Wyrażalność w logice, topologia i automaty

Michał Skrzypczak

Instytut Informatyki, Uniwersytet Warszawski

Wręczenie Nagrody im. W. Lipskiego

13.10.2016

Definiowalność w logice

Definiowalność w logice

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

Definiowalność w logice

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

[jeżeli $\mathrm{L}(\varphi) = \mathrm{L}(\psi)$, to uważamy, że $\varphi \equiv \psi$]

Definiowalność w logice

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

[jeżeli $\mathrm{L}(\varphi) = \mathrm{L}(\psi)$, to uważamy, że $\varphi \equiv \psi$]

- gęste porządki liniowe

Definiowalność w logice

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

[jeżeli $\mathrm{L}(\varphi) = \mathrm{L}(\psi)$, to uważamy, że $\varphi \equiv \psi$]

- gęste porządki liniowe
- grafy z cyklem Eulera

Definiowalność w logice

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

[jeżeli $\mathrm{L}(\varphi) = \mathrm{L}(\psi)$, to uważamy, że $\varphi \equiv \psi$]

- gęste porządki liniowe
- grafy z cyklem Eulera
- grafy z cyklem Hamiltona

Definiowalność w logice

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

[jeżeli $\mathrm{L}(\varphi) = \mathrm{L}(\psi)$, to uważamy, że $\varphi \equiv \psi$]

- gęste porządki liniowe
- grafy z cyklem Eulera
- grafy z cyklem Hamiltona
- ...

Definiowalność w logice

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

[jeżeli $\mathrm{L}(\varphi) = \mathrm{L}(\psi)$, to uważamy, że $\varphi \equiv \psi$]

- gęste porządki liniowe
- grafy z cyklem Eulera
- grafy z cyklem Hamiltona
- ...

Zrozumieć siłę wyrazu $\mathcal{L}$

Definiowalność w logice

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

[jeżeli $\mathrm{L}(\varphi) = \mathrm{L}(\psi)$, to uważamy, że $\varphi \equiv \psi$]

- gęste porządki liniowe
- grafy z cyklem Eulera
- grafy z cyklem Hamiltona
- ...

Zrozumieć siłę wyrazu $\mathcal{L}$

|||

Definiowalność w logice

$$\varphi \in \mathcal{L} \qquad \rightsquigarrow \qquad \mathrm{L}(\varphi) \overset{\text{def}}{=} \{M \in \mathcal{M} \mid M \models \varphi\}$$

[jeżeli $\mathrm{L}(\varphi) = \mathrm{L}(\psi)$, to uważamy, że $\varphi \equiv \psi$]

- gęste porządki liniowe
- grafy z cyklem Eulera
- grafy z cyklem Hamiltona
- ...

Zrozumieć siłę wyrazu $\mathcal{L}$

$|||$

Określić, które zbiory $L \subseteq \mathcal{M}$ są postaci $L = \mathrm{L}(\varphi)$ dla $\varphi \in \mathcal{L}$

[„są $\mathcal{L}$-definiowalne"]

Definiowalność w logice

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

[jeżeli $\mathrm{L}(\varphi) = \mathrm{L}(\psi)$, to uważamy, że $\varphi \equiv \psi$]

- gęste porządki liniowe
- grafy z cyklem Eulera
- grafy z cyklem Hamiltona
- ...

Zrozumieć siłę wyrazu $\mathcal{L}$

|||

Określić, które zbiory $L \subseteq \mathcal{M}$ są postaci $L = \mathrm{L}(\varphi)$ dla $\varphi \in \mathcal{L}$

[„są $\mathcal{L}$-definiowalne"]

$\mathcal{L}$ jest silniejsza od $\mathcal{L}'$ jeśli $\mathcal{L}$ definiuje więcej zbiorów niż $\mathcal{L}'$

Definiowalność w logice

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

[jeżeli $\mathrm{L}(\varphi) = \mathrm{L}(\psi)$, to uważamy, że $\varphi \equiv \psi$]

- gęste porządki liniowe $\in$ FO (logika pierwszego rzędu)
- grafy z cyklem Eulera
- grafy z cyklem Hamiltona
- ...

Zrozumieć siłę wyrazu $\mathcal{L}$

$|||$

Określić, które zbiory $L \subseteq \mathcal{M}$ są postaci $L = \mathrm{L}(\varphi)$ dla $\varphi \in \mathcal{L}$

[„są $\mathcal{L}$-definiowalne”]

$\mathcal{L}$ jest silniejsza od $\mathcal{L}'$ jeśli $\mathcal{L}$ definiuje więcej zbiorów niż $\mathcal{L}'$

Definiowalność w logice

$$\varphi \in \mathcal{L} \qquad \rightsquigarrow \qquad \mathrm{L}(\varphi) \overset{\text{def}}{=} \{M \in \mathcal{M} \mid M \models \varphi\}$$

[jeżeli $\mathrm{L}(\varphi) = \mathrm{L}(\psi)$, to uważamy, że $\varphi \equiv \psi$]

- gęste porządki liniowe $\in$ FO (logika pierwszego rzędu)
- grafy z cyklem Eulera $\in$ FO $+$ FIXPOINT $+ \leqslant$
- grafy z cyklem Hamiltona
- ...

Zrozumieć siłę wyrazu $\mathcal{L}$

$\parallel\!|$

Określić, które zbiory $L \subseteq \mathcal{M}$ są postaci $L = \mathrm{L}(\varphi)$ dla $\varphi \in \mathcal{L}$

[„są $\mathcal{L}$-definiowalne"]

$\mathcal{L}$ jest silniejsza od $\mathcal{L}'$ jeśli $\mathcal{L}$ definiuje więcej zbiorów niż $\mathcal{L}'$

Definiowalność w logice

$$\varphi \in \mathcal{L} \qquad \rightsquigarrow \qquad \mathrm{L}(\varphi) \overset{\text{def}}{=} \{M \in \mathcal{M} \mid M \models \varphi\}$$

[jeżeli $\mathrm{L}(\varphi) = \mathrm{L}(\psi)$, to uważamy, że $\varphi \equiv \psi$]

- gęste porządki liniowe $\in$ FO (logika pierwszego rzędu)
- grafy z cyklem Eulera $\in$ FO + FIXPOINT + $\leqslant$
- grafy z cyklem Hamiltona $\in$ SO (logika drugiego rzędu)
- ...

Zrozumieć siłę wyrazu $\mathcal{L}$

|||

Określić, które zbiory $L \subseteq \mathcal{M}$ są postaci $L = \mathrm{L}(\varphi)$ dla $\varphi \in \mathcal{L}$

[„są $\mathcal{L}$-definiowalne"]

$\mathcal{L}$ jest silniejsza od $\mathcal{L}'$ jeśli $\mathcal{L}$ definiuje więcej zbiorów niż $\mathcal{L}'$

Siła wyrazu logiki

Siła wyrazu logiki

- Warunki niedefiniowalności:

Siła wyrazu logiki

- Warunki niedefiniowalności:

  Np.: wśród skończonych porządków liniowych $\big(\{1, \ldots, n\}, \leqslant\big)$

  FO-niedefiniowalne jest, że $n \equiv 0 \pmod 2$

Siła wyrazu logiki

- Warunki niedefiniowalności: Gry Ehrenfeuchta-Fraïsse'go, ...

  Np.: wśród skończonych porządków liniowych $\big(\{1,\ldots,n\},\leqslant\big)$

  FO-niedefiniowalne jest, że $n \equiv 0 \pmod 2$

Siła wyrazu logiki

- Warunki niedefiniowalności: Gry Ehrenfeuchta-Fraïsse'go, ...

  Np.: wśród skończonych porządków liniowych $\big(\{1,\ldots,n\},\leqslant\big)$

  $\textsc{fo}$-niedefiniowalne jest, że $n \equiv 0 \pmod 2$

- Własności zbiorów definiowalnych:

Siła wyrazu logiki

- Warunki niedefiniowalności: Gry Ehrenfeuchta-Fraïsse'go, ...

  Np.: wśród skończonych porządków liniowych $\big(\{1,\ldots,n\},\leqslant\big)$

  FO-niedefiniowalne jest, że $n \equiv 0 \pmod 2$

- Własności zbiorów definiowalnych:

  Np.: jeśli L zawiera dowolnie duże skończone modele

  i jest FO-definiowalny, to zawiera model nieskończony

Siła wyrazu logiki

- Warunki niedefiniowalności: Gry Ehrenfeuchta-Fraïsse'go, ...

  Np.: wśród skończonych porządków liniowych $\big(\{1,\ldots,n\},\leqslant\big)$

  FO-niedefiniowalne jest, że $n \equiv 0 \pmod 2$

- Własności zbiorów definiowalnych: Twierdzenie o zwartości, ...

  Np.: jeśli L zawiera dowolnie duże skończone modele

  i jest FO-definiowalny, to zawiera model nieskończony

Siła wyrazu logiki

- Warunki niedefiniowalności: Gry Ehrenfeuchta-Fraïsse'go, ...

  Np.: wśród skończonych porządków liniowych $\big(\{1, \ldots, n\}, \leqslant\big)$

  FO-niedefiniowalne jest, że $n \equiv 0 \pmod 2$

- Własności zbiorów definiowalnych: Twierdzenie o zwartości, ...

  Np.: jeśli L zawiera dowolnie duże skończone modele

  i jest FO-definiowalny, to zawiera model nieskończony

- Konstruowanie formuł:

Siła wyrazu logiki

- Warunki niedefiniowalności: Gry Ehrenfeuchta-Fraïsse'go, ...

  Np.: wśród skończonych porządków liniowych $\big(\{1, \ldots, n\}, \leqslant\big)$

  $\textsc{fo}$-niedefiniowalne jest, że $n \equiv 0 \pmod 2$

- Własności zbiorów definiowalnych: Twierdzenie o zwartości, ...

  Np.: jeśli L zawiera dowolnie duże skończone modele

  i jest $\textsc{fo}$-definiowalny, to zawiera model nieskończony

- Konstruowanie formuł:

  Dany jest L, skonstruować takie $\varphi \in \mathcal{L}$, by $L = \mathrm{L}(\varphi)$

Siła wyrazu logiki

- Warunki niedefiniowalności: Gry Ehrenfeuchta-Fraïsse'go, ...

  Np.: wśród skończonych porządków liniowych $\big(\{1,\ldots,n\},\leqslant\big)$

  FO-niedefiniowalne jest, że $n \equiv 0 \pmod 2$

- Własności zbiorów definiowalnych: Twierdzenie o zwartości, ...

  Np.: jeśli L zawiera dowolnie duże skończone modele

  i jest FO-definiowalny, to zawiera model nieskończony

- Konstruowanie formuł: Gaifman, Hintikka, automaty, ...

  Dany jest L, skonstruować takie $\varphi \in \mathcal{L}$, by $L = \mathrm{L}(\varphi)$

Siła wyrazu logiki

- Warunki niedefiniowalności: Gry Ehrenfeuchta-Fraïsse'go, ...

  Np.: wśród skończonych porządków liniowych $\big(\{1, \ldots, n\}, \leqslant\big)$

  $\textsc{fo}$-niedefiniowalne jest, że $n \equiv 0 \pmod 2$

- Własności zbiorów definiowalnych: Twierdzenie o zwartości, ...

  Np.: jeśli L zawiera dowolnie duże skończone modele

  i jest $\textsc{fo}$-definiowalny, to zawiera model nieskończony

- Konstruowanie formuł: Gaifman, Hintikka, automaty, ...

  Dany jest L, skonstruować takie $\varphi \in \mathcal{L}$, by $L = \mathrm{L}(\varphi)$

- Złożoność zbiorów $\mathrm{L}(\varphi)$:

  Określić położenie zbiorów $\mathrm{L}(\varphi)$ w pewnej hierarchii

Siła wyrazu logiki

- Warunki niedefiniowalności: Gry Ehrenfeuchta-Fraïsse'go, ...

  Np.: wśród skończonych porządków liniowych $\big(\{1,\ldots,n\},\leqslant\big)$

  FO-niedefiniowalne jest, że $n \equiv 0 \pmod 2$

- Własności zbiorów definiowalnych: Twierdzenie o zwartości, ...

  Np.: jeśli L zawiera dowolnie duże skończone modele

  i jest FO-definiowalny, to zawiera model nieskończony

- Konstruowanie formuł: Gaifman, Hintikka, automaty, ...

  Dany jest L, skonstruować takie $\varphi \in \mathcal{L}$, by $L = \mathrm{L}(\varphi)$

- Złożoność zbiorów $\mathrm{L}(\varphi)$: Hier. borelowska, arytmetyczna, ...

  Określić położenie zbiorów $\mathrm{L}(\varphi)$ w pewnej hierarchii

Siła wyrazu logiki

- Warunki niedefiniowalności: Gry Ehrenfeuchta-Fraïsse'go, ...

  Np.: wśród skończonych porządków liniowych $\big(\{1,\ldots,n\},\leqslant\big)$

  $\textsc{fo}$-niedefiniowalne jest, że $n \equiv 0 \pmod 2$

- Własności zbiorów definiowalnych: Twierdzenie o zwartości, ...

  Np.: jeśli L zawiera dowolnie duże skończone modele

  i jest $\textsc{fo}$-definiowalny, to zawiera model nieskończony

- Konstruowanie formuł: Gaifman, Hintikka, automaty, ...

  Dany jest L, skonstruować takie $\varphi \in \mathcal{L}$, by $L = \mathrm{L}(\varphi)$

- Złożoność zbiorów $\mathrm{L}(\varphi)$: Hier. borelowska, arytmetyczna, ...

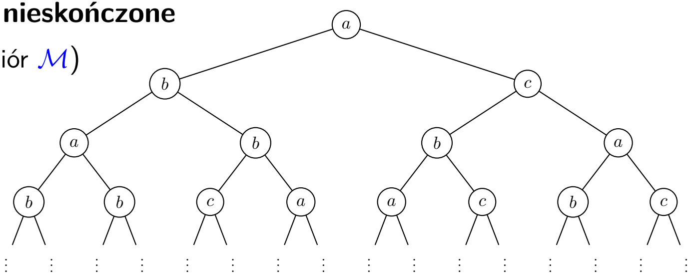
  Określić położenie zbiorów $\mathrm{L}(\varphi)$ w pewnej hierarchii

Drzewa nieskończone

(zbiór $\mathcal{M}$)

Drzewa nieskończone

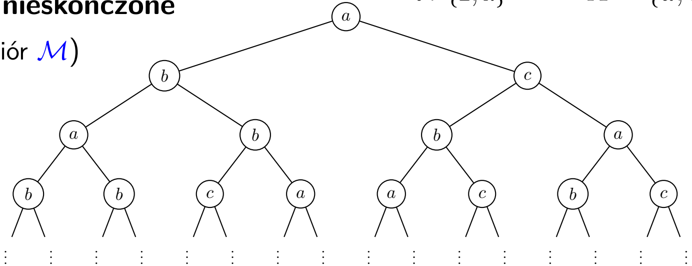
(zbiór $\mathcal{M}$)



Drzewa nieskończone

(zbiór $\mathcal{M}$)

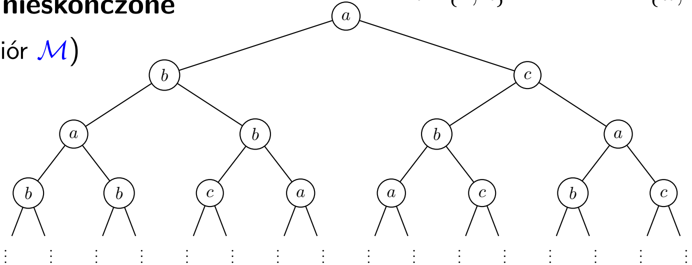
$t\colon \{\mathtt{L},\mathtt{R}\}^{<\omega} \to A = \{a, b, \ldots, e\}$



Drzewa nieskończone

(zbiór $\mathcal{M}$)

$t\colon \{\mathtt{L},\mathtt{R}\}^{<\omega} \to A = \{a, b, \ldots, e\}$

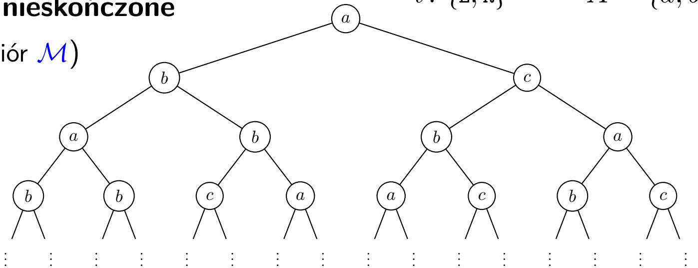


$$A^{(\{\mathtt{L},\mathtt{R}\}^{<\omega})}$$

Drzewa nieskończone

(zbiór $\mathcal{M}$)

$t\colon \{\mathtt{L},\mathtt{R}\}^{<\omega} \to A = \{a, b, \ldots, e\}$



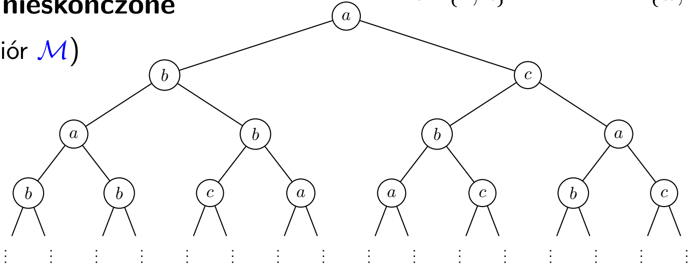
$$A^{(\{\mathtt{L},\mathtt{R}\}^{<\omega})}$$

$$\cong$$

Drzewa nieskończone

(zbiór $\mathcal{M}$)

$$t\colon \{\mathtt{L},\mathtt{R}\}^{<\omega} \to A = \{a, b, \dots, e\}$$



$$A^{(\{\mathtt{L},\mathtt{R}\}^{<\omega})}$$

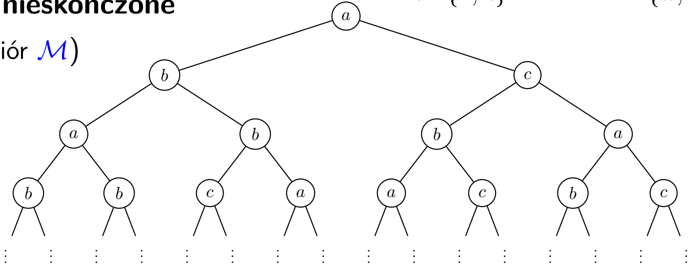
$$\cong$$

$$\{0,1\}^{\omega}$$

Drzewa nieskończone

(zbiór $\mathcal{M}$)

$t\colon \{\mathtt{L},\mathtt{R}\}^{<\omega} \to A = \{a, b, \ldots, e\}$

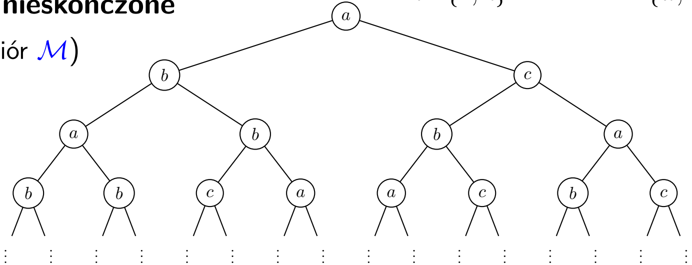


$$A^{(\{\mathtt{L},\mathtt{R}\}^{<\omega})}$$
$$\cong$$
$$\{0,1\}^{\omega}$$
$$\cong$$

Drzewa nieskończone

$$t\colon \{\mathtt{L},\mathtt{R}\}^{<\omega} \to A = \{a, b, \ldots, e\}$$

(zbiór $\mathcal{M}$)

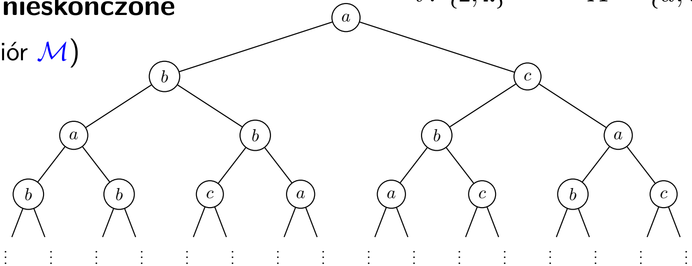


$$A^{(\{\mathtt{L},\mathtt{R}\}^{<\omega})}$$
$$\cong$$
$$\{0,1\}^{\omega}$$
$$\cong$$

Drzewa nieskończone

(zbiór $\mathcal{M}$)

$$t\colon \{\mathtt{L},\mathtt{R}\}^{<\omega} \to A = \{a, b, \ldots, e\}$$



$$A^{(\{\mathtt{L},\mathtt{R}\}^{<\omega})}$$

$$\cong$$

$$\{0,1\}^{\omega}$$

$$\cong$$

zbiór Cantora

Logiki

Logiki

Monadyczna logika drugiego rzędu (MSO):

Logiki

Monadyczna logika drugiego rzędu (MSO):

- $\vee$, $\wedge$, $\neg$ (spójniki logiczne)

Logiki

Monadyczna logika drugiego rzędu (MSO):

- $\vee$, $\wedge$, $\neg$ (spójniki logiczne)
- $\exists_x$ x — wierzchołek

Logiki

Monadyczna logika drugiego rzędu (MSO):

- $\vee$, $\wedge$, $\neg$ (spójniki logiczne)
- $\exists_x$ x — wierzchołek
- $\exists^{\mathsf{fin}}_X$ X — skończony zbiór wierzchołków

Logiki

Monadyczna logika drugiego rzędu (MSO):

- $\vee$, $\wedge$, $\neg$ (spójniki logiczne)
- $\exists_x$ x — wierzchołek
- $\exists_X^{\mathsf{fin}}$ X — skończony zbiór wierzchołków
- $\exists_X$ X — dowolny zbiór wierzchołków

Logiki

Monadyczna logika drugiego rzędu (MSO):

- $\vee$, $\wedge$, $\neg$ (spójniki logiczne)
- $\exists_x$ x — wierzchołek
- $\exists_X^{\mathsf{fin}}$ X — skończony zbiór wierzchołków
- $\exists_X$ X — dowolny zbiór wierzchołków
- $x \in X$, $x = y$, nawigacja w drzewie, etykiety

Logiki

Monadyczna logika drugiego rzędu (MSO):

- $\vee$, $\wedge$, $\neg$ (spójniki logiczne)
- $\exists_x$ x — wierzchołek
- $\exists^{\mathsf{fin}}_X$ X — skończony zbiór wierzchołków
- $\exists_X$ X — dowolny zbiór wierzchołków
- $x \in X$, $x = y$, nawigacja w drzewie, etykiety

Słaba monadyczna logika drugiego rzędu (WMSO):

Logiki

Monadyczna logika drugiego rzędu (MSO):

- $\vee$, $\wedge$, $\neg$ (spójniki logiczne)
- $\exists_x$ x — wierzchołek
- $\exists^{\mathsf{fin}}_X$ X — skończony zbiór wierzchołków
- $\exists_X$ X — dowolny zbiór wierzchołków
- $x \in X$, $x = y$, nawigacja w drzewie, etykiety

Słaba monadyczna logika drugiego rzędu (WMSO):

- tylko kwantyfikatory $\exists_x$ i $\exists^{\mathsf{fin}}_X$

Logiki

Monadyczna logika drugiego rzędu (MSO):

- $\vee, \wedge, \neg$ (spójniki logiczne)
- $\exists_x$ x — wierzchołek
- $\exists^{\mathsf{fin}}_X$ X — skończony zbiór wierzchołków
- $\exists_X$ X — dowolny zbiór wierzchołków
- $x \in X$, $x = y$, nawigacja w drzewie, etykiety

Słaba monadyczna logika drugiego rzędu (WMSO):

- tylko kwantyfikatory $\exists_x$ i $\exists^{\mathsf{fin}}_X$

Egzystencjalny fragment monadycznej logiki drugiego rzędu (∃MSO):

Logiki

Monadyczna logika drugiego rzędu (MSO):

- $\vee$, $\wedge$, $\neg$ (spójniki logiczne)
- $\exists_x$ x — wierzchołek
- $\exists^{\mathsf{fin}}_X$ X — skończony zbiór wierzchołków
- $\exists_X$ X — dowolny zbiór wierzchołków
- $x \in X$, $x = y$, nawigacja w drzewie, etykiety

Słaba monadyczna logika drugiego rzędu (WMSO):

- tylko kwantyfikatory $\exists_x$ i $\exists^{\mathsf{fin}}_X$

Egzystencjalny fragment monadycznej logiki drugiego rzędu (∃MSO):

- $\exists_{X_1} \ldots \exists_{X_n}\ \psi$ dla $\psi \in$ WMSO

Logiki

Monadyczna logika drugiego rzędu (MSO):

- $\vee$, $\wedge$, $\neg$ (spójniki logiczne)
- $\exists_x$ x — wierzchołek
- $\exists^{\mathsf{fin}}_X$ X — skończony zbiór wierzchołków
- $\exists_X$ X — dowolny zbiór wierzchołków
- $x \in X$, $x = y$, nawigacja w drzewie, etykiety

Słaba monadyczna logika drugiego rzędu (WMSO):

- tylko kwantyfikatory $\exists_x$ i $\exists^{\mathsf{fin}}_X$

Egzystencjalny fragment monadycznej logiki drugiego rzędu ($\exists$MSO):

- $\exists_{X_1} \ldots \exists_{X_n}\ \psi$ dla $\psi \in$ WMSO

Oraz wiele innych: LTL, CTL*, modalny rachunek μ, ...

Przykład

Przykład

Istnieje gałąź z nieskończenie wieloma literami a:

$$\varphi \stackrel{\text{def}}{=} \exists X.\ \big(\exists x.\ x \in X\big) \wedge$$

Przykład

Istnieje gałąź z nieskończenie wieloma literami a:

$$\begin{aligned}\varphi \overset{\mathsf{def}}{=} \exists X.\ &\big(\exists x.\ x \in X\big) \wedge \\ &\big(\forall x.\ x \in X \Rightarrow a(x)\big) \wedge\end{aligned}$$

Przykład

Istnieje gałąź z nieskończenie wieloma literami a:

$$\varphi \stackrel{\text{def}}{=} \exists X.\ \big(\exists x.\ x \in X\big) \wedge \\ \big(\forall x.\ x \in X \Rightarrow a(x)\big) \wedge \\ \big(\forall x.\ x \in X \Rightarrow \exists y.\ x \prec y \ \wedge \ y \in X\big)$$

Przykład

Istnieje gałąź z nieskończenie wieloma literami a:

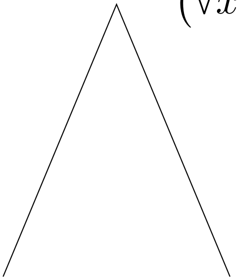
$$\begin{aligned}\varphi \stackrel{\text{def}}{=} \exists X.\ & \big(\exists x.\ x \in X\big) \land \\ & \big(\forall x.\ x \in X \Rightarrow a(x)\big) \land \\ & \big(\forall x.\ x \in X \Rightarrow \exists y.\ x \prec y \ \land\ y \in X\big)\end{aligned}$$

prefiks

Przykład

Istnieje gałąź z nieskończenie wieloma literami a:

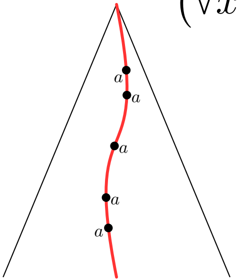
$$\begin{aligned}\varphi \stackrel{\text{def}}{=} \exists X.\ & \big(\exists x.\ x \in X\big) \wedge \\ & \big(\forall x.\ x \in X \Rightarrow a(x)\big) \wedge \\ & \big(\forall x.\ x \in X \Rightarrow \exists y.\ x \underset{\substack{\uparrow \\ \text{prefiks}}}{\prec} y \ \wedge\ y \in X\big)\end{aligned}$$



Przykład

Istnieje gałąź z nieskończenie wieloma literami a:

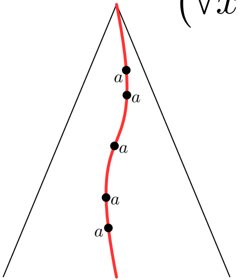
$$\varphi \stackrel{\text{def}}{=} \exists X.\ \big(\exists x.\ x \in X\big) \wedge \big(\forall x.\ x \in X \Rightarrow a(x)\big) \wedge \big(\forall x.\ x \in X \Rightarrow \exists y.\ x \underset{\substack{\uparrow \\ \text{prefiks}}}{\prec} y \ \wedge\ y \in X\big)$$



Przykład

Istnieje gałąź z nieskończenie wieloma literami a:

$$\varphi \stackrel{\text{def}}{=} \exists X.\ \big(\exists x.\ x \in X\big) \wedge \big(\forall x.\ x \in X \Rightarrow a(x)\big) \wedge \big(\forall x.\ x \in X \Rightarrow \exists y.\ \underset{\substack{\uparrow \\ \text{prefiks}}}{x \prec y} \ \wedge\ y \in X\big)$$



Twierdzenie (Niwiński [1985])

Zbiór $\mathrm{L}(\varphi)$ jest nieborelowski ($\boldsymbol{\Sigma}^1_1$-zupełny).

Deskryptywna teoria mnogości

Deskryptywna teoria mnogości

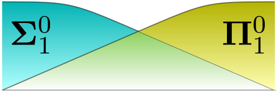
Zaczynamy od prostych zbiorów

— otwartych ($\mathbf{\Sigma}^0_1$) i domkniętych ($\mathbf{\Pi}^0_1$)

Deskryptywna teoria mnogości

Zaczynamy od prostych zbiorów

— otwartych ($\mathbf{\Sigma}^0_1$) i domkniętych ($\mathbf{\Pi}^0_1$)



Deskryptywna teoria mnogości

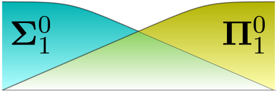
Zaczynamy od prostych zbiorów

— otwartych ($\mathbf{\Sigma}^0_1$) i domkniętych ($\mathbf{\Pi}^0_1$)

Używamy przeliczalnych sum ($\bigcup$)

i przeliczalnych przecięć ($\bigcap$)

— $\mathbf{\Sigma}^0_{\eta+1}$ oraz $\mathbf{\Pi}^0_{\eta+1}$



Deskryptywna teoria mnogości

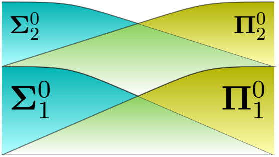
Zaczynamy od prostych zbiorów

— otwartych ($\mathbf{\Sigma}^0_1$) i domkniętych ($\mathbf{\Pi}^0_1$)

Używamy przeliczalnych sum ($\bigcup$)

i przeliczalnych przecięć ($\bigcap$)

— $\mathbf{\Sigma}^0_{\eta+1}$ oraz $\mathbf{\Pi}^0_{\eta+1}$



Deskryptywna teoria mnogości

Zaczynamy od prostych zbiorów

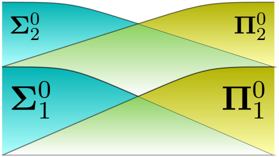
— otwartych ($\mathbf{\Sigma}^0_1$) i domkniętych ($\mathbf{\Pi}^0_1$)

Używamy przeliczalnych sum ($\bigcup$)

i przeliczalnych przecięć ($\bigcap$)

— $\mathbf{\Sigma}^0_{\eta+1}$ oraz $\mathbf{\Pi}^0_{\eta+1}$

Przez indukcję



Deskryptywna teoria mnogości

Zaczynamy od prostych zbiorów

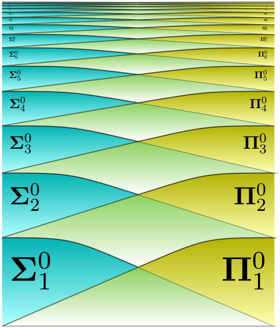
— otwartych ($\mathbf{\Sigma}^0_1$) i domkniętych ($\mathbf{\Pi}^0_1$)

Używamy przeliczalnych sum ($\bigcup$)

i przeliczalnych przecięć ($\bigcap$)

— $\mathbf{\Sigma}^0_{\eta+1}$ oraz $\mathbf{\Pi}^0_{\eta+1}$

Przez indukcję



Deskryptywna teoria mnogości

Zaczynamy od prostych zbiorów

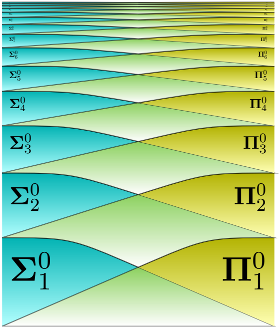
— otwartych ($\mathbf{\Sigma}^0_1$) i domkniętych ($\mathbf{\Pi}^0_1$)

Używamy przeliczalnych sum ($\bigcup$)

i przeliczalnych przecięć ($\bigcap$)

— $\mathbf{\Sigma}^0_{\eta+1}$ oraz $\mathbf{\Pi}^0_{\eta+1}$

Przez indukcję (pozaskończoną)



Deskryptywna teoria mnogości

Zaczynamy od prostych zbiorów

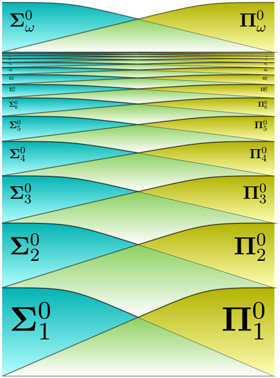
— otwartych ($\mathbf{\Sigma}^0_1$) i domkniętych ($\mathbf{\Pi}^0_1$)

Używamy przeliczalnych sum ($\bigcup$)

i przeliczalnych przecięć ($\bigcap$)

— $\mathbf{\Sigma}^0_{\eta+1}$ oraz $\mathbf{\Pi}^0_{\eta+1}$

Przez indukcję (pozaskończoną)



Deskryptywna teoria mnogości

Zaczynamy od prostych zbiorów

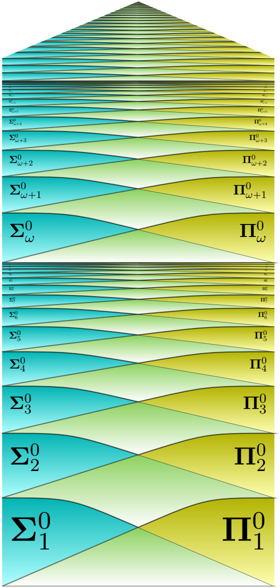
— otwartych ($\mathbf{\Sigma}^0_1$) i domkniętych ($\mathbf{\Pi}^0_1$)

Używamy przeliczalnych sum ($\bigcup$)

i przeliczalnych przecięć ($\bigcap$)

— $\mathbf{\Sigma}^0_{\eta+1}$ oraz $\mathbf{\Pi}^0_{\eta+1}$

Przez indukcję (pozaskończoną)



Deskryptywna teoria mnogości

Zaczynamy od prostych zbiorów

— otwartych ($\mathbf{\Sigma}^0_1$) i domkniętych ($\mathbf{\Pi}^0_1$)

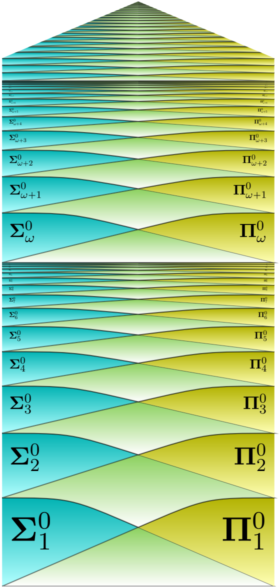
Używamy przeliczalnych sum ($\bigcup$)

i przeliczalnych przecięć ($\bigcap$)

— $\mathbf{\Sigma}^0_{\eta+1}$ oraz $\mathbf{\Pi}^0_{\eta+1}$

Przez indukcję (pozaskończoną)

— zbiory borelowskie: $\mathbf{\Sigma}^0_\eta$, $\mathbf{\Pi}^0_\eta$ dla $\eta < \omega_1$



Deskryptywna teoria mnogości

Zaczynamy od prostych zbiorów

— otwartych ($\mathbf{\Sigma}^0_1$) i domkniętych ($\mathbf{\Pi}^0_1$)

Używamy przeliczalnych sum ($\bigcup$)

i przeliczalnych przecięć ($\bigcap$)

— $\mathbf{\Sigma}^0_{\eta+1}$ oraz $\mathbf{\Pi}^0_{\eta+1}$

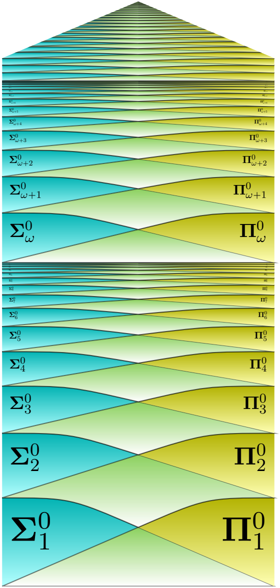
Przez indukcję (pozaskończoną)

— zbiory borelowskie: $\mathbf{\Sigma}^0_\eta$, $\mathbf{\Pi}^0_\eta$ dla $\eta < \omega_1$

Używamy rzutów i dopełnień

— zbiory analityczne ($\mathbf{\Sigma}^1_1$)

oraz koanalityczne ($\mathbf{\Pi}^1_1$)



Deskryptywna teoria mnogości

Zaczynamy od prostych zbiorów

— otwartych ($\mathbf{\Sigma}^0_1$) i domkniętych ($\mathbf{\Pi}^0_1$)

Używamy przeliczalnych sum ($\bigcup$)

i przeliczalnych przecięć ($\bigcap$)

— $\mathbf{\Sigma}^0_{\eta+1}$ oraz $\mathbf{\Pi}^0_{\eta+1}$

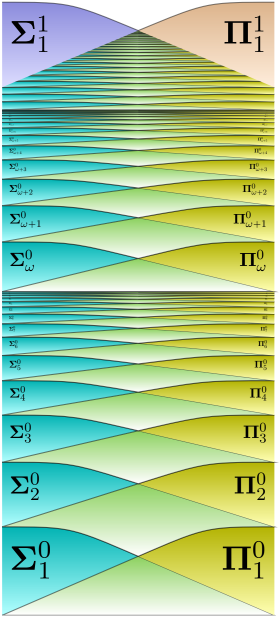
Przez indukcję (pozaskończoną)

— zbiory borelowskie: $\mathbf{\Sigma}^0_\eta$, $\mathbf{\Pi}^0_\eta$ dla $\eta < \omega_1$

Używamy rzutów i dopełnień

— zbiory analityczne ($\mathbf{\Sigma}^1_1$)

oraz koanalityczne ($\mathbf{\Pi}^1_1$)



Deskryptywna teoria mnogości

Zaczynamy od prostych zbiorów

— otwartych ($\mathbf{\Sigma}^0_1$) i domkniętych ($\mathbf{\Pi}^0_1$)

Używamy przeliczalnych sum ($\bigcup$)

i przeliczalnych przecięć ($\bigcap$)

— $\mathbf{\Sigma}^0_{\eta+1}$ oraz $\mathbf{\Pi}^0_{\eta+1}$

Przez indukcję (pozaskończoną)

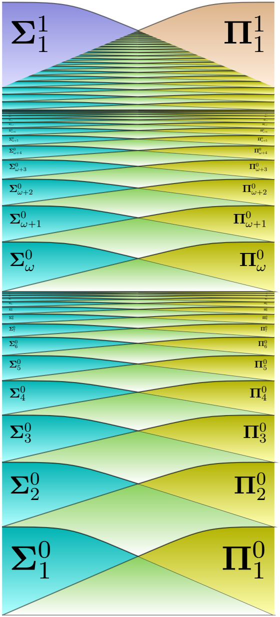
— zbiory borelowskie: $\mathbf{\Sigma}^0_\eta$, $\mathbf{\Pi}^0_\eta$ dla $\eta < \omega_1$

Używamy rzutów i dopełnień

— zbiory analityczne ($\mathbf{\Sigma}^1_1$)

oraz koanalityczne ($\mathbf{\Pi}^1_1$)

Przez indukcję



Deskryptywna teoria mnogości

Zaczynamy od prostych zbiorów

— otwartych ($\mathbf{\Sigma}^0_1$) i domkniętych ($\mathbf{\Pi}^0_1$)

Używamy przeliczalnych sum ($\bigcup$)

i przeliczalnych przecięć ($\bigcap$)

— $\mathbf{\Sigma}^0_{\eta+1}$ oraz $\mathbf{\Pi}^0_{\eta+1}$

Przez indukcję (pozaskończoną)

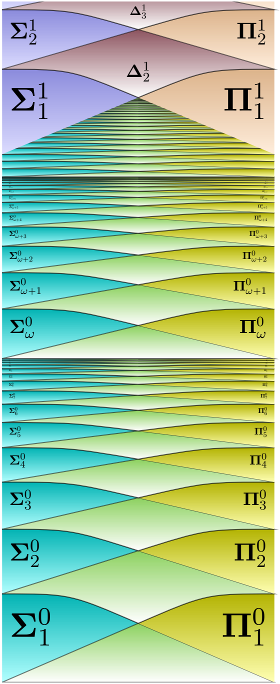
— zbiory borelowskie: $\mathbf{\Sigma}^0_\eta$, $\mathbf{\Pi}^0_\eta$ dla $\eta < \omega_1$

Używamy rzutów i dopełnień

— zbiory analityczne ($\mathbf{\Sigma}^1_1$)

oraz koanalityczne ($\mathbf{\Pi}^1_1$)

Przez indukcję



Deskryptywna teoria mnogości

Zaczynamy od prostych zbiorów

— otwartych ($\mathbf{\Sigma}^0_1$) i domkniętych ($\mathbf{\Pi}^0_1$)

Używamy przeliczalnych sum ($\bigcup$)

i przeliczalnych przecięć ($\bigcap$)

— $\mathbf{\Sigma}^0_{\eta+1}$ oraz $\mathbf{\Pi}^0_{\eta+1}$

Przez indukcję (pozaskończoną)

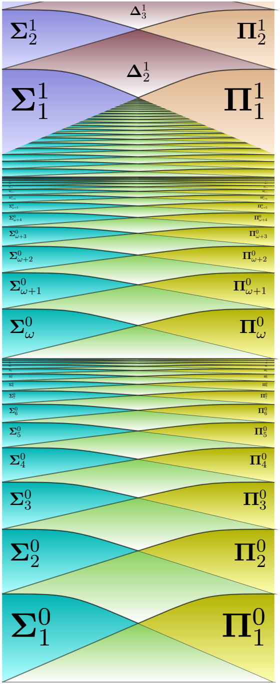
— zbiory borelowskie: $\mathbf{\Sigma}^0_\eta$, $\mathbf{\Pi}^0_\eta$ dla $\eta < \omega_1$

Używamy rzutów i dopełnień

— zbiory analityczne ($\mathbf{\Sigma}^1_1$) oraz koanalityczne ($\mathbf{\Pi}^1_1$)

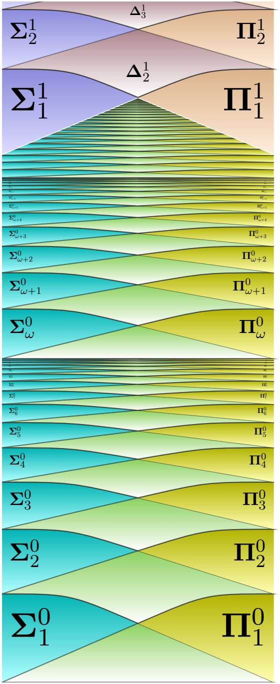
Przez indukcję

— zbiory rzutowe: $\mathbf{\Sigma}^1_n$, $\mathbf{\Pi}^1_n$ dla $n < \omega$



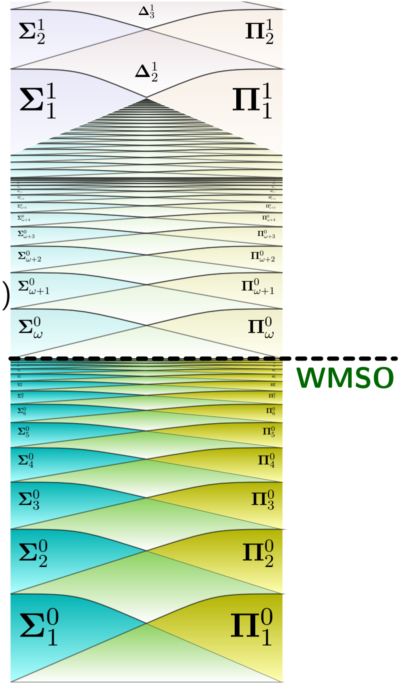
Ograniczenia górne

Ograniczenia górne



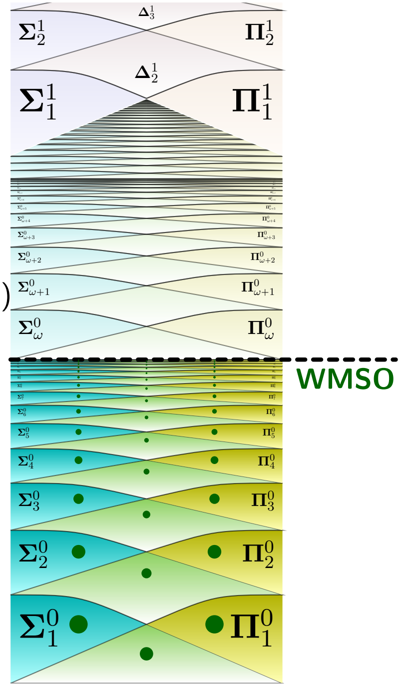
Ograniczenia górne

$L \in$ **WMSO** $\implies$ $L \in \mathbf{\Sigma}^0_n$ (dla pewnego n)



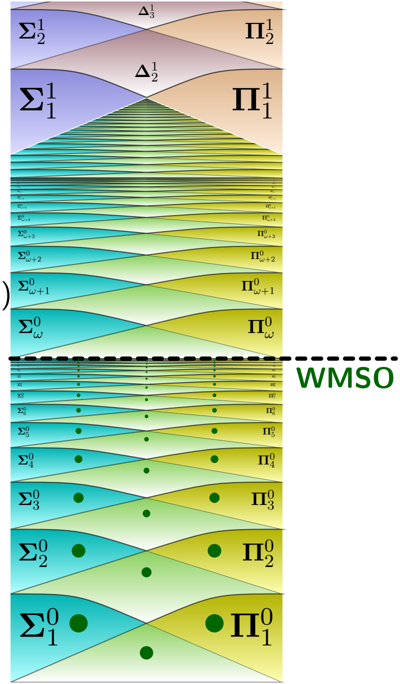
Ograniczenia górne

$$L \in \textbf{WMSO} \implies L \in \boldsymbol{\Sigma}^0_n \ \text{(dla pewnego } n\text{)}$$



Ograniczenia górne

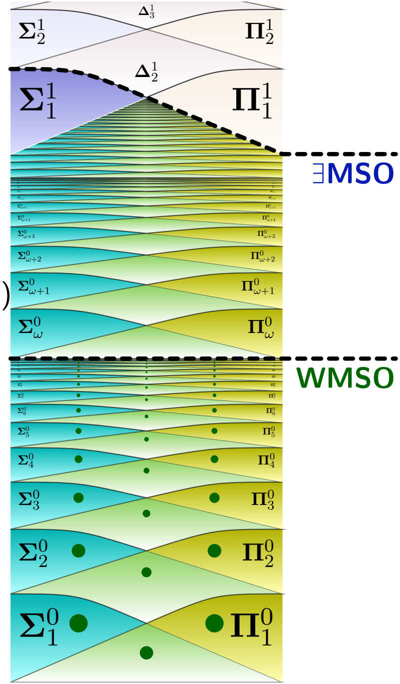
$$L \in \textbf{WMSO} \implies L \in \boldsymbol{\Sigma}^0_n \quad \text{(dla pewnego } n\text{)}$$



Ograniczenia górne

$$L \in \exists\text{MSO} \implies L \in \Sigma^1_1$$

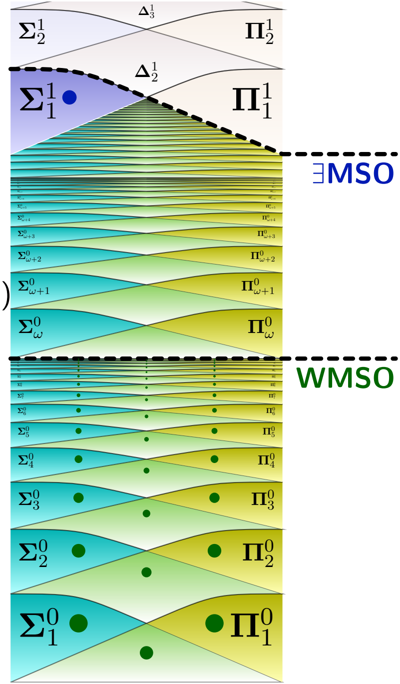
$$L \in \text{WMSO} \implies L \in \Sigma^0_n \quad (\text{dla pewnego } n)$$



Ograniczenia górne

$L \in \exists\text{MSO} \implies L \in \Sigma_1^1$

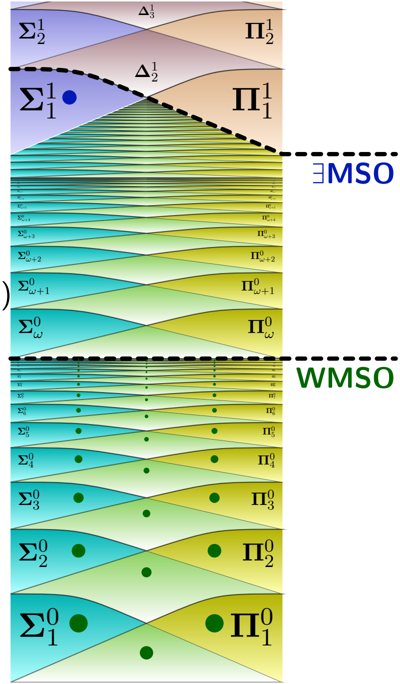
$L \in \text{WMSO} \implies L \in \Sigma_n^0$ (dla pewnego n)



Ograniczenia górne

$$L \in \exists\text{MSO} \implies L \in \mathbf{\Sigma}^1_1$$

$$L \in \text{WMSO} \implies L \in \mathbf{\Sigma}^0_n \quad \text{(dla pewnego } n\text{)}$$

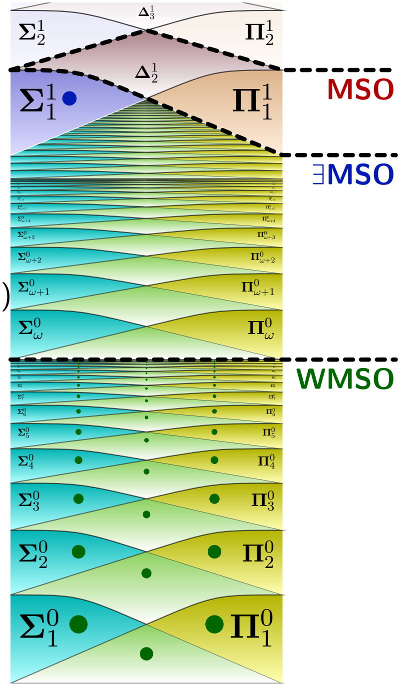


Ograniczenia górne

$L \in$ **MSO** $\implies L \in \mathbf{\Delta}^1_2$

$L \in$ **∃MSO** $\implies L \in \mathbf{\Sigma}^1_1$

$L \in$ **WMSO** $\implies L \in \mathbf{\Sigma}^0_n$ (dla pewnego n)

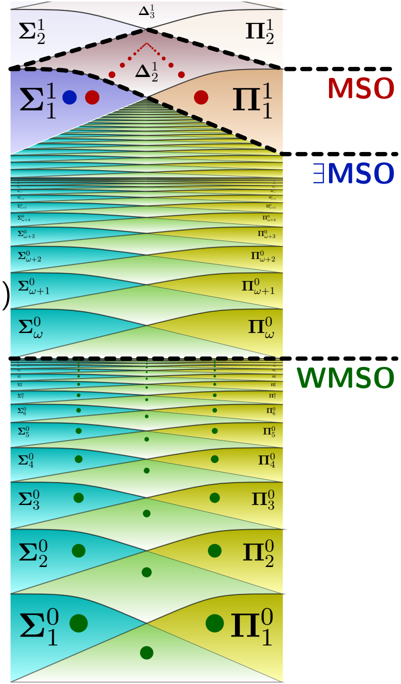


Ograniczenia górne

$L \in$ **MSO** $\implies L \in \mathbf{\Delta}^1_2$

$L \in$ **∃MSO** $\implies L \in \mathbf{\Sigma}^1_1$

$L \in$ **WMSO** $\implies L \in \mathbf{\Sigma}^0_n$ (dla pewnego n)

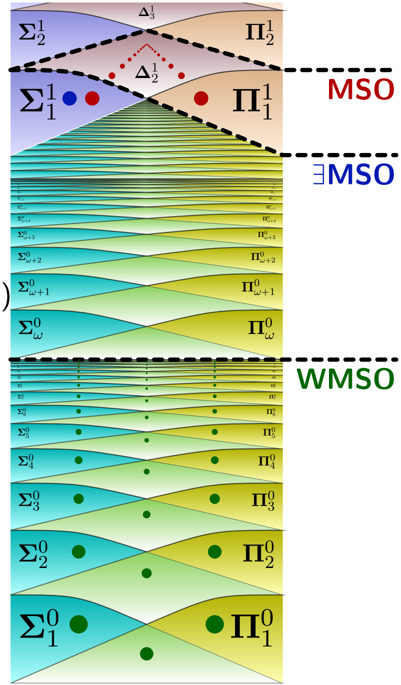


Ograniczenia górne

$$L \in \textbf{MSO} \implies L \in \boldsymbol{\Delta}^1_2$$

$$L \in \exists\textbf{MSO} \implies L \in \boldsymbol{\Sigma}^1_1$$

$$L \in \textbf{WMSO} \implies L \in \boldsymbol{\Sigma}^0_n \quad \text{(dla pewnego } n\text{)}$$



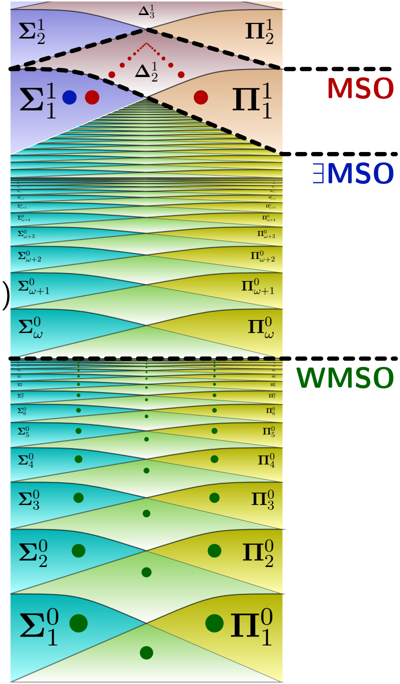
Ograniczenia górne

$L \in$ **MSO** $\implies L \in \mathbf{\Delta}^1_2$

$L \in$ **∃MSO** $\implies L \in \mathbf{\Sigma}^1_1$

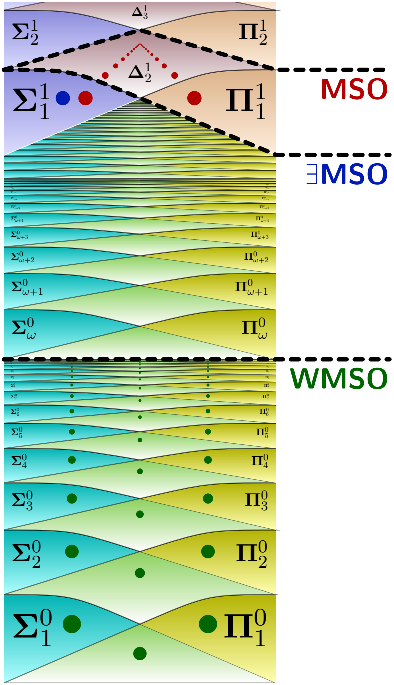
$L \in$ **WMSO** $\implies L \in \mathbf{\Sigma}^0_n$ (dla pewnego n)

WMSO $\subsetneq$ **∃MSO** $\subsetneq$ **MSO**



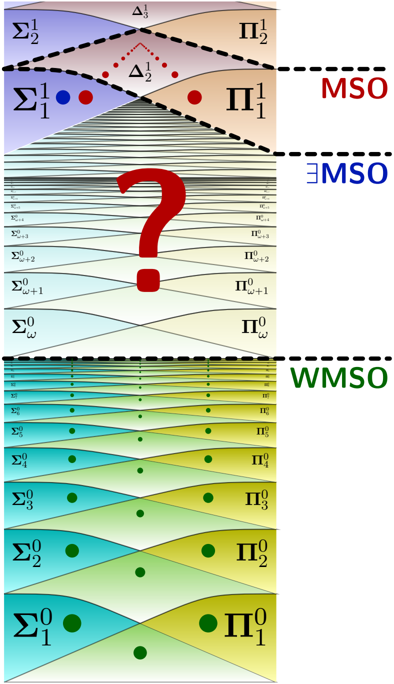
Dychotomia

Dychotomia



Δ^1_3
Σ^1_2
Π^1_2
Δ^1_2
MSO
Σ^1_1
Π^1_1
∃MSO
$\Sigma^0_{\omega+4}$
$\Pi^0_{\omega+4}$
$\Sigma^0_{\omega+3}$
$\Pi^0_{\omega+3}$
$\Sigma^0_{\omega+2}$
$\Pi^0_{\omega+2}$
$\Sigma^0_{\omega+1}$
$\Pi^0_{\omega+1}$
Σ^0_ω
Π^0_ω
WMSO
Σ^0_6
Π^0_6
Σ^0_5
Π^0_5
Σ^0_4
Π^0_4
Σ^0_3
Π^0_3
Σ^0_2
Π^0_2
Σ^0_1
Π^0_1

Dychotomia



Δ^1_3
Σ^1_2
Π^1_2
Δ^1_2
MSO
Σ^1_1
Π^1_1
$\exists$MSO
?
$\Sigma^0_{\omega+4}$
$\Pi^0_{\omega+4}$
$\Sigma^0_{\omega+3}$
$\Pi^0_{\omega+3}$
$\Sigma^0_{\omega+2}$
$\Pi^0_{\omega+2}$
$\Sigma^0_{\omega+1}$
$\Pi^0_{\omega+1}$
Σ^0_ω
Π^0_ω
WMSO
Σ^0_6
Π^0_6
Σ^0_5
Π^0_5
Σ^0_4
Π^0_4
Σ^0_3
Π^0_3
Σ^0_2
Π^0_2
Σ^0_1
Π^0_1

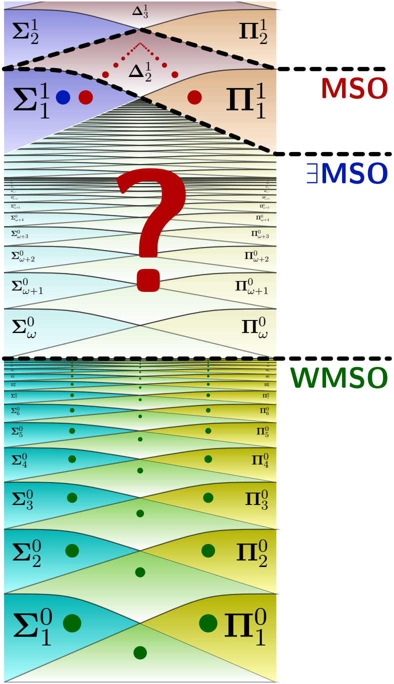
Dychotomia

Hipoteza (Skurczyński [1993])

MSO-definiowalny zbiór,

jeśli jest borelowski,

to jest WMSO-definiowalny.



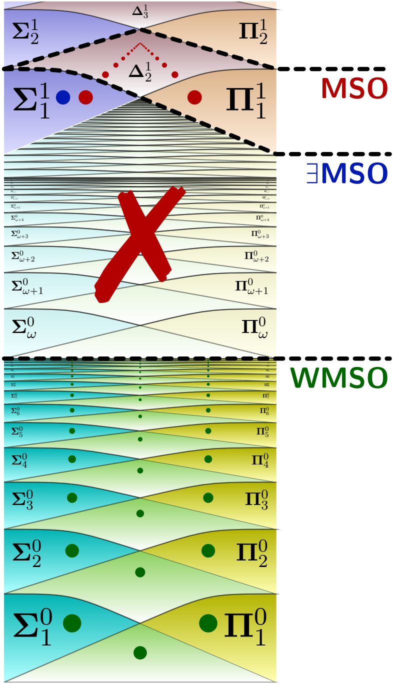
Dychotomia

Hipoteza (Skurczyński [1993])

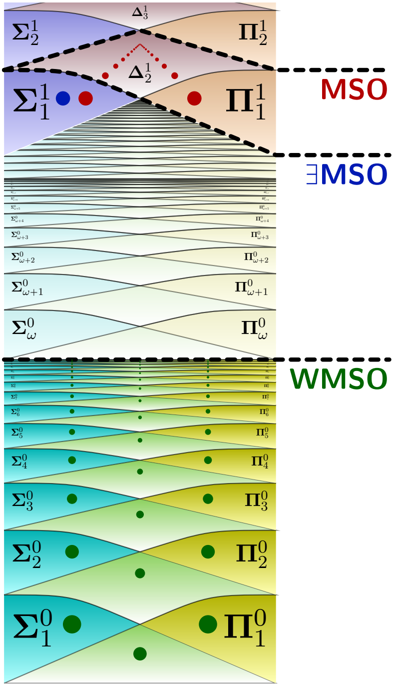
MSO-definiowalny zbiór,

jeśli jest borelowski,

to jest WMSO-definiowalny.



Dychotomia



Hipoteza (Skurczyński [1993])

MSO-definiowalny zbiór,

jeśli jest borelowski,

to jest WMSO-definiowalny.

Theorem (Walukiewicz, S. [2016])

∃MSO-definiowalny zbiór jest albo:

— WMSO-definiowalny i borelowski

— **nie** WMSO-definiowalny i Σ^1_1-zupełny.

Dychotomia

Hipoteza (Skurczyński [1993])

MSO-definiowalny zbiór,

jeśli jest borelowski,

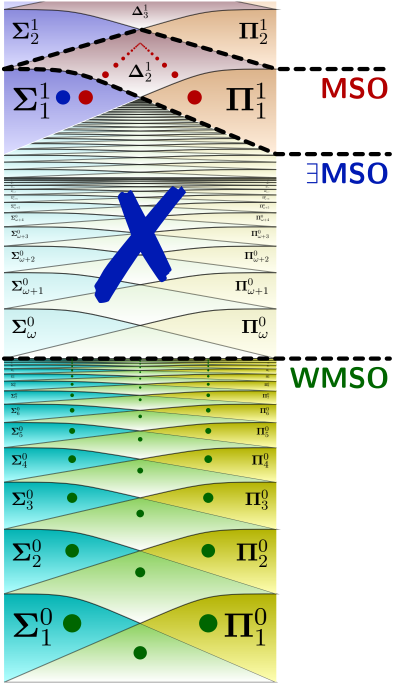
to jest WMSO-definiowalny.

Theorem (Walukiewicz, S. [2016])

∃MSO-definiowalny zbiór jest albo:

— WMSO-definiowalny i borelowski

— **nie** WMSO-definiowalny i Σ^1_1-zupełny.



Dychotomia

Hipoteza (Skurczyński [1993])

MSO-definiowalny zbiór,

jeśli jest borelowski,

to jest WMSO-definiowalny.

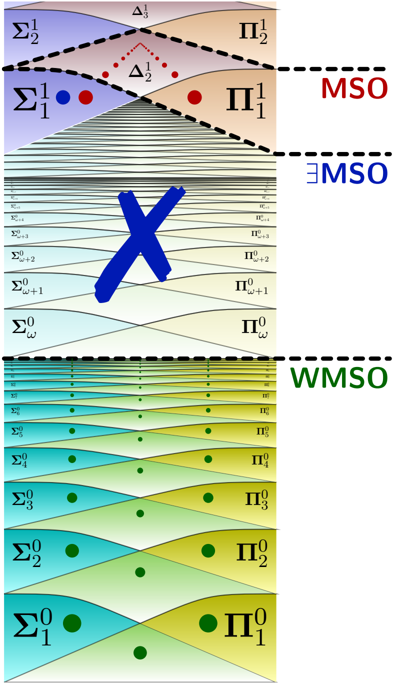
Theorem (Walukiewicz, S. [2016])

∃MSO-definiowalny zbiór jest albo:

— WMSO-definiowalny i borelowski

— **nie** WMSO-definiowalny i Σ^1_1-zupełny.

Oraz ta dychotomia jest efektywna.



Dychotomia

Hipoteza (Skurczyński [1993])

MSO-definiowalny zbiór,

jeśli jest borelowski,

to jest WMSO-definiowalny.

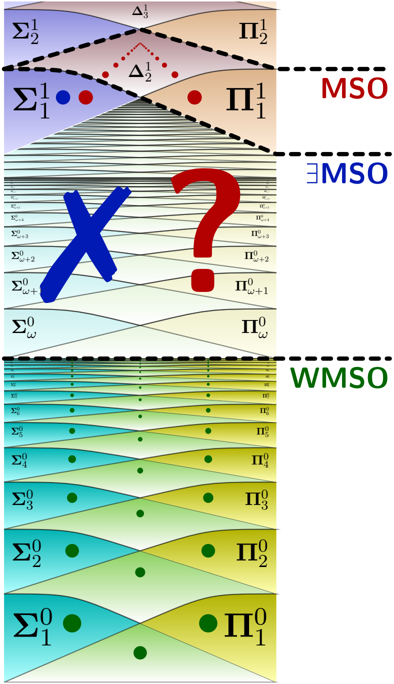
Theorem (Walukiewicz, S. [2016])

∃MSO-definiowalny zbiór jest albo:

— WMSO-definiowalny i borelowski

— **nie** WMSO-definiowalny i Σ^1_1-zupełny.

Oraz ta dychotomia jest efektywna.



Twierdzenie (Walukiewicz, S. [2016])

$L \in$ **∃MSO** jest efektywnie albo:

— **WMSO**-definiowalny,

— $\mathbf{\Sigma}_1^1$-zupełny.

Twierdzenie (Walukiewicz, S. [2016])

$L \in$ **∃MSO** jest efektywnie albo:

— **WMSO**-definiowalny,

— $\mathbf{\Sigma}_1^1$-zupełny.

Dowód

Twierdzenie (Walukiewicz, S. [2016])

$L \in$ **∃MSO** jest efektywnie albo:

— **WMSO**-definiowalny,

— $\boldsymbol{\Sigma}^1_1$-zupełny.

Dowód

Weź $\varphi \in$ **∃MSO** dla L (czyli $L = \mathrm{L}(\varphi)$)

Twierdzenie (Walukiewicz, S. [2016])

$L \in$ **∃MSO** jest efektywnie albo:

— **WMSO**-definiowalny,

— $\mathbf{\Sigma}^1_1$-zupełny.

Dowód

Weź $\varphi \in$ **∃MSO** dla L (czyli $L = \mathrm{L}(\varphi)$)

Stwórz grę $\mathcal{F}$ o skończonej arenie i nieskończonych rozgrywkach.

Twierdzenie (Walukiewicz, S. [2016])

$L \in \exists$MSO jest efektywnie albo:

— WMSO-definiowalny,

— $\mathbf{\Sigma}_1^1$-zupełny.

Dowód

Weź $\varphi \in \exists$MSO dla L (czyli $L = \mathrm{L}(\varphi)$)

Stwórz grę $\mathcal{F}$ o skończonej arenie i nieskończonych rozgrywkach.

$\rightsquigarrow$ Jeśli $\forall$ wygrywa $\mathcal{F}$:

Twierdzenie (Walukiewicz, S. [2016])

$L \in \exists$MSO jest efektywnie albo:

— WMSO-definiowalny,

— $\mathbf{\Sigma}_1^1$-zupełny.

Dowód

Weź $\varphi \in \exists$MSO dla L (czyli $L = \mathrm{L}(\varphi)$)

Stwórz grę $\mathcal{F}$ o skończonej arenie i nieskończonych rozgrywkach.

$\rightsquigarrow$ Jeśli $\forall$ wygrywa $\mathcal{F}$:

$\sigma_\forall$ — skończenie pamięciowa strategia wygrywająca dla $\forall$

Twierdzenie (Walukiewicz, S. [2016])

$L \in \exists$MSO jest efektywnie albo:

— WMSO-definiowalny,

— $\mathbf{\Sigma}_1^1$-zupełny.

Dowód

Weź $\varphi \in \exists$MSO dla L (czyli $L = \mathrm{L}(\varphi)$)

Stwórz grę $\mathcal{F}$ o skończonej arenie i nieskończonych rozgrywkach.

$\rightsquigarrow$ Jeśli $\forall$ wygrywa $\mathcal{F}$:

$\sigma_\forall$ — skończenie pamięciowa strategia wygrywająca dla $\forall$

$\sigma_\forall$ pozwala napisać formułę WMSO dla L

Twierdzenie (Walukiewicz, S. [2016])

$L \in$ **∃MSO** jest efektywnie albo:

— **WMSO**-definiowalny,

— $\mathbf{\Sigma}^1_1$-zupełny.

Dowód

Weź $\varphi \in$ **∃MSO** dla L (czyli $L = \mathrm{L}(\varphi)$)

Stwórz grę $\mathcal{F}$ o skończonej arenie i nieskończonych rozgrywkach.

$\rightsquigarrow$ Jeśli $\forall$ wygrywa $\mathcal{F}$: $\rightsquigarrow$ $L \in$ **WMSO**

$\sigma_\forall$ — skończenie pamięciowa strategia wygrywająca dla $\forall$

$\sigma_\forall$ pozwala napisać formułę **WMSO** dla L

Twierdzenie (Walukiewicz, S. [2016])

$L \in \exists$MSO jest efektywnie albo:

— WMSO-definiowalny,

— Σ^1_1-zupełny.

Dowód

Weź $\varphi \in \exists$MSO dla L (czyli $L = \mathrm{L}(\varphi)$)

Stwórz grę $\mathcal{F}$ o skończonej arenie i nieskończonych rozgrywkach.

$\rightsquigarrow$ Jeśli $\forall$ wygrywa $\mathcal{F}$: $\rightsquigarrow L \in$ **WMSO**

$\sigma_\forall$ — skończenie pamięciowa strategia wygrywająca dla $\forall$

$\sigma_\forall$ pozwala napisać formułę WMSO dla L

$\rightsquigarrow$ Jeśli $\exists$ wygrywa $\mathcal{F}$:

Twierdzenie (Walukiewicz, S. [2016])

$L \in$ **∃MSO** jest efektywnie albo:

— **WMSO**-definiowalny,

— $\mathbf{\Sigma}^1_1$-zupełny.

Dowód

Weź $\varphi \in$ **∃MSO** dla L (czyli $L = \mathrm{L}(\varphi)$)

Stwórz grę $\mathcal{F}$ o skończonej arenie i nieskończonych rozgrywkach.

$\rightsquigarrow$ Jeśli ∀ wygrywa $\mathcal{F}$: $\rightsquigarrow$ $L \in$ **WMSO**

$\sigma_\forall$ — skończenie pamięciowa strategia wygrywająca dla ∀

$\sigma_\forall$ pozwala napisać formułę **WMSO** dla L

$\rightsquigarrow$ Jeśli ∃ wygrywa $\mathcal{F}$:

$\sigma_\exists$ — strategia wygrywająca dla ∃

Twierdzenie (Walukiewicz, S. [2016])

$L \in$ **∃MSO** jest efektywnie albo:

— **WMSO**-definiowalny,

— $\mathbf{\Sigma}^1_1$-zupełny.

Dowód

Weź $\varphi \in$ **∃MSO** dla L (czyli $L = \mathrm{L}(\varphi)$)

Stwórz grę $\mathcal{F}$ o skończonej arenie i nieskończonych rozgrywkach.

$\rightsquigarrow$ Jeśli ∀ wygrywa $\mathcal{F}$: $\rightsquigarrow L \in$ **WMSO**

$\sigma_\forall$ — skończenie pamięciowa strategia wygrywająca dla ∀

$\sigma_\forall$ pozwala napisać formułę **WMSO** dla L

$\rightsquigarrow$ Jeśli ∃ wygrywa $\mathcal{F}$:

$\sigma_\exists$ — strategia wygrywająca dla ∃

$\sigma_\exists$ pozwala stworzyć ciągłą redukcję zbioru IF do L

Twierdzenie (Walukiewicz, S. [2016])

$L \in$ **∃MSO** jest efektywnie albo:

— **WMSO**-definiowalny,

— $\boldsymbol{\Sigma}^1_1$-zupełny.

Dowód

Weź $\varphi \in$ **∃MSO** dla L (czyli $L = \mathrm{L}(\varphi)$)

Stwórz grę $\mathcal{F}$ o skończonej arenie i nieskończonych rozgrywkach.

⇝ Jeśli ∀ wygrywa $\mathcal{F}$: ⇝ $L \in$ **WMSO**

$\sigma_\forall$ — skończenie pamięciowa strategia wygrywająca dla ∀

$\sigma_\forall$ pozwala napisać formułę **WMSO** dla L

⇝ Jeśli ∃ wygrywa $\mathcal{F}$: ⇝ L jest $\boldsymbol{\Sigma}^1_1$-zupełny

$\sigma_\exists$ — strategia wygrywająca dla ∃

$\sigma_\exists$ pozwala stworzyć ciągłą redukcję zbioru IF do L

Dychotomie

Dychotomie

Twierdzenie (Niwiński, Walukiewicz [2003])

L rozpoznawany automatem deterministycznym jest:

WMSO-definiowalny **albo** $\mathbf{\Pi}^1_1$-zupełny.

Dychotomie

Twierdzenie (Niwiński, Walukiewicz [2003])

L rozpoznawany automatem deterministycznym jest:

WMSO-definiowalny **albo** $\mathbf{\Pi}^1_1$-zupełny.

Twierdzenie (Bojańczyk, Idziaszek, S. [2013])

L — język drzew cienkich jest:

WMSO-definiowalny **albo** $\mathbf{\Pi}^1_1$-zupełny.

Dychotomie

Twierdzenie (Niwiński, Walukiewicz [2003])

L rozpoznawany automatem deterministycznym jest:

WMSO-definiowalny **albo** $\mathbf{\Pi}^1_1$-zupełny.

Twierdzenie (Bojańczyk, Idziaszek, S. [2013])

L — język drzew cienkich jest:

WMSO-definiowalny **albo** $\mathbf{\Pi}^1_1$-zupełny.

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

L rozpoznawany automatem growym jest:

WMSO-definiowalny **albo** $\mathbf{\Sigma}^1_1$- lub $\mathbf{\Pi}^1_1$-trudny.

Dychotomie

Twierdzenie (Niwiński, Walukiewicz [2003])

L rozpoznawany automatem deterministycznym jest:

WMSO-definiowalny **albo** $\mathbf{\Pi}^1_1$-zupełny.

Twierdzenie (Bojańczyk, Idziaszek, S. [2013])

L — język drzew cienkich jest:

WMSO-definiowalny **albo** $\mathbf{\Pi}^1_1$-zupełny.

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

L rozpoznawany automatem growym jest:

WMSO-definiowalny **albo** $\mathbf{\Sigma}^1_1$- lub $\mathbf{\Pi}^1_1$-trudny.

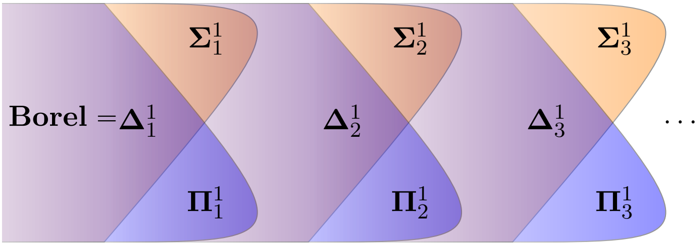
Twierdzenie (Walukiewicz, S. [2016])

$L \in \exists$MSO jest:

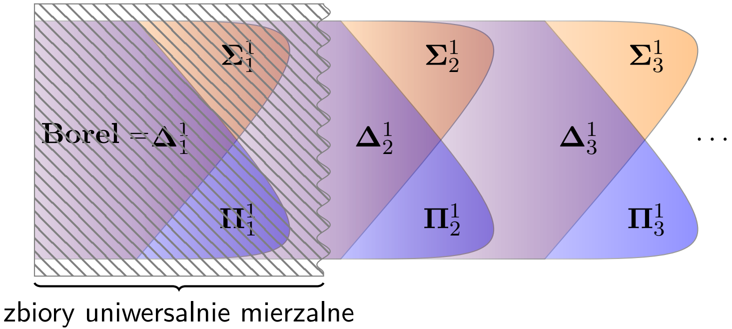
WMSO-definiowalny **albo** $\mathbf{\Sigma}^1_1$-zupełny.

Mierzalność

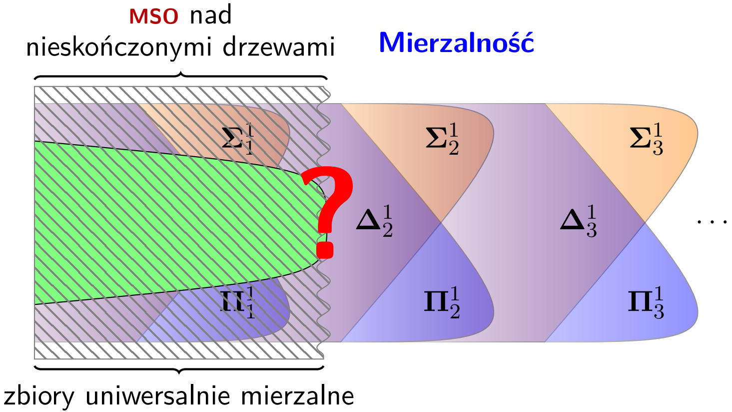
Mierzalność



Mierzalność

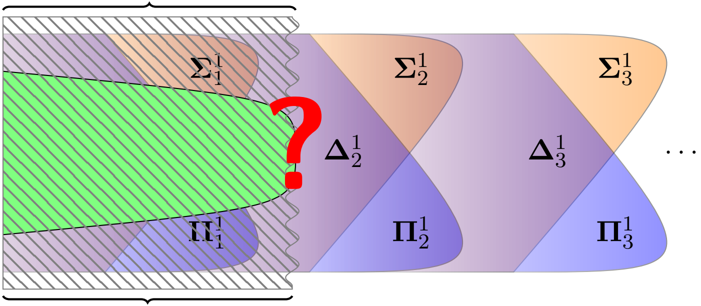


Mierzalność



Mierzalność

MSO nad nieskończonymi drzewami

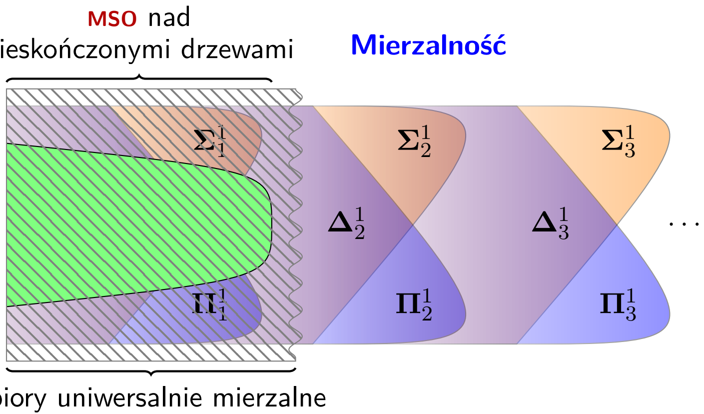


zbiory uniwersalnie mierzalne

Twierdzenie (Mio [2012]) $\left(\mathbf{MA}_{\aleph_1}\right)$

Wszystkie języki MSO-definiowalne są uniwersalnie mierzalne.

Mierzalność

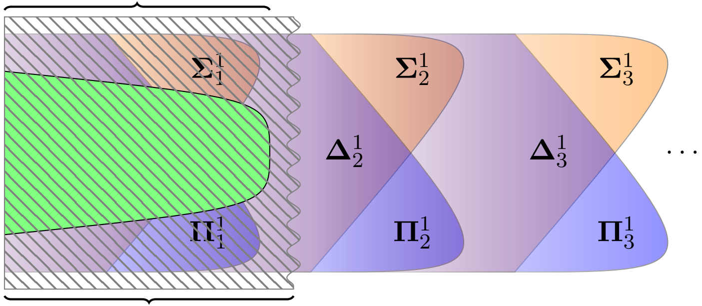


Twierdzenie (Mio [2012]) $\left(\mathbf{MA}_{\aleph_1}\right)$

Wszystkie języki MSO-definiowalne są uniwersalnie mierzalne.

Mierzalność

MSO nad nieskończonymi drzewami



zbiory uniwersalnie mierzalne

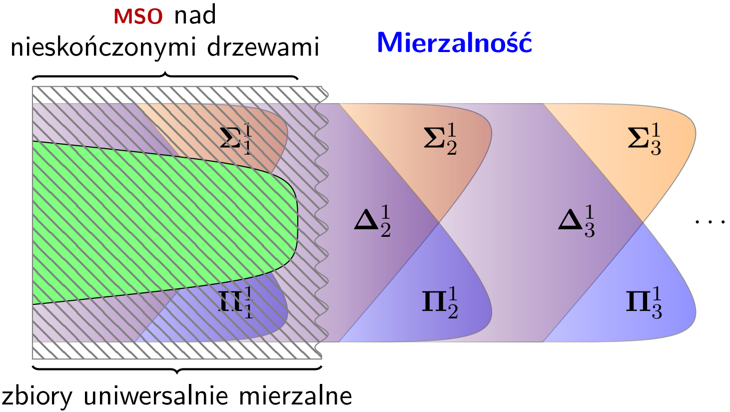
Twierdzenie (Mio [2012]) $(\mathbf{MA}_{\aleph_1})$

Wszystkie języki MSO-definiowalne są uniwersalnie mierzalne.

Twierdzenie (Gogacz, Michalewski, Mio, S. [2014])

Wszystkie języki MSO-definiowalne są uniwersalnie mierzalne.

Mierzalność



Twierdzenie (Mio [2012]) $\left(\mathbf{MA}_{\aleph_1}\right)$

Wszystkie języki MSO-definiowalne są uniwersalnie mierzalne.

Twierdzenie (Gogacz, Michalewski, Mio, S. [2014])

Wszystkie języki MSO-definiowalne są uniwersalnie mierzalne.

$\rightsquigarrow$ związki z teorią $\mathcal{R}$-zbiorów Kołmogorowa [1928]

Automaty history-deterministic

Automaty history-deterministic

determinizm $\leftrightsquigarrow$ sekwencyjna konstrukcja obliczenia

Automaty history-deterministic

determinizm $\leftrightsquigarrow$ sekwencyjna konstrukcja obliczenia

niedeterminizm $\leftrightsquigarrow$ zgadywanie (z uwzględnieniem całego wejścia)

Automaty history-deterministic

determinizm $\leftrightsquigarrow$ sekwencyjna konstrukcja obliczenia

niedeterminizm $\leftrightsquigarrow$ zgadywanie (z uwzględnieniem całego wejścia)

hist.-det. $\leftrightsquigarrow$ zgadywanie oparte **wyłącznie** o już wczytaną część wejścia

Automaty history-deterministic

determinizm $\leftrightsquigarrow$ sekwencyjna konstrukcja obliczenia

niedeterminizm $\leftrightsquigarrow$ zgadywanie (z uwzględnieniem całego wejścia)

hist.-det. $\leftrightsquigarrow$ zgadywanie oparte **wyłącznie** o
już wczytaną część wejścia

Twierdzenie (Henzinger, Piterman [2006])

History-determinizm może zastępować determinizm
przy rozwiązywaniu problemu syntezy.

Automaty history-deterministic

determinizm $\leftrightsquigarrow$ sekwencyjna konstrukcja obliczenia

niedeterminizm $\leftrightsquigarrow$ zgadywanie (z uwzględnieniem całego wejścia)

hist.-det. $\leftrightsquigarrow$ zgadywanie oparte **wyłącznie** o
już wczytaną część wejścia

Twierdzenie (Henzinger, Piterman [2006])

History-determinizm może zastępować determinizm
przy rozwiązywaniu problemu syntezy.

$\mathcal{A}$ jest history-deterministic jeśli

Automaty history-deterministic

determinizm $\leftrightsquigarrow$ sekwencyjna konstrukcja obliczenia

niedeterminizm $\leftrightsquigarrow$ zgadywanie (z uwzględnieniem całego wejścia)

hist.-det. $\leftrightsquigarrow$ zgadywanie oparte **wyłącznie** o

już wczytaną część wejścia

Twierdzenie (Henzinger, Piterman [2006])

History-determinizm może zastępować determinizm

przy rozwiązywaniu problemu syntezy.

$\mathcal{A}$ jest history-deterministic jeśli

$$\exists_{\sigma\colon \Sigma^*\to Q}$$

Automaty history-deterministic

determinizm $\leftsquigarrow\!\!\rightsquigarrow$ sekwencyjna konstrukcja obliczenia

niedeterminizm $\leftsquigarrow\!\!\rightsquigarrow$ zgadywanie (z uwzględnieniem całego wejścia)

hist.-det. $\leftsquigarrow\!\!\rightsquigarrow$ zgadywanie oparte **wyłącznie** o już wczytaną część wejścia

Twierdzenie (Henzinger, Piterman [2006])

History-determinizm może zastępować determinizm przy rozwiązywaniu problemu syntezy.

$\mathcal{A}$ jest history-deterministic jeśli

$$\exists_{\sigma\colon \Sigma^* \to Q} \qquad \forall_{w \in \mathrm{L}(\mathcal{A})}$$

Automaty history-deterministic

determinizm $\leftrightsquigarrow$ sekwencyjna konstrukcja obliczenia

niedeterminizm $\leftrightsquigarrow$ zgadywanie (z uwzględnieniem całego wejścia)

hist.-det. $\leftrightsquigarrow$ zgadywanie oparte **wyłącznie** o już wczytaną część wejścia

Twierdzenie (Henzinger, Piterman [2006])

History-determinizm może zastępować determinizm przy rozwiązywaniu problemu syntezy.

$\mathcal{A}$ jest history-deterministic jeśli

$$\exists_{\sigma\colon \Sigma^*\to Q} \qquad \forall_{w\in \mathrm{L}(\mathcal{A})} \qquad \text{bieg } \sigma(w) \text{ jest akceptujący}$$

Automaty history-deterministic

determinizm $\leftrightsquigarrow$ sekwencyjna konstrukcja obliczenia

niedeterminizm $\leftrightsquigarrow$ zgadywanie (z uwzględnieniem całego wejścia)

hist.-det. $\leftrightsquigarrow$ zgadywanie oparte **wyłącznie** o już wczytaną część wejścia

Twierdzenie (Henzinger, Piterman [2006])

History-determinizm może zastępować determinizm przy rozwiązywaniu problemu syntezy.

$\mathcal{A}$ jest history-deterministic jeśli

$$\Big(\sigma(), \sigma(w_0), \sigma(w_0 w_1), \ldots\Big)$$

$$\exists_{\sigma\colon \Sigma^* \to Q} \qquad \forall_{w \in \mathrm{L}(\mathcal{A})} \qquad \text{bieg } \sigma(w) \text{ jest akceptujący}$$

Automaty history-deterministic

determinizm $\leftrightsquigarrow$ sekwencyjna konstrukcja obliczenia

niedeterminizm $\leftrightsquigarrow$ zgadywanie (z uwzględnieniem całego wejścia)

hist.-det. $\leftrightsquigarrow$ zgadywanie oparte **wyłącznie** o już wczytaną część wejścia

Twierdzenie (Henzinger, Piterman [2006])

History-determinizm może zastępować determinizm przy rozwiązywaniu problemu syntezy.

$\mathcal{A}$ jest history-deterministic jeśli

$$\big(\sigma(), \sigma(w_0), \sigma(w_0 w_1), \ldots\big)$$

$$\exists_{\underbrace{\sigma\colon \Sigma^* \to Q}_{\text{sugestia/porada}}} \quad \forall_{w \in \mathrm{L}(\mathcal{A})} \quad \text{bieg } \sigma(w) \text{ jest akceptujący}$$

Automaty history-deterministic

determinizm $\leftrightsquigarrow$ sekwencyjna konstrukcja obliczenia

niedeterminizm $\leftrightsquigarrow$ zgadywanie (z uwzględnieniem całego wejścia)

hist.-det. $\leftrightsquigarrow$ zgadywanie oparte **wyłącznie** o
już wczytaną część wejścia

Twierdzenie (Henzinger, Piterman [2006])

History-determinizm może zastępować determinizm
przy rozwiązywaniu problemu syntezy.

$\mathcal{A}$ jest history-deterministic jeśli

$\big(\sigma(), \sigma(w_0), \sigma(w_0 w_1), \ldots\big)$

$$\underbrace{\exists_{\sigma\colon \Sigma^* \to Q}}_{\text{sugestia/porada}} \quad \underbrace{\forall_{w \in \mathrm{L}(\mathcal{A})} \quad \text{bieg } \sigma(w) \text{ jest akceptujący}}_{\sigma \text{ akceptuje kiedy tylko to jest możliwe}}$$

Zwięzłość automatów history-deterministic?

Zwięzłość automatów history-deterministic?

Twierdzenie (Löding [2011])

Każdy automat history-deterministic nad słowami skończonymi zawiera równoważny podautomat deterministyczny.

Zwięzłość automatów history-deterministic?

Twierdzenie (Löding [2011])

Każdy automat history-deterministic nad słowami skończonymi zawiera równoważny podautomat deterministyczny.

Hipoteza (Colcombet [2012])

To samo zachodzi dla słów nieskończonych.

Zwięzłość automatów history-deterministic?

Twierdzenie (Löding [2011])

Każdy automat history-deterministic nad słowami skończonymi zawiera równoważny podautomat deterministyczny.

Hipoteza (Colcombet [2012])

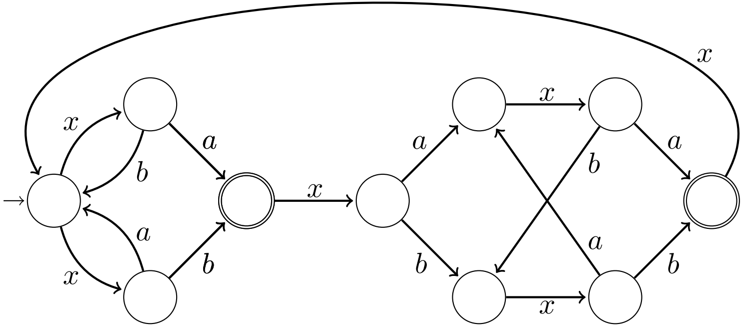
To samo zachodzi dla słów nieskończonych.

Przykład (Boker [2013])

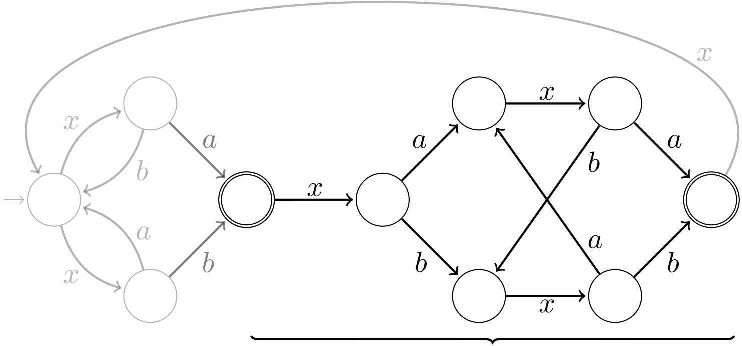
Hipoteza Colcombeta jest **fałszywa**.

Przykład Bokera

Przykład Bokera

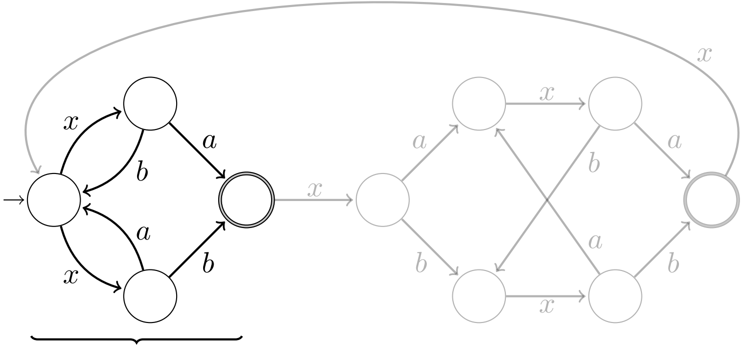


Przykład Bokera



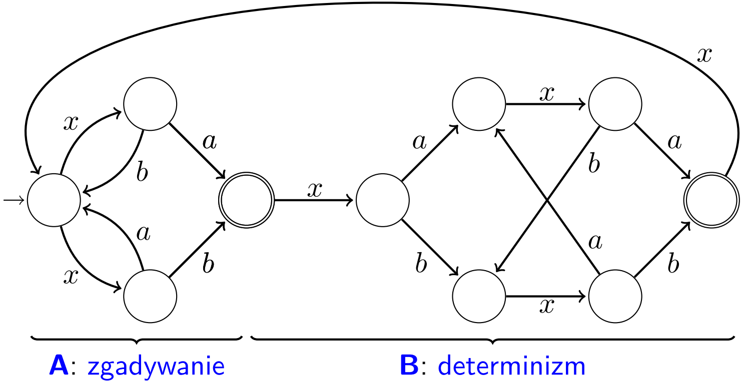
B: determinizm

Przykład Bokera

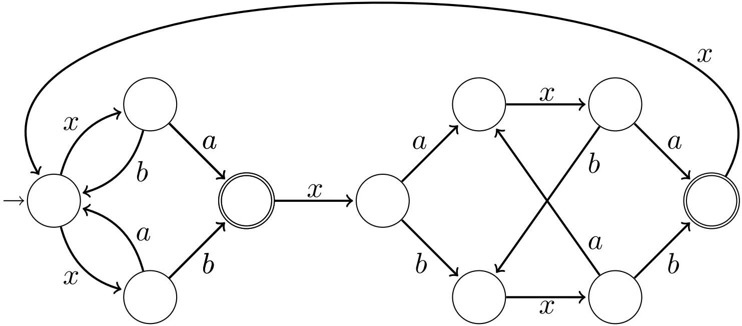


A: zgadywanie

Przykład Bokera



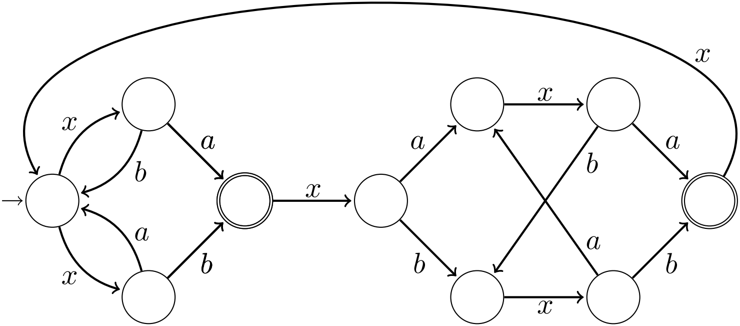
Przykład Bokera



A: zgadywanie   **B**: determinizm

nie ma równoważnego det. podautomatu

Przykład Bokera



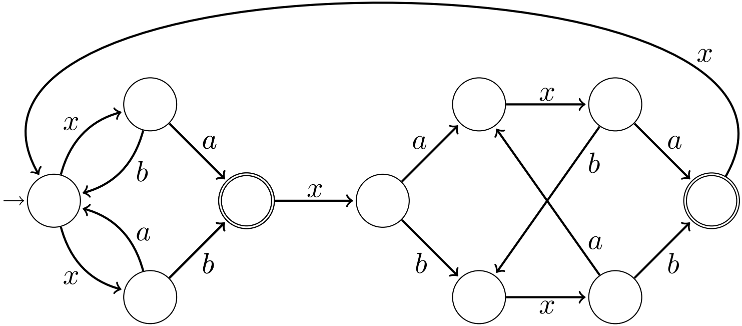
A: zgadywanie

B: determinizm

nie ma równoważnego det. podautomatu

$\sigma =$“zgaduj w **A** korzystając z **B**”

Przykład Bokera



A: zgadywanie　　　**B**: determinizm

nie ma równoważnego det. podautomatu

$\sigma =$"zgaduj w **A** korzystając z **B**"

$\big[$istnieje $|\mathcal{A}_{\mathrm{det}}| \sim |\mathcal{A}|\big]$

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $|\mathcal{A}_{\mathrm{det}}| \sim 2^{|\mathcal{A}|}$

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $|\mathcal{A}_{\mathrm{det}}| \sim 2^{|\mathcal{A}|}$

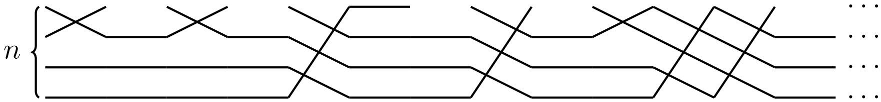
„Warkocze”:

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $|\mathcal{A}_{\mathrm{det}}| \sim 2^{|\mathcal{A}|}$

„Warkocze":

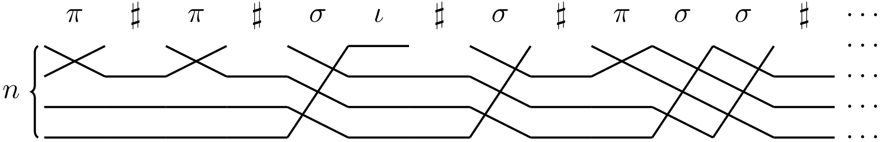
$$\pi \quad \sharp \quad \pi \quad \sharp \quad \sigma \quad \iota \quad \sharp \quad \sigma \quad \sharp \quad \pi \quad \sigma \quad \sigma \quad \sharp \quad \cdots$$



Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $|\mathcal{A}_{\mathrm{det}}| \sim 2^{|\mathcal{A}|}$

„Warkocze":

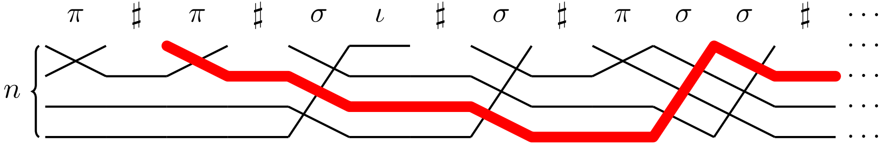


$\mathrm{L}(\mathcal{A}) =$"istnieje nieskończona ścieżka"

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $|\mathcal{A}_{\mathrm{det}}| \sim 2^{|\mathcal{A}|}$

„Warkocze":

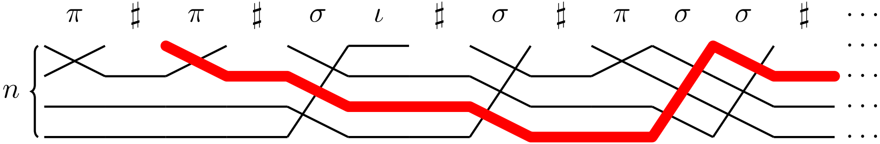


$\mathrm{L}(\mathcal{A}) =$"istnieje nieskończona ścieżka"

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $|\mathcal{A}_{\mathrm{det}}| \sim 2^{|\mathcal{A}|}$

„Warkocze”:



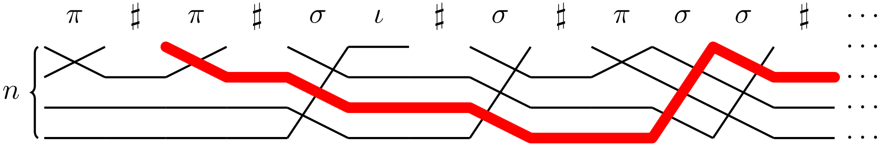
$\mathrm{L}(\mathcal{A}) =$“istnieje nieskończona ścieżka”

$\mathcal{A}$: zgadnij ścieżkę i śledź

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $|\mathcal{A}_{\mathrm{det}}| \sim 2^{|\mathcal{A}|}$

„Warkocze”:



$\mathrm{L}(\mathcal{A}) =$“istnieje nieskończona ścieżka”

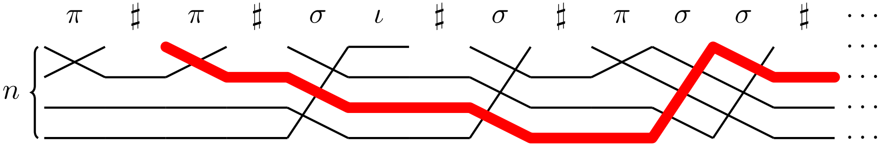
$\mathcal{A}$: zgadnij ścieżkę i śledź

gdy się zablokujesz, zgadnij nową ścieżkę (odrzucające przejście)

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $\left|\mathcal{A}_{\mathrm{det}}\right| \sim 2^{|\mathcal{A}|}$

„Warkocze”:



$\mathrm{L}(\mathcal{A}) =$"istnieje nieskończona ścieżka"

$\mathcal{A}$: zgadnij ścieżkę i śledź

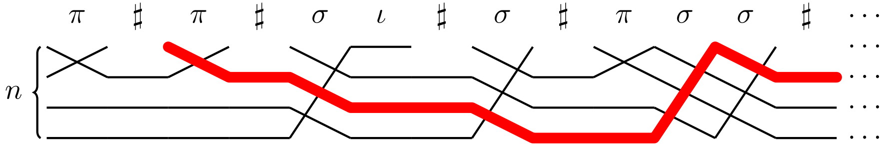
gdy się zablokujesz, zgadnij nową ścieżkę (odrzucające przejście)

$|\mathcal{A}| \sim n$

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $|\mathcal{A}_{\mathrm{det}}| \sim 2^{|\mathcal{A}|}$

„Warkocze":



$\mathrm{L}(\mathcal{A}) =$"istnieje nieskończona ścieżka"

$\mathcal{A}$: zgadnij ścieżkę i śledź

gdy się zablokujesz, zgadnij nową ścieżkę (odrzucające przejście)

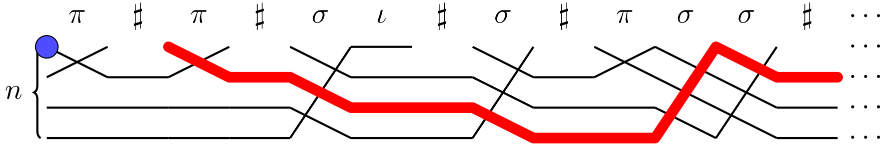
$|\mathcal{A}| \sim n$

σ: spróbuj najstarszą ścieżkę z dostępnych

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $|\mathcal{A}_{\mathrm{det}}| \sim 2^{|\mathcal{A}|}$

„Warkocze”:



$\mathrm{L}(\mathcal{A}) =$“istnieje nieskończona ścieżka”

$\mathcal{A}$: zgadnij ścieżkę i śledź

gdy się zablokujesz, zgadnij nową ścieżkę (odrzucające przejście)

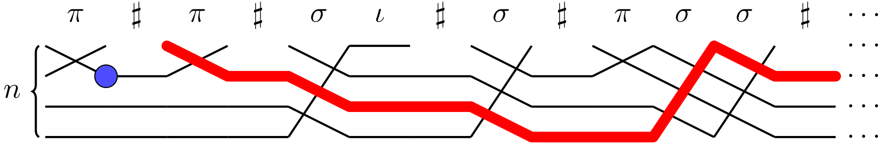
$|\mathcal{A}| \sim n$

σ: spróbuj najstarszą ścieżkę z dostępnych

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $|\mathcal{A}_{\mathrm{det}}| \sim 2^{|\mathcal{A}|}$

„Warkocze”:



$\mathrm{L}(\mathcal{A}) =$“istnieje nieskończona ścieżka”

$\mathcal{A}$: zgadnij ścieżkę i śledź

gdy się zablokujesz, zgadnij nową ścieżkę (odrzucające przejście)

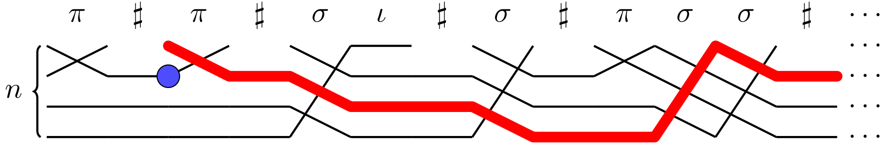
$|\mathcal{A}| \sim n$

σ: spróbuj najstarszą ścieżkę z dostępnych

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $|\mathcal{A}_{\mathrm{det}}| \sim 2^{|\mathcal{A}|}$

„Warkocze":



$\mathrm{L}(\mathcal{A}) =$"istnieje nieskończona ścieżka"

$\mathcal{A}$: zgadnij ścieżkę i śledź

gdy się zablokujesz, zgadnij nową ścieżkę (odrzucające przejście)

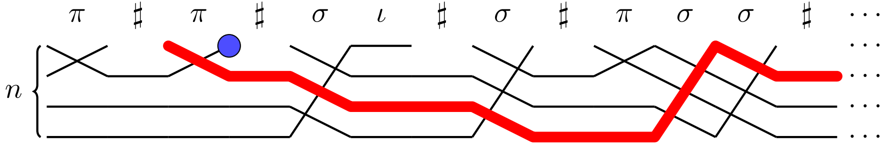
$|\mathcal{A}| \sim n$

σ: spróbuj najstarszą ścieżkę z dostępnych

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $\left|\mathcal{A}_{\mathrm{det}}\right| \sim 2^{|\mathcal{A}|}$

„Warkocze”:



$\mathrm{L}(\mathcal{A}) =$“istnieje nieskończona ścieżka”

$\mathcal{A}$: zgadnij ścieżkę i śledź

gdy się zablokujesz, zgadnij nową ścieżkę (odrzucające przejście)

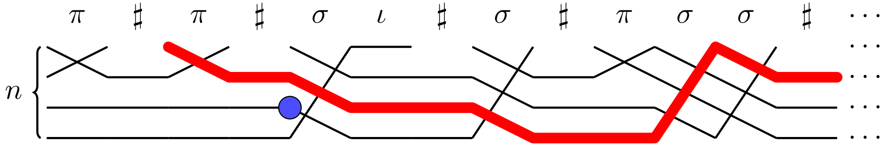
$|\mathcal{A}| \sim n$

σ: spróbuj najstarszą ścieżkę z dostępnych

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $|\mathcal{A}_{\mathrm{det}}| \sim 2^{|\mathcal{A}|}$

„Warkocze":



$\mathrm{L}(\mathcal{A}) =$"istnieje nieskończona ścieżka"

$\mathcal{A}$: zgadnij ścieżkę i śledź

gdy się zablokujesz, zgadnij nową ścieżkę (odrzucające przejście)

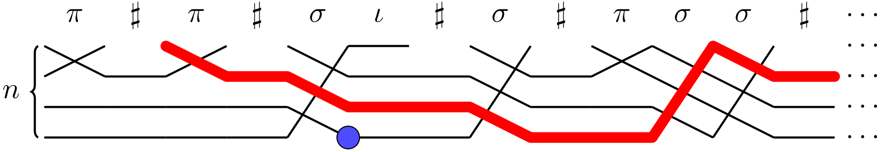
$|\mathcal{A}| \sim n$

σ: spróbuj najstarszą ścieżkę z dostępnych

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $|\mathcal{A}_{\mathrm{det}}| \sim 2^{|\mathcal{A}|}$

„Warkocze":



$\mathrm{L}(\mathcal{A}) =$"istnieje nieskończona ścieżka"

$\mathcal{A}$: zgadnij ścieżkę i śledź

gdy się zablokujesz, zgadnij nową ścieżkę (odrzucające przejście)

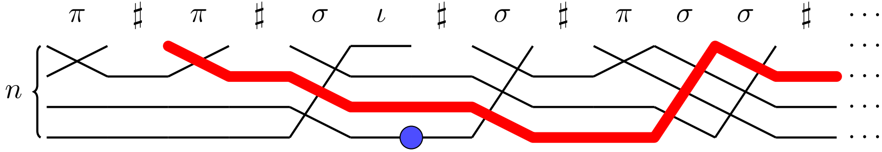
$|\mathcal{A}| \sim n$

σ: spróbuj najstarszą ścieżkę z dostępnych

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $|\mathcal{A}_{\mathrm{det}}| \sim 2^{|\mathcal{A}|}$

„Warkocze":



$\mathrm{L}(\mathcal{A}) =$"istnieje nieskończona ścieżka"

$\mathcal{A}$: zgadnij ścieżkę i śledź

gdy się zablokujesz, zgadnij nową ścieżkę (odrzucające przejście)

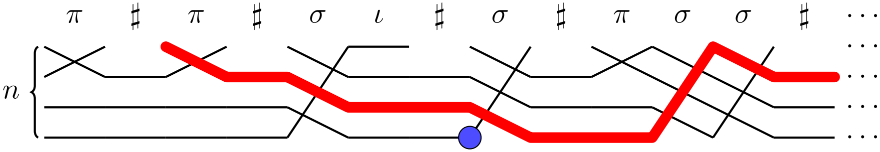
$|\mathcal{A}| \sim n$

σ: spróbuj najstarszą ścieżkę z dostępnych

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $|\mathcal{A}_{\mathrm{det}}| \sim 2^{|\mathcal{A}|}$

„Warkocze”:



$\mathrm{L}(\mathcal{A}) =$“istnieje nieskończona ścieżka”

$\mathcal{A}$: zgadnij ścieżkę i śledź

gdy się zablokujesz, zgadnij nową ścieżkę (odrzucające przejście)

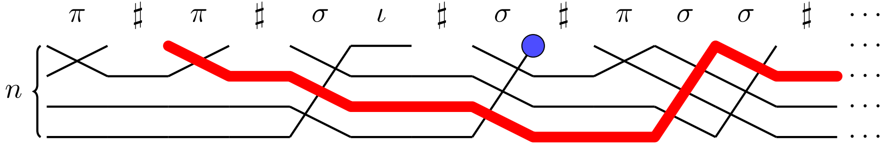
$|\mathcal{A}| \sim n$

σ: spróbuj najstarszą ścieżkę z dostępnych

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $|\mathcal{A}_{\mathrm{det}}| \sim 2^{|\mathcal{A}|}$

„Warkocze”:



$\mathrm{L}(\mathcal{A}) =$“istnieje nieskończona ścieżka”

$\mathcal{A}$: zgadnij ścieżkę i śledź

gdy się zablokujesz, zgadnij nową ścieżkę (odrzucające przejście)

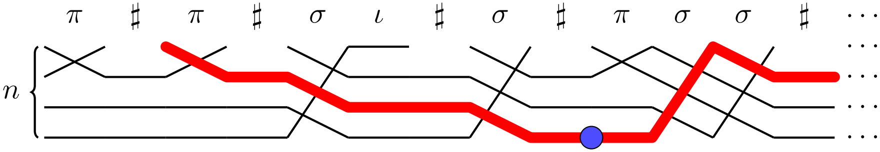
$|\mathcal{A}| \sim n$

σ: spróbuj najstarszą ścieżkę z dostępnych

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $|\mathcal{A}_{\mathrm{det}}| \sim 2^{|\mathcal{A}|}$

„Warkocze”:



$\mathrm{L}(\mathcal{A}) =$“istnieje nieskończona ścieżka”

$\mathcal{A}$: zgadnij ścieżkę i śledź

gdy się zablokujesz, zgadnij nową ścieżkę (odrzucające przejście)

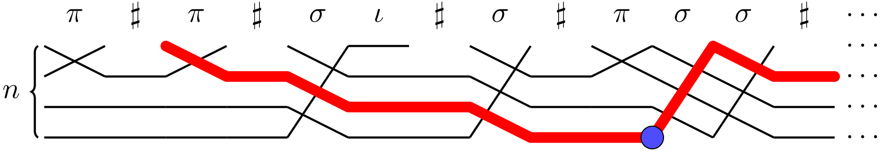
$|\mathcal{A}| \sim n$

σ: spróbuj najstarszą ścieżkę z dostępnych

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $|\mathcal{A}_{\mathrm{det}}| \sim 2^{|\mathcal{A}|}$

„Warkocze”:



$\mathrm{L}(\mathcal{A}) =$“istnieje nieskończona ścieżka”

$\mathcal{A}$: zgadnij ścieżkę i śledź

gdy się zablokujesz, zgadnij nową ścieżkę (odrzucające przejście)

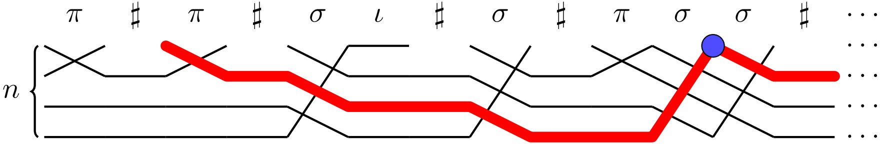
$|\mathcal{A}| \sim n$

σ: spróbuj najstarszą ścieżkę z dostępnych

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $|\mathcal{A}_{\mathrm{det}}| \sim 2^{|\mathcal{A}|}$

„Warkocze":



$\mathrm{L}(\mathcal{A}) =$"istnieje nieskończona ścieżka"

$\mathcal{A}$: zgadnij ścieżkę i śledź

gdy się zablokujesz, zgadnij nową ścieżkę (odrzucające przejście)

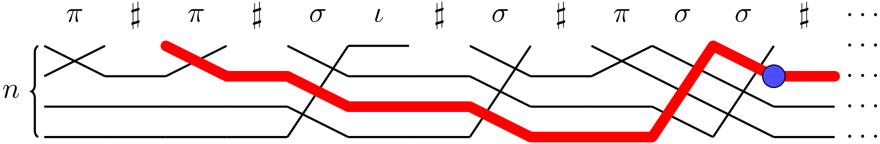
$|\mathcal{A}| \sim n$

σ: spróbuj najstarszą ścieżkę z dostępnych

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $\left|\mathcal{A}_{\mathrm{det}}\right| \sim 2^{|\mathcal{A}|}$

„Warkocze":



$\mathrm{L}(\mathcal{A}) =$"istnieje nieskończona ścieżka"

$\mathcal{A}$: zgadnij ścieżkę i śledź

gdy się zablokujesz, zgadnij nową ścieżkę (odrzucające przejście)

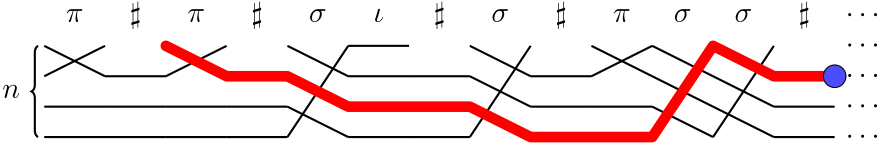
$|\mathcal{A}| \sim n$

σ: spróbuj najstarszą ścieżkę z dostępnych

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $|\mathcal{A}_{\mathrm{det}}| \sim 2^{|\mathcal{A}|}$

„Warkocze”:



$\mathrm{L}(\mathcal{A}) =$“istnieje nieskończona ścieżka”

$\mathcal{A}$: zgadnij ścieżkę i śledź

gdy się zablokujesz, zgadnij nową ścieżkę (odrzucające przejście)

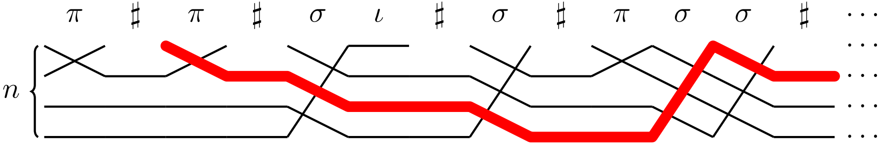
$|\mathcal{A}| \sim n$

σ: spróbuj najstarszą ścieżkę z dostępnych

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $|\mathcal{A}_{\mathrm{det}}| \sim 2^{|\mathcal{A}|}$

„Warkocze”:



$\mathrm{L}(\mathcal{A}) =$“istnieje nieskończona ścieżka”

$\mathcal{A}$: zgadnij ścieżkę i śledź

gdy się zablokujesz, zgadnij nową ścieżkę (odrzucające przejście)

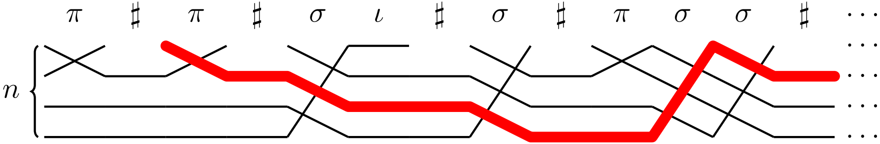
$|\mathcal{A}| \sim n$

σ: spróbuj najstarszą ścieżkę z dostępnych

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $|\mathcal{A}_{\mathrm{det}}| \sim 2^{|\mathcal{A}|}$

„Warkocze":



$\mathrm{L}(\mathcal{A}) =$"istnieje nieskończona ścieżka"

$\mathcal{A}$: zgadnij ścieżkę i śledź

gdy się zablokujesz, zgadnij nową ścieżkę (odrzucające przejście)

$|\mathcal{A}| \sim n$

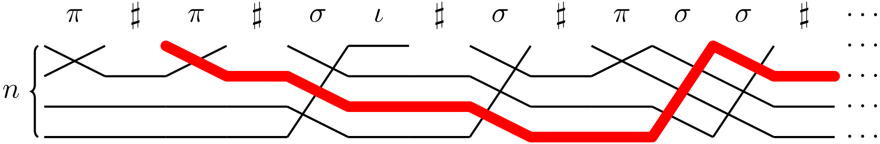
σ: spróbuj najstarszą ścieżkę z dostępnych

Lemat $|\mathcal{A}_{\mathrm{det}}| \geq 2^n$

Twierdzenie (Kuperberg, S. [2014])

Dla automatów co-Büchiego $|\mathcal{A}_{\mathrm{det}}| \sim 2^{|\mathcal{A}|}$

„Warkocze":



$\mathrm{L}(\mathcal{A}) =$"istnieje nieskończona ścieżka"

$\mathcal{A}$: zgadnij ścieżkę i śledź

gdy się zablokujesz, zgadnij nową ścieżkę (odrzucające przejście)

$|\mathcal{A}| \sim n$

σ: spróbuj najstarszą ścieżkę z dostępnych

Lemat $|\mathcal{A}_{\mathrm{det}}| \geq 2^n$

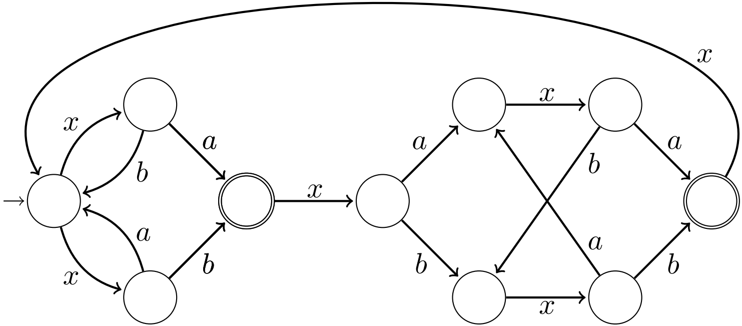
Dowód Aproksymacje biegów + argument zwartościowy

Twierdzenie (Kuperberg, S. [2015])

Dla automatów Büchiego $|\mathcal{A}_{\mathrm{det}}| \leqslant |\mathcal{A}|^2$

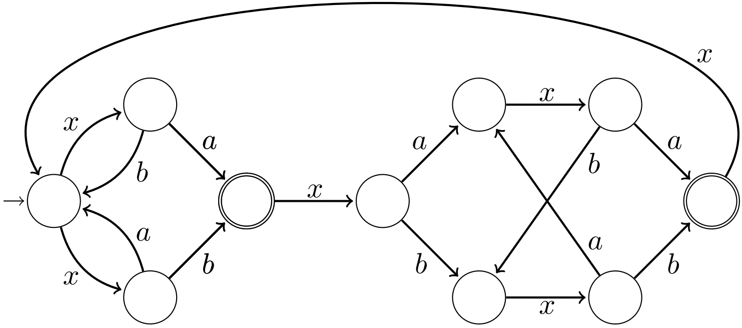
Twierdzenie (Kuperberg, S. [2015])

Dla automatów Büchiego $|\mathcal{A}_{\mathrm{det}}| \leqslant |\mathcal{A}|^2$



Twierdzenie (Kuperberg, S. [2015])

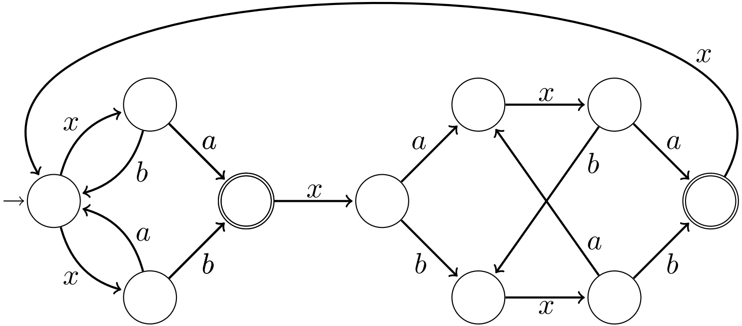
Dla automatów Büchiego $|\mathcal{A}_{\mathrm{det}}| \leqslant |\mathcal{A}|^2$



Skąd pochodzi σ?

Twierdzenie (Kuperberg, S. [2015])

Dla automatów Büchiego $|\mathcal{A}_{\mathrm{det}}| \leqslant |\mathcal{A}|^2$



Skąd pochodzi σ?

$\rightsquigarrow$ sygnatury Walukiewicza + iteracyjna normalizacja automatu

Determinizacja $\mathcal{A} \rightsquigarrow \mathcal{A}_{\mathrm{det}}$

Determinizacja $\mathcal{A} \rightsquigarrow \mathcal{A}_{\mathrm{det}}$

reachability	co-Büchi	$(1,3)$	$(0,3)$	$\cdots$
safety	Büchi	$(0,2)$	$(1,4)$	$\cdots$

Determinizacja $\mathcal{A} \rightsquigarrow \mathcal{A}_{\mathrm{det}}$

reachability	co-Büchi	$(1,3)$	$(0,3)$	$\cdots$
safety	Büchi	$(0,2)$	$(1,4)$	$\cdots$

$\underbrace{\text{safety} \quad \text{Büchi}}_{\textbf{wielomianowa}}$

Determinizacja $\mathcal{A} \rightsquigarrow \mathcal{A}_{\mathrm{det}}$

wykładnicza

reachability	co-Büchi	$(1,3)$	$(0,3)$	$\cdots$
safety	Büchi	$(0,2)$	$(1,4)$	$\cdots$

wielomianowa

Podziękowania

Podziękowania

Promotorowie

Mikołaj Bojańczyk i Igor Walukiewicz

Podziękowania

Promotorowie

Mikołaj Bojańczyk i Igor Walukiewicz

Współautorzy

M. Bilkowski
M. Bojańczyk
U. Boker
A. Facchini
N. Fijalkow
O. Finkel
T. Gogacz
F. Horn
S. Hummel
T. Idziaszek
L. Kołodziejczyk
D. Kuperberg
O. Kupferman
H. Michalewski
M. Mio
F. Murlak
D. Niwiński
P. Pradic
M. Przybyłko
A. Rabinovich
A. Radziwończyk-Syta
S. Toruńczyk
I. Walukiewicz