letter is a_i"

Descriptive set theory

basic open sets in A^ω = finite Boolean combinations of: "the i-th letter is a_i"

open sets = (countable) unions of basic sets

Descriptive set theory

basic open sets in A^ω	$=$	finite Boolean combinations of: "the i-th letter is a_i"
open sets	$=$	(countable) unions of basic sets
closed sets	$=$	complements of open sets

Descriptive set theory

basic open sets in A^ω		=	finite Boolean combinations of: "the i-th letter is a_i"
open sets	$\mathbf{\Sigma}_1^0$	=	(countable) unions of basic sets
closed sets	$\mathbf{\Pi}_1^0$	=	complements of open sets

Descriptive set theory

basic open sets in A^ω		$=$	finite Boolean combinations of: "the i-th letter is a_i"
open sets	$\mathbf{\Sigma}^0_1$	$=$	(countable) unions of basic sets
closed sets	$\mathbf{\Pi}^0_1$	$=$	complements of open sets
	$\mathbf{\Sigma}^0_\eta$	$=$	countable unions of $\bigcup_{\beta<\eta} \mathbf{\Pi}^0_\beta$ sets

Descriptive set theory

basic open sets in A^ω		$=$	finite Boolean combinations of: "the i-th letter is a_i"
open sets	$\mathbf{\Sigma}^0_1$	$=$	(countable) unions of basic sets
closed sets	$\mathbf{\Pi}^0_1$	$=$	complements of open sets
	$\mathbf{\Sigma}^0_\eta$	$=$	countable unions of $\bigcup_{\beta<\eta} \mathbf{\Pi}^0_\beta$ sets
	$\mathbf{\Pi}^0_\eta$	$=$	complements of $\mathbf{\Sigma}^0_\eta$ sets

Descriptive set theory

basic open sets in A^ω		$=$	finite Boolean combinations of: "the i-th letter is a_i"
open sets	$\mathbf{\Sigma}^0_1$	$=$	(countable) unions of basic sets
closed sets	$\mathbf{\Pi}^0_1$	$=$	complements of open sets
	$\mathbf{\Sigma}^0_\eta$	$=$	countable unions of $\bigcup_{\beta<\eta} \mathbf{\Pi}^0_\beta$ sets
	$\mathbf{\Pi}^0_\eta$	$=$	complements of $\mathbf{\Sigma}^0_\eta$ sets
Borel sets		$=$	$\bigcup_{\eta<\omega_1} \mathbf{\Sigma}^0_\eta = \bigcup_{\eta<\omega_1} \mathbf{\Pi}^0_\eta$

Descriptive set theory

basic open sets in A^ω		$=$	finite Boolean combinations of: "the i-th letter is a_i"
open sets	$\mathbf{\Sigma}^0_1$	$=$	(countable) unions of basic sets
closed sets	$\mathbf{\Pi}^0_1$	$=$	complements of open sets
	$\mathbf{\Sigma}^0_\eta$	$=$	countable unions of $\bigcup_{\beta<\eta} \mathbf{\Pi}^0_\beta$ sets
	$\mathbf{\Pi}^0_\eta$	$=$	complements of $\mathbf{\Sigma}^0_\eta$ sets
Borel sets		$=$	$\bigcup_{\eta<\omega_1} \mathbf{\Sigma}^0_\eta = \bigcup_{\eta<\omega_1} \mathbf{\Pi}^0_\eta$
analytic sets		$=$	projections of Borel sets

Descriptive set theory

basic open sets in A^ω		$=$	finite Boolean combinations of: "the i-th letter is a_i"
open sets	$\boldsymbol{\Sigma}^0_1$	$=$	(countable) unions of basic sets
closed sets	$\boldsymbol{\Pi}^0_1$	$=$	complements of open sets
	$\boldsymbol{\Sigma}^0_\eta$	$=$	countable unions of $\bigcup_{\beta<\eta} \boldsymbol{\Pi}^0_\beta$ sets
	$\boldsymbol{\Pi}^0_\eta$	$=$	complements of $\boldsymbol{\Sigma}^0_\eta$ sets
Borel sets		$=$	$\bigcup_{\eta<\omega_1} \boldsymbol{\Sigma}^0_\eta = \bigcup_{\eta<\omega_1} \boldsymbol{\Pi}^0_\eta$
analytic sets		$=$	projections of Borel sets
co-analytic sets		$=$	complements of analytic sets

Descriptive set theory

basic open sets in A^ω		$=$	finite Boolean combinations of: "the i-th letter is a_i"
open sets	$\mathbf{\Sigma}^0_1$	$=$	(countable) unions of basic sets
closed sets	$\mathbf{\Pi}^0_1$	$=$	complements of open sets
	$\mathbf{\Sigma}^0_\eta$	$=$	countable unions of $\bigcup_{\beta<\eta} \mathbf{\Pi}^0_\beta$ sets
	$\mathbf{\Pi}^0_\eta$	$=$	complements of $\mathbf{\Sigma}^0_\eta$ sets
Borel sets		$=$	$\bigcup_{\eta<\omega_1} \mathbf{\Sigma}^0_\eta = \bigcup_{\eta<\omega_1} \mathbf{\Pi}^0_\eta$
analytic sets	$\mathbf{\Sigma}^1_1$	$=$	projections of Borel sets
co-analytic sets	$\mathbf{\Pi}^1_1$	$=$	complements of analytic sets

Descriptive set theory

basic open sets in A^ω		$=$	finite Boolean combinations of: "the i-th letter is a_i"
open sets	$\mathbf{\Sigma}^0_1$	$=$	(countable) unions of basic sets
closed sets	$\mathbf{\Pi}^0_1$	$=$	complements of open sets
	$\mathbf{\Sigma}^0_\eta$	$=$	countable unions of $\bigcup_{\beta<\eta} \mathbf{\Pi}^0_\beta$ sets
	$\mathbf{\Pi}^0_\eta$	$=$	complements of $\mathbf{\Sigma}^0_\eta$ sets
Borel sets		$=$	$\bigcup_{\eta<\omega_1} \mathbf{\Sigma}^0_\eta = \bigcup_{\eta<\omega_1} \mathbf{\Pi}^0_\eta$
analytic sets	$\mathbf{\Sigma}^1_1$	$=$	projections of Borel sets
co-analytic sets	$\mathbf{\Pi}^1_1$	$=$	complements of analytic sets
	$\mathbf{\Sigma}^1_{n+1}$	$=$	projections of $\mathbf{\Pi}^1_n$ sets

Descriptive set theory

basic open sets in A^ω		=	finite Boolean combinations of: "the i-th letter is a_i"
open sets	$\boldsymbol{\Sigma}^0_1$	=	(countable) unions of basic sets
closed sets	$\boldsymbol{\Pi}^0_1$	=	complements of open sets
	$\boldsymbol{\Sigma}^0_\eta$	=	countable unions of $\bigcup_{\beta<\eta} \boldsymbol{\Pi}^0_\beta$ sets
	$\boldsymbol{\Pi}^0_\eta$	=	complements of $\boldsymbol{\Sigma}^0_\eta$ sets
Borel sets		=	$\bigcup_{\eta<\omega_1} \boldsymbol{\Sigma}^0_\eta = \bigcup_{\eta<\omega_1} \boldsymbol{\Pi}^0_\eta$
analytic sets	$\boldsymbol{\Sigma}^1_1$	=	projections of Borel sets
co-analytic sets	$\boldsymbol{\Pi}^1_1$	=	complements of analytic sets
	$\boldsymbol{\Sigma}^1_{n+1}$	=	projections of $\boldsymbol{\Pi}^1_n$ sets
	$\boldsymbol{\Pi}^1_n$	=	complements of $\boldsymbol{\Sigma}^1_n$ sets

Descriptive set theory

basic open sets in A^ω		$=$	finite Boolean combinations of: "the i-th letter is a_i"
open sets	$\boldsymbol{\Sigma}^0_1$	$=$	(countable) unions of basic sets
closed sets	$\boldsymbol{\Pi}^0_1$	$=$	complements of open sets
	$\boldsymbol{\Sigma}^0_\eta$	$=$	countable unions of $\bigcup_{\beta<\eta} \boldsymbol{\Pi}^0_\beta$ sets
	$\boldsymbol{\Pi}^0_\eta$	$=$	complements of $\boldsymbol{\Sigma}^0_\eta$ sets
Borel sets		$=$	$\bigcup_{\eta<\omega_1} \boldsymbol{\Sigma}^0_\eta = \bigcup_{\eta<\omega_1} \boldsymbol{\Pi}^0_\eta$
analytic sets	$\boldsymbol{\Sigma}^1_1$	$=$	projections of Borel sets
co-analytic sets	$\boldsymbol{\Pi}^1_1$	$=$	complements of analytic sets
	$\boldsymbol{\Sigma}^1_{n+1}$	$=$	projections of $\boldsymbol{\Pi}^1_n$ sets
	$\boldsymbol{\Pi}^1_n$	$=$	complements of $\boldsymbol{\Sigma}^1_n$ sets

There are continuum many open sets!

Upper bounds

Upper bounds

Syntax:

Upper bounds

Syntax:

$a(x)$

Upper bounds

Syntax:	Set theory:
$a(x)$	basic open set

Upper bounds

Syntax:	Set theory:
$a(x)$	basic open set
$\exists_x$, $\exists^{\mathsf{fin}}_X$	countable union

Upper bounds

Syntax:	Set theory:
$a(x)$	basic open set
$\exists_x$, $\exists_X^{\mathsf{fin}}$	countable union
$\exists_X$	projection from $A \times \{0,1\}$ to A

Upper bounds

Syntax:	Set theory:
$a(x)$	basic open set
$\exists_x$, $\exists^{\mathsf{fin}}_X$	countable union
$\exists_X$	projection from $A \times \{0,1\}$ to A
$\neg$	complementation

Upper bounds

Syntax:	Set theory:
$a(x)$	basic open set
$\exists_x$, $\exists_X^{\text{fin}}$	countable union
$\exists_X$	projection from $A \times \{0,1\}$ to A
$\neg$	complementation

$$\mathrm{L}(\varphi) \stackrel{\text{def}}{=} \{M \mid M \models \varphi\}$$

Upper bounds

Syntax:	Set theory:
$a(x)$	basic open set
$\exists_x$, $\exists^{\text{fin}}_X$	countable union
$\exists_X$	projection from $A \times \{0,1\}$ to A
$\neg$	complementation

$$\mathrm{L}(\varphi) \stackrel{\text{def}}{=} \{M \mid M \models \varphi\}$$

Logic:

Upper bounds

Syntax:	Set theory:
$a(x)$	basic open set
$\exists_x$, $\exists^{\mathsf{fin}}_X$	countable union
$\exists_X$	projection from $A \times \{0,1\}$ to A
$\neg$	complementation

$$\mathrm{L}(\varphi) \stackrel{\mathsf{def}}{=} \{M \mid M \models \varphi\}$$

Logic:

$\varphi \in \mathrm{FO}$

Upper bounds

Syntax:	Set theory:
$a(x)$	basic open set
$\exists_x$, $\exists^{\mathsf{fin}}_X$	countable union
$\exists_X$	projection from $A \times \{0,1\}$ to A
$\neg$	complementation

$$\mathrm{L}(\varphi) \stackrel{\text{def}}{=} \{M \mid M \models \varphi\}$$

Logic:	Upper bound:
$\varphi \in \mathrm{FO}$	$\mathrm{L}(\varphi) \in \mathbf{\Sigma}^0_n$ (finite level of the Borel hierarchy)

Upper bounds

Syntax:	Set theory:
$a(x)$	basic open set
$\exists_x$, $\exists^{\mathsf{fin}}_X$	countable union
$\exists_X$	projection from $A \times \{0,1\}$ to A
$\neg$	complementation

$$\mathrm{L}(\varphi) \stackrel{\mathsf{def}}{=} \{M \mid M \models \varphi\}$$

Logic:	Upper bound:	
$\varphi \in \mathrm{FO}$	$\mathrm{L}(\varphi) \in \boldsymbol{\Sigma}^0_n$	(finite level of the Borel hierarchy)
$\varphi \in \mathrm{WMSO}$	$\mathrm{L}(\varphi) \in \boldsymbol{\Sigma}^0_n$	(finite level of the Borel hierarchy)

Upper bounds

Syntax:	Set theory:
$a(x)$	basic open set
$\exists_x$, $\exists^{\text{fin}}_X$	countable union
$\exists_X$	projection from $A \times \{0,1\}$ to A
$\neg$	complementation

$$\mathrm{L}(\varphi) \stackrel{\text{def}}{=} \{M \mid M \models \varphi\}$$

Logic:	Upper bound:	
$\varphi \in \mathrm{FO}$	$\mathrm{L}(\varphi) \in \mathbf{\Sigma}^0_n$	(finite level of the Borel hierarchy)
$\varphi \in \mathrm{WMSO}$	$\mathrm{L}(\varphi) \in \mathbf{\Sigma}^0_n$	(finite level of the Borel hierarchy)
$\varphi \in \exists\mathrm{MSO}$	$\mathrm{L}(\varphi) \in \mathbf{\Sigma}^1_1$	(analytic set)

Upper bounds

Syntax:	Set theory:
$a(x)$	basic open set
$\exists_x$, $\exists^{\mathsf{fin}}_X$	countable union
$\exists_X$	projection from $A \times \{0,1\}$ to A
$\neg$	complementation

$$\mathrm{L}(\varphi) \stackrel{\mathsf{def}}{=} \{M \mid M \models \varphi\}$$

Logic:	Upper bound:	
$\varphi \in \mathrm{FO}$	$\mathrm{L}(\varphi) \in \mathbf{\Sigma}^0_n$	(finite level of the Borel hierarchy)
$\varphi \in \mathrm{WMSO}$	$\mathrm{L}(\varphi) \in \mathbf{\Sigma}^0_n$	(finite level of the Borel hierarchy)
$\varphi \in \exists\mathrm{MSO}$	$\mathrm{L}(\varphi) \in \mathbf{\Sigma}^1_1$	(analytic set)
$\varphi \in \mathrm{MSO}$	$\mathrm{L}(\varphi) \in \mathbf{\Sigma}^1_n$	(finite level of the projective hierarchy)

Upper bounds

Syntax:	Set theory:
$a(x)$	basic open set
$\exists_x$, $\exists^{\mathsf{fin}}_X$	countable union
$\exists_X$	projection from $A \times \{0,1\}$ to A
$\neg$	complementation

$$\mathrm{L}(\varphi) \stackrel{\mathsf{def}}{=} \{M \mid M \models \varphi\}$$

Logic:	Upper bound:	
$\varphi \in \mathrm{FO}$	$\mathrm{L}(\varphi) \in \mathbf{\Sigma}^0_n$	(finite level of the Borel hierarchy)
$\varphi \in \mathrm{WMSO}$	$\mathrm{L}(\varphi) \in \mathbf{\Sigma}^0_n$	(finite level of the Borel hierarchy)
$\varphi \in \exists\mathrm{MSO}$	$\mathrm{L}(\varphi) \in \mathbf{\Sigma}^1_1$	(analytic set)
$\varphi \in \mathrm{MSO}$	$\mathrm{L}(\varphi) \in \mathbf{\Sigma}^1_n$	(finite level of the projective hierarchy)

using automata:

Upper bounds

Syntax:	Set theory:
$a(x)$	basic open set
$\exists_x$, $\exists^{\mathsf{fin}}_X$	countable union
$\exists_X$	projection from $A \times \{0,1\}$ to A
$\neg$	complementation

$$\mathrm{L}(\varphi) \stackrel{\mathsf{def}}{=} \{M \mid M \models \varphi\}$$

Logic:	Upper bound:	
$\varphi \in \mathrm{FO}$	$\mathrm{L}(\varphi) \in \boldsymbol{\Sigma}^0_n$	(finite level of the Borel hierarchy)
$\varphi \in \mathrm{WMSO}$	$\mathrm{L}(\varphi) \in \boldsymbol{\Sigma}^0_n$	(finite level of the Borel hierarchy)
$\varphi \in \exists\mathrm{MSO}$	$\mathrm{L}(\varphi) \in \boldsymbol{\Sigma}^1_1$	(analytic set)
$\varphi \in \mathrm{MSO}$	$\mathrm{L}(\varphi) \in \boldsymbol{\Sigma}^1_n$	(finite level of the projective hierarchy)

using automata:

$\varphi \in \mathrm{MSO}$	$\mathrm{L}(\varphi) \in \boldsymbol{\Sigma}^1_2 \cap \boldsymbol{\Pi}^1_2$	(second level of the projective hierarchy)

Lower bounds (infinite trees)

Lower bounds (infinite trees)

Theorem (Niwiński [1985])

There exists a $\mathbf{\Sigma}^1_1$-complete language definable in $\exists\textsc{mso}$.

Lower bounds (infinite trees)

Theorem (Niwiński [1985])

There exists a $\mathbf{\Sigma}^1_1$-complete language definable in $\exists$MSO.

Proof

Lower bounds (infinite trees)

Theorem (Niwiński [1985])

There exists a $\mathbf{\Sigma}^1_1$-complete language definable in $\exists\mathrm{MSO}$.

Proof

Infinite chain of a:

$$\varphi = \exists X.\ \big(\exists x.\ x \in X\big) \land \big(\forall x.\ x \in X \Rightarrow a(x)\big) \land \forall x.\ x \in X \Rightarrow\ \exists y.\ x \prec y\ \land\ y \in X$$

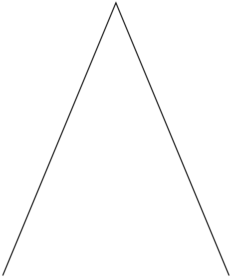
Theorem (Niwiński [1985])

There exists a $\boldsymbol{\Sigma}^1_1$-complete language definable in $\exists\text{MSO}$.

Proof

Infinite chain of a:

$$\varphi = \exists X.\ \big(\exists x.\ x \in X\big) \land \big(\forall x.\ x \in X \Rightarrow a(x)\big) \land \forall x.\ x \in X \Rightarrow \exists y.\ x \prec y \ \land\ y \in X$$



Lower bounds (infinite trees)

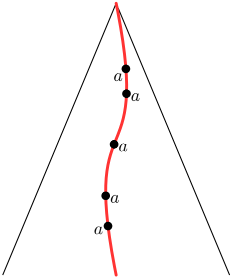
Theorem (Niwiński [1985])

There exists a $\mathbf{\Sigma}_1^1$-complete language definable in $\exists$MSO.

Proof

Infinite chain of a:

$$\varphi = \exists X.\ \big(\exists x.\ x \in X\big) \wedge \big(\forall x.\ x \in X \Rightarrow a(x)\big) \wedge \forall x.\ x \in X \Rightarrow \ \exists y.\ x \prec y \ \wedge \ y \in X$$



Lower bounds (infinite trees)

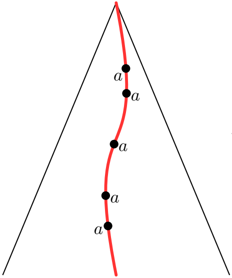
Theorem (Niwiński [1985])

There exists a $\mathbf{\Sigma}^1_1$-complete language definable in $\exists$MSO.

Proof

Infinite chain of a:

$$\varphi = \exists X.\ \big(\exists x.\ x \in X\big) \wedge \big(\forall x.\ x \in X \Rightarrow a(x)\big) \wedge \forall x.\ x \in X \Rightarrow\ \exists y.\ x \prec y\ \wedge\ y \in X$$



$+$ a continuous reduction from IF to $\mathrm{L}(\varphi)$

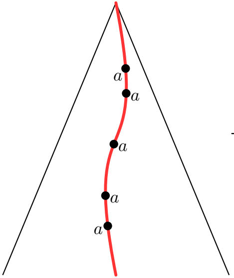
Theorem (Niwiński [1985])

There exists a $\mathbf{\Sigma}^1_1$-complete language definable in $\exists\mathrm{MSO}$.

Proof

Infinite chain of a:

$$\varphi = \exists X.\ \big(\exists x.\ x \in X\big) \land \big(\forall x.\ x \in X \Rightarrow a(x)\big) \land \forall x.\ x \in X \Rightarrow\ \exists y.\ x \prec y\ \land\ y \in X$$



$+$ a continuous reduction from IF to $\mathrm{L}(\varphi)$ ■

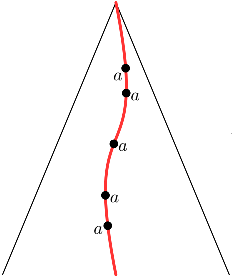
Theorem (Niwiński [1985])

There exists a $\boldsymbol{\Sigma}^1_1$-complete language definable in $\exists\text{MSO}$.

Proof

Infinite chain of a:

$$\varphi = \exists X.\ \big(\exists x.\ x \in X\big) \land \big(\forall x.\ x \in X \Rightarrow a(x)\big) \land \forall x.\ x \in X \Rightarrow\ \exists y.\ x \prec y\ \land\ y \in X$$



$+$ a continuous reduction from IF to $\mathrm{L}(\varphi)$ ■

$\rightsquigarrow$ $\mathrm{L}(\varphi)$ is non-Borel

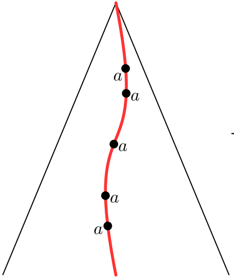
Theorem (Niwiński [1985])

There exists a $\mathbf{\Sigma}^1_1$-complete language definable in $\exists$MSO.

Proof

Infinite chain of a:

$$\varphi = \exists X.\ \big(\exists x.\ x \in X\big) \wedge \big(\forall x.\ x \in X \Rightarrow a(x)\big) \wedge \forall x.\ x \in X \Rightarrow\ \exists y.\ x \prec y\ \wedge\ y \in X$$



$+$ a continuous reduction from IF to $\mathrm{L}(\varphi)$ ■

$\rightsquigarrow$ $\mathrm{L}(\varphi)$ is non-Borel

$\rightsquigarrow$ $\mathrm{L}(\varphi)$ is **not** WMSO-definable

Lower bounds (infinite trees)

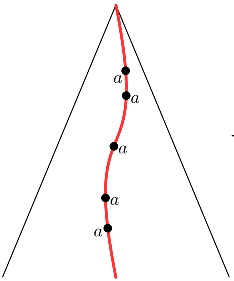
Theorem (Niwiński [1985])

There exists a $\mathbf{\Sigma}^1_1$-complete language definable in $\exists$MSO.

Proof

Infinite chain of a:

$$\varphi = \exists X.\ \big(\exists x.\ x \in X\big) \land \big(\forall x.\ x \in X \Rightarrow a(x)\big) \land \forall x.\ x \in X \Rightarrow\ \exists y.\ x \prec y\ \land\ y \in X$$



$+$ a continuous reduction from IF to $\mathrm{L}(\varphi)$ ∎

⇝ $\mathrm{L}(\varphi)$ is non-Borel

⇝ $\mathrm{L}(\varphi)$ is **not** WMSO-definable

⇝ $\mathrm{L}(\neg\varphi)$ is **not** $\exists$MSO-definable

Lower bounds (infinite trees)

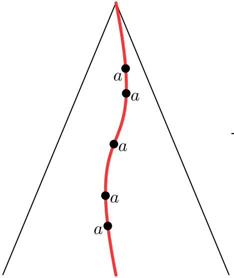
Theorem (Niwiński [1985])

There exists a $\mathbf{\Sigma}^1_1$-complete language definable in $\exists$MSO.

Proof

Infinite chain of a:

$$\varphi = \exists X.\ \big(\exists x.\ x \in X\big) \wedge \big(\forall x.\ x \in X \Rightarrow a(x)\big) \wedge \forall x.\ x \in X \Rightarrow \ \exists y.\ x \prec y \ \wedge \ y \in X$$



$+$ a continuous reduction from IF to $\mathrm{L}(\varphi)$ ∎

⇝ $\mathrm{L}(\varphi)$ is non-Borel

⇝ $\mathrm{L}(\varphi)$ is **not** WMSO-definable

⇝ $\mathrm{L}(\neg\varphi)$ is **not** $\exists$MSO-definable

$$\mathrm{FO} \subsetneq \mathrm{WMSO} \subsetneq \exists\mathrm{MSO} \subsetneq \mathrm{MSO}$$

Correspondence

Correspondence

Conjecture (Skurczyński [1993])

If $\mathrm{L}(\varphi)$ is Borel then $\mathrm{L}(\varphi)$ is WMSO-definable.

Correspondence

Conjecture (Skurczyński [1993])

If $\mathrm{L}(\varphi)$ is Borel then $\mathrm{L}(\varphi)$ is WMSO-definable.

$\rightsquigarrow$ a gap property:

if MSO-definable language is Borel then it is $\mathbf{\Sigma}^1_n$ for some n

Correspondence

Conjecture (Skurczyński [1993])

If $\mathrm{L}(\varphi)$ is Borel then $\mathrm{L}(\varphi)$ is WMSO-definable.

$\rightsquigarrow$ a gap property:

if MSO-definable language is Borel then it is $\mathbf{\Sigma}^1_n$ for some n

Theorem (Niwiński, Walukiewicz [2003])

Skurczyński's conjecture for deterministic automata (stronger gap).

Correspondence

Conjecture (Skurczyński [1993])

If $\mathrm{L}(\varphi)$ is Borel then $\mathrm{L}(\varphi)$ is WMSO-definable.

$\rightsquigarrow$ a gap property:

if MSO-definable language is Borel then it is $\boldsymbol{\Sigma}^1_n$ for some n

Theorem (Niwiński, Walukiewicz [2003])

Skurczyński's conjecture for deterministic automata (stronger gap).

Theorem (Facchini, Murlak, S. [2013])

Skurczyński's conjecture for game automata.

Correspondence

Conjecture (Skurczyński [1993])
If $\mathrm{L}(\varphi)$ is Borel then $\mathrm{L}(\varphi)$ is $\textsc{wmso}$-definable.

$\rightsquigarrow$ a gap property:

if $\textsc{mso}$-definable language is Borel then it is $\mathbf{\Sigma}^1_n$ for some n

Theorem (Niwiński, Walukiewicz [2003])
Skurczyński's conjecture for deterministic automata (stronger gap).

Theorem (Facchini, Murlak, S. [2013])
Skurczyński's conjecture for game automata.

Theorem (S., Walukiewicz [2016])
Skurczyński's conjecture for $\exists\textsc{mso}$.

Correspondence

Conjecture (Skurczyński [1993])
If $\mathrm{L}(\varphi)$ is Borel then $\mathrm{L}(\varphi)$ is WMSO-definable.

$\rightsquigarrow$ a gap property:

if MSO-definable language is Borel then it is $\mathbf{\Sigma}^1_n$ for some n

Theorem (Niwiński, Walukiewicz [2003])
Skurczyński's conjecture for deterministic automata (stronger gap).

Theorem (Facchini, Murlak, S. [2013])
Skurczyński's conjecture for game automata.

Theorem (S., Walukiewicz [2016])
Skurczyński's conjecture for $\exists$MSO.

Open for full MSO :(

Lack of correspondence

Lack of correspondence

Theorem (Milewski, Niwiński [2008?])

There is an MSO-definable language in $\mathbf{\Sigma}_1^1$ that is **not** $\exists$MSO-definable.

Lack of correspondence

Theorem (Milewski, Niwiński [2008?])

There is an MSO-definable language in $\mathbf{\Sigma}_1^1$ that is **not** $\exists$MSO-definable.

Proof

Either 0 or $\geqslant 2$ branches with infinitely many a.

Lack of correspondence

Theorem (Milewski, Niwiński [2008?])

There is an MSO-definable language in $\mathbf{\Sigma}_1^1$ that is **not** $\exists$MSO-definable.

Proof

Either 0 or $\geqslant 2$ branches with infinitely many a.

- The language is $\mathbf{\Sigma}_1^1$ by Lusin.

Lack of correspondence

Theorem (Milewski, Niwiński [2008?])

There is an MSO-definable language in $\mathbf{\Sigma}^1_1$ that is **not** $\exists$MSO-definable.

Proof

Either 0 or $\geqslant 2$ branches with infinitely many a.

- The language is $\mathbf{\Sigma}^1_1$ by Lusin. (the proof is not automata-like)

Lack of correspondence

Theorem (Milewski, Niwiński [2008?])

There is an MSO-definable language in $\mathbf{\Sigma}_1^1$ that is **not** $\exists$MSO-definable.

Proof

Either 0 or $\geqslant 2$ branches with infinitely many a.

- The language is $\mathbf{\Sigma}_1^1$ by Lusin. (the proof is not automata-like)
- The language is **not** $\exists$MSO-definable by pumping.

Lack of correspondence

Theorem (Milewski, Niwiński [2008?])

There is an MSO-definable language in $\mathbf{\Sigma}_1^1$ that is **not** $\exists$MSO-definable.

Proof

Either 0 or $\geqslant 2$ branches with infinitely many a.

- The language is $\mathbf{\Sigma}_1^1$ by Lusin. (the proof is not automata-like)
- The language is **not** $\exists$MSO-definable by pumping. ■

Lack of correspondence

Theorem (Milewski, Niwiński [2008?])

There is an MSO-definable language in $\boldsymbol{\Sigma}^1_1$ that is **not** $\exists$MSO-definable.

Proof

Either 0 or $\geqslant 2$ branches with infinitely many a.

- The language is $\boldsymbol{\Sigma}^1_1$ by Lusin. (the proof is not automata-like)
- The language is **not** $\exists$MSO-definable by pumping. ■

Correspondences for more complicated sets in $\boldsymbol{\Sigma}^1_2 \cap \boldsymbol{\Pi}^1_2$: **OPEN**

Topological complexity vs. decidability

Topological complexity vs. decidability

Theorem (Shelah [1975]; Gurevich, Shelah [1982])

The MSO-theory of $(\mathbb{R}, \leqslant)$ is undecidable.

Topological complexity vs. decidability

Theorem (Shelah [1975]; Gurevich, Shelah [1982])
The MSO-theory of $(\mathbb{R}, \leqslant)$ is undecidable.

Theorem (Bojańczyk, Gogacz, Michalewski, S. [2014])
If $\mathcal{L}$ is an extension of MSO such that:

Topological complexity vs. decidability

Theorem (Shelah [1975]; Gurevich, Shelah [1982])
The MSO-theory of $(\mathbb{R}, \leqslant)$ is undecidable.

Theorem (Bojańczyk, Gogacz, Michalewski, S. [2014])
If $\mathcal{L}$ is an extension of MSO such that:

$\mathcal{L}$ defines $\mathbf{\Sigma}^1_6$-complete languages in A^ω

Topological complexity vs. decidability

Theorem (Shelah [1975]; Gurevich, Shelah [1982])
The MSO-theory of $(\mathbb{R}, \leqslant)$ is undecidable.

Theorem (Bojańczyk, Gogacz, Michalewski, S. [2014])
If $\mathcal{L}$ is an extension of MSO such that:

$\mathcal{L}$ defines $\mathbf{\Sigma}^1_6$-complete languages in A^ω

then it is consistent with ZFC that:

Topological complexity vs. decidability

Theorem (Shelah [1975]; Gurevich, Shelah [1982])
The MSO-theory of $(\mathbb{R}, \leqslant)$ is undecidable.

Theorem (Bojańczyk, Gogacz, Michalewski, S. [2014])
If $\mathcal{L}$ is an extension of MSO such that:

$$\mathcal{L} \text{ defines } \boldsymbol{\Sigma}^1_6\text{-complete languages in } A^\omega$$

then it is consistent with ZFC that:

$$\text{the } \mathcal{L}\text{-theory of } \left(\{\mathtt{L}, \mathtt{R}\}^*, \leqslant_{\mathrm{lex}}, \preceq\right) \text{ is undecidable.}$$

Topological complexity vs. decidability

Theorem (Shelah [1975]; Gurevich, Shelah [1982])
The MSO-theory of $(\mathbb{R}, \leqslant)$ is undecidable.

Theorem (Bojańczyk, Gogacz, Michalewski, S. [2014])
If $\mathcal{L}$ is an extension of MSO such that:

$\mathcal{L}$ defines $\mathbf{\Sigma}^1_6$-complete languages in A^ω

then it is consistent with ZFC that:

the $\mathcal{L}$-theory of $\big(\{\mathtt{L}, \mathtt{R}\}^*, \leqslant_{\mathrm{lex}}, \preceq\big)$ is undecidable.

Proof
Use Shelah's argument in the constructible universe of Gödel (V=L).

Topological complexity vs. decidability

Theorem (Shelah [1975]; Gurevich, Shelah [1982])

The MSO-theory of $(\mathbb{R}, \leqslant)$ is undecidable.

Theorem (Bojańczyk, Gogacz, Michalewski, S. [2014])

If $\mathcal{L}$ is an extension of MSO such that:

$\mathcal{L}$ defines $\boldsymbol{\Sigma}^1_6$-complete languages in A^ω

then it is consistent with ZFC that:

the $\mathcal{L}$-theory of $\big(\{\mathtt{L}, \mathtt{R}\}^*, \leqslant_{\mathrm{lex}}, \preceq\big)$ is undecidable.

Proof

Use Shelah's argument in the constructible universe of Gödel $(\mathrm{V}{=}\mathrm{L})$.

■

Applications to MSO+U

Applications to MSO+U

$$\mathsf{U}X.\ \varphi(X) \quad \equiv \quad \forall n.\ \exists X.\ \varphi(X)\ \wedge\ n < |X| < \infty.$$

Applications to MSO+U

$$\mathsf{U}X.\ \varphi(X) \quad \equiv \quad \forall n.\ \exists X.\ \varphi(X)\ \wedge\ n < |X| < \infty.$$

$\varphi(X)$ holds for arbitrarily big finite sets

Applications to MSO+U

$$\mathsf{U}X.\ \varphi(X) \quad\equiv\quad \forall n.\ \exists X.\ \varphi(X)\ \wedge\ n < |X| < \infty.$$

$\varphi(X)$ holds for arbitrarily big finite sets

Large expressive power: cost functions, distance automata, ...

Applications to MSO+U

$$\mathsf{U}X.\ \varphi(X) \quad \equiv \quad \forall n.\ \exists X.\ \varphi(X)\ \wedge\ n < |X| < \infty.$$

$\varphi(X)$ holds for arbitrarily big finite sets

Applications to MSO+U

$$\mathsf{U}X.\ \varphi(X) \quad \equiv \quad \forall n.\ \exists X.\ \varphi(X)\ \wedge\ n < |X| < \infty.$$

$\varphi(X)$ holds for arbitrarily big finite sets

Example

The delays between REQUEST and RESPONSE are uniformly bounded.

Applications to MSO+U

$$\mathsf{U}X.\ \varphi(X) \quad \equiv \quad \forall n.\ \exists X.\ \varphi(X)\ \wedge\ n < |X| < \infty.$$

$\varphi(X)$ holds for arbitrarily big finite sets

Example

The delays between REQUEST and RESPONSE are uniformly bounded.

Conjecture (Bojańczyk [2004])

The MSO+U theory of $(\omega, \leqslant)$ is decidable.

Applications to MSO+U

$$\mathsf{U}X.\ \varphi(X) \quad \equiv \quad \forall n.\ \exists X.\ \varphi(X)\ \wedge\ n < |X| < \infty.$$

$\varphi(X)$ holds for arbitrarily big finite sets

Example

The delays between REQUEST and RESPONSE are uniformly bounded.

Conjecture (Bojańczyk [2004])

The MSO+U theory of $(\omega, \leqslant)$ is decidable.

Theorem (Bojańczyk, Gogacz, Michalewski, S. [2014])

It is consistent with ZFC that

the MSO+U-theory of $\big(\{\mathtt{L},\mathtt{R}\}^*, \leqslant_{\mathrm{lex}}, \preceq\big)$ is undecidable.

Applications to MSO+U

$$\mathsf{U}X.\ \varphi(X) \quad \equiv \quad \forall n.\ \exists X.\ \varphi(X)\ \wedge\ n < |X| < \infty.$$

$\varphi(X)$ holds for arbitrarily big finite sets

Example

The delays between REQUEST and RESPONSE are uniformly bounded.

Conjecture (Bojańczyk [2004])

The MSO+U theory of $(\omega, \leqslant)$ is decidable.

Theorem (Bojańczyk, Gogacz, Michalewski, S. [2014])

It is consistent with ZFC that

the MSO+U-theory of $\left(\{\mathtt{L}, \mathtt{R}\}^*, \leqslant_{\mathrm{lex}}, \preceq\right)$ is undecidable.

Theorem (Bojańczyk, Parys, Toruńczyk [2016])

The MSO+U-theory of $\left(\omega, \leqslant\right)$ is undecidable.

Alternation hierarchy of modal μ-calculus

Alternation hierarchy of modal μ-calculus

$\cong$

Alternation hierarchy of modal μ-calculus

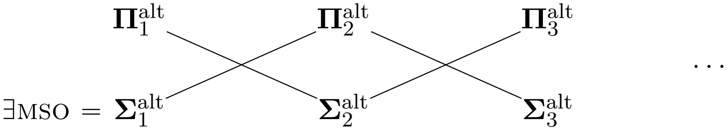
$\wr\!|\!|$

Index hierarchy for alternating parity automata

Alternation hierarchy of modal μ-calculus

$\cong$

Index hierarchy for alternating parity automata



Alternation hierarchy of modal μ-calculus

$$\cong$$

Index hierarchy for alternating parity automata

Alternation hierarchy of modal μ-calculus

$\cong$

Index hierarchy for alternating parity automata

Languages $W_{i,k}$ (Walukiewicz [1996])

Alternation hierarchy of modal μ-calculus

$\wr|$

Index hierarchy for alternating parity automata

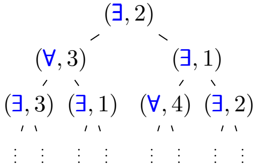
Languages $W_{i,k}$ (Walukiewicz [1996]) $A_{i,k} = \{\exists, \forall\} \times \{i, i+1, \ldots, k\}$

Alternation hierarchy of modal μ-calculus

$\cong$

Index hierarchy for alternating parity automata

Languages $W_{i,k}$ (Walukiewicz [1996]) $A_{i,k} = \{\exists, \forall\} \times \{i, i+1, \ldots, k\}$

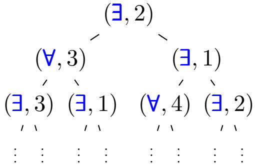


Alternation hierarchy of modal μ-calculus

$\cong$

Index hierarchy for alternating parity automata

Languages $W_{i,k}$ (Walukiewicz [1996]) $\quad A_{i,k} = \{\exists, \forall\} \times \{i, i+1, \ldots, k\}$



$t \in W_{i,k}$ if $\exists$ wins $\mathcal{G}_t$

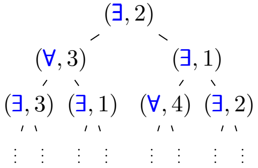
Alternation hierarchy of modal μ-calculus

$\cong$

Index hierarchy for alternating parity automata

Languages $W_{i,k}$ (Walukiewicz [1996]) $\quad A_{i,k} = \{\exists, \forall\} \times \{i, i+1, \ldots, k\}$

If $L \in \mathbf{\Pi}_n^{\mathrm{alt}}$ then



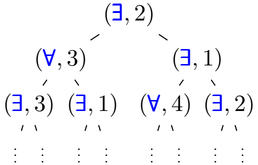
$t \in W_{i,k}$ if $\exists$ wins $\mathcal{G}_t$

Alternation hierarchy of modal μ-calculus

$\wr\parallel$

Index hierarchy for alternating parity automata

Languages $W_{i,k}$ (Walukiewicz [1996]) $A_{i,k} = \{\exists, \forall\} \times \{i, i+1, \ldots, k\}$



$t \in W_{i,k}$ if $\exists$ wins $\mathcal{G}_t$

If $L \in \mathbf{\Pi}_n^{\mathrm{alt}}$ then

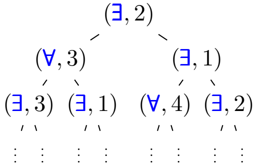
L $\frac{1}{2}$-Lipschitz reduces to $W_{0,n}$

Alternation hierarchy of modal μ-calculus

$\wr\!\!\|$

Index hierarchy for alternating parity automata

Languages $W_{i,k}$ (Walukiewicz [1996]) $A_{i,k} = \{\exists, \forall\} \times \{i, i+1, \ldots, k\}$



If $L \in \mathbf{\Pi}_n^{\mathrm{alt}}$ then

L $\frac{1}{2}$-Lipschitz reduces to $W_{0,n}$

If $L \in \mathbf{\Sigma}_n^{\mathrm{alt}}$ then

L $\frac{1}{2}$-Lipschitz reduces to $W_{1,n+1}$

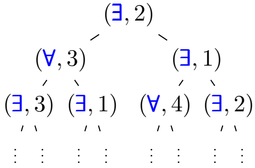
$t \in W_{i,k}$ if $\exists$ wins $\mathcal{G}_t$

Alternation hierarchy of modal μ-calculus

$\cong$

Index hierarchy for alternating parity automata

Languages $W_{i,k}$ (Walukiewicz [1996]) $A_{i,k} = \{\exists, \forall\} \times \{i, i+1, \ldots, k\}$



$t \in W_{i,k}$ if $\exists$ wins $\mathcal{G}_t$

If $L \in \mathbf{\Pi}_n^{\mathrm{alt}}$ then

L $\frac{1}{2}$-Lipschitz reduces to $W_{0,n}$

If $L \in \mathbf{\Sigma}_n^{\mathrm{alt}}$ then

L $\frac{1}{2}$-Lipschitz reduces to $W_{1,n+1}$

$W_{0,n}$ 1-Lischitz reduces to $W^{\mathrm{c}}_{1,n+1}$

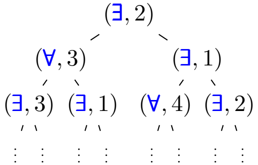
$W_{1,n+1}$ 1-Lischitz reduces to $W^{\mathrm{c}}_{0,n}$

Alternation hierarchy of modal μ-calculus

$\cong$

Index hierarchy for alternating parity automata

Languages $W_{i,k}$ (Walukiewicz [1996]) $\quad A_{i,k} = \{\exists, \forall\} \times \{i, i+1, \ldots, k\}$



$t \in W_{i,k}$ if $\exists$ wins $\mathcal{G}_t$

If $L \in \mathbf{\Pi}_n^{\mathrm{alt}}$ then

L $\frac{1}{2}$-Lipschitz reduces to $W_{0,n}$

If $L \in \mathbf{\Sigma}_n^{\mathrm{alt}}$ then

L $\frac{1}{2}$-Lipschitz reduces to $W_{1,n+1}$

$W_{0,n}$ 1-Lischitz reduces to $W_{1,n+1}^{\mathrm{c}}$ $\quad W_{1,n+1}$ 1-Lischitz reduces to $W_{0,n}^{\mathrm{c}}$

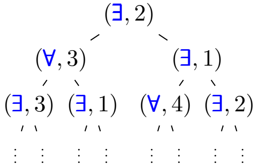
Theorem (Arnold [1999]) $W_{0,n} \notin \mathbf{\Sigma}_n^{\mathrm{alt}}$

Alternation hierarchy of modal μ-calculus

$\cong$

Index hierarchy for alternating parity automata

Languages $W_{i,k}$ (Walukiewicz [1996]) $A_{i,k} = \{\exists, \forall\} \times \{i, i+1, \dots, k\}$



$t \in W_{i,k}$ if $\exists$ wins $\mathcal{G}_t$

If $L \in \mathbf{\Pi}_n^{\mathrm{alt}}$ then

L $\frac{1}{2}$-Lipschitz reduces to $W_{0,n}$

If $L \in \mathbf{\Sigma}_n^{\mathrm{alt}}$ then

L $\frac{1}{2}$-Lipschitz reduces to $W_{1,n+1}$

$W_{0,n}$ 1-Lischitz reduces to $W_{1,n+1}^{\mathrm{c}}$ $W_{1,n+1}$ 1-Lischitz reduces to $W_{0,n}^{\mathrm{c}}$

Theorem (Arnold [1999]) $W_{0,n} \notin \mathbf{\Sigma}_n^{\mathrm{alt}}$

Proof

If $W_{0,n} \in \mathbf{\Sigma}_n^{\mathrm{alt}}$ then by Banach's fix-point theorem $W_{0,n} \cap W_{0,n}^{\mathrm{c}} \neq \emptyset$.

∎

Summary

Summary

Topological complexity can measure expressive power of logics

Summary

Topological complexity can measure expressive power of logics

$\rightsquigarrow$ simple and nice hardness arguments

Summary

Topological complexity can measure expressive power of logics

$\rightsquigarrow$ simple and nice hardness arguments

$\rightsquigarrow$ undefinability and undecidability results

Summary

Topological complexity can measure expressive power of logics

- $\rightsquigarrow$ simple and nice hardness arguments
- $\rightsquigarrow$ undefinability and undecidability results

In natural cases we have correspondence

Summary

Topological complexity can measure expressive power of logics

- $\rightsquigarrow$ simple and nice hardness arguments
- $\rightsquigarrow$ undefinability and undecidability results

In natural cases we have correspondence (but not always...)

Summary

Topological complexity can measure expressive power of logics

$\rightsquigarrow$ simple and nice hardness arguments

$\rightsquigarrow$ undefinability and undecidability results

In natural cases we have correspondence (but not always...)

Methods from descriptive set theory:

Summary

Topological complexity can measure expressive power of logics

$\rightsquigarrow$ simple and nice hardness arguments

$\rightsquigarrow$ undefinability and undecidability results

In natural cases we have correspondence (but not always...)

Methods from descriptive set theory:

$\rightsquigarrow$ continuous reductions

Summary

Topological complexity can measure expressive power of logics

- ⇝ simple and nice hardness arguments
- ⇝ undefinability and undecidability results

In natural cases we have correspondence (but not always. . .)

Methods from descriptive set theory:

- ⇝ continuous reductions
- ⇝ separation arguments

Summary

Topological complexity can measure expressive power of logics

- $\rightsquigarrow$ simple and nice hardness arguments
- $\rightsquigarrow$ undefinability and undecidability results

In natural cases we have correspondence (but not always...)

Methods from descriptive set theory:

- $\rightsquigarrow$ continuous reductions
- $\rightsquigarrow$ separation arguments
- $\rightsquigarrow$ ranks

Summary

Topological complexity can measure expressive power of logics

- ⇝ simple and nice hardness arguments
- ⇝ undefinability and undecidability results

In natural cases we have correspondence (but not always. . .)

Methods from descriptive set theory:

- ⇝ continuous reductions
- ⇝ separation arguments
- ⇝ ranks
- ⇝ games and determinacy

Summary

Topological complexity can measure expressive power of logics

- ⇝ simple and nice hardness arguments
- ⇝ undefinability and undecidability results

In natural cases we have correspondence (but not always...)

Methods from descriptive set theory:

- ⇝ continuous reductions
- ⇝ separation arguments
- ⇝ ranks
- ⇝ games and determinacy
- ⇝ measurability and Baire Property

Summary

Topological complexity can measure expressive power of logics

- ⇝ simple and nice hardness arguments
- ⇝ undefinability and undecidability results

In natural cases we have correspondence (but not always...)

Methods from descriptive set theory:

- ⇝ continuous reductions
- ⇝ separation arguments
- ⇝ ranks
- ⇝ games and determinacy
- ⇝ measurability and Baire Property
- ⇝ ...