

Języki, automaty i obliczenia

Wykład 14: Klasa PSPACE. Języki kontekstowe

Sławomir Lasota

Uniwersytet Warszawski

3 czerwca 2026 $\mapsto$ 10 czerwca 2026

1 NP-zupełność

2 PSPACE

3 Języki kontekstowe

SAT jest NP-zupełny (dowód)

Niech $\mathcal{M}$ – maszyna niedet. działająca w czasie n^c . Pokażemy $L(\mathcal{M}) \leq_p \text{SAT}$.

funkcja obliczalna w czasie wiel.:	$w = a_1 \dots a_n$	$\mapsto$	$\phi_{\mathcal{M},w}$
poprawność:	$w \in L(\mathcal{M})$	$\iff$	$\phi_{\mathcal{M},w}$ spełnialna

Zmienne:

- $t^{i,j,a}$ – po i krokach, na pozycji j taśmy jest symbol a
- $s^{i,q}$ – po i krokach, stan maszyny to q
- $g^{i,j}$ – po i krokach, głowica maszyny jest na pozycji j taśmy

$\bigwedge_{j \leq |w|} t^{0,j,a_j} \wedge \bigwedge_{j > |w|} t^{0,j,\mathbb{B}} \wedge s^{0,q_0} \wedge g^{0,1}$ (konfiguracja początkowa)

$\bigwedge_{i,j} \left(\bigvee_a t^{i,j,a} \wedge \bigwedge_{a \neq b} \neg(t^{i,j,a} \wedge t^{i,j,b}) \right) \wedge \dots$ (niesprzeczność)

$\bigvee_i s^{i,q_{\text{tak}}}$ (akceptacja)

$\bigwedge_{i,j} \bigwedge_{a,q} t^{i,j,a} \wedge s^{i,q} \wedge g^{i,j} \implies$ przejścia z (q, a) :

$t^{i+1,j,a'} \wedge s^{i+1,q'} \wedge g^{i+1,j} \quad \vee \quad (q, a, q', a', \circlearrowright)$

$t^{i+1,j,a''} \wedge s^{i+1,q''} \wedge g^{i+1,j-1} \quad (q, a, q'', a'', \leftarrow)$

$\bigwedge_{i,j} \bigwedge_a t^{i,j,a} \wedge \neg g^{i,j} \implies t^{i+1,j,a}$

co-NP = problemy, których dopełnienie jest w NP

Problem L jest *co-NP-trudny* jeśli każdy $K \in \text{co-NP}$ redukuje się wielomianowo do L .

Problem L jest *co-NP-zupełny* jeśli jest co-NP-trudny i należy do co-NP.

Fakt

Jeśli $K \leq_p L$ i K jest co-NP-trudny to L jest co-NP-trudny.

Problem tautologii zdaniowej

Dane:	formuła zdaniowa ϕ .
Wynik:	czy ϕ jest tautologią?

Wniosek

Problem tautologii zdaniowej jest co-NP-zupełny.

Problemy w NP, o których nie wiemy ani że są NP-zupełne, ani że są w P:

Izomorfizm grafów

Dane: Dwa grafy G , H .

Wynik: Czy G i H są izomorficzne?

Gra parzystości:

- graf skierowany, wierzchołek startowy
- każdy wierzchołek należy do jednego z graczy: Parzystego lub Nieparzystego
- wierzchołki etykietowane liczbami
- gramy do pierwszej powtórki (cyklu)
- zwycięzca określony przez parzystość największej liczby na cyklu

Kto wygrywa grę parzystości?

Dane: gra parzystości.

Wynik: czy Parzysty ma strategię wygrywającą?

Problem należy do $NP \cap co-NP$.

Pierwszość

Dane: liczba naturalna $n \in \mathbb{N}$, reprezentowana binarnie.
Wynik: czy n jest liczbą pierwszą?

Twierdzenie (Agrawal, Kayal, Saxena 2004)

Pierwszość jest w P.

nagroda Gödla '2006 !

2-SAT

Dane: formuła zdaniowa ϕ w postaci 2-CNF.
Wynik: czy ϕ jest spełnialna?

np. $\phi \equiv (x \vee y) \wedge (\neg y \vee \neg z)$

Skojarzenie doskonałe

Dane: graf dwudzielny G .
Wynik: czy G ma skojarzenie doskonałe?

1 NP-zupełność

2 PSPACE

3 Języki kontekstowe

Klasy złożoności z dokładnością do wielomianu:

...

$$\text{EXPSPACE} = \bigcup_p \text{DSpace}(2^{p(n)}) = \bigcup_p \text{NSpace}(2^{p(n)})$$

$$\text{NEXPTIME} = \bigcup_p \text{NTIME}(2^{p(n)})$$

$$\text{EXPTIME} = \bigcup_p \text{DTIME}(2^{p(n)})$$

$$\text{PSPACE} = \bigcup_p \text{DSpace}(p(n)) = \bigcup_p \text{NSpace}(p(n))$$

$$NP = \bigcup_p \text{NTIME}(p(n))$$

$$P = \bigcup_p \text{DTIME}(p(n))$$

Twierdzenie (Savitch 1970)

$$NSPACE(f(n)) \subseteq DSPACE(f(n)^2).$$

założenie!

Dowód:

Niech $\mathcal{M}$ – maszyna niedet. działająca w pamięci $f(n)$. Zaprojektujemy algorytm deterministyczny sprawdzający czy $c_0 \rightarrow_{\mathcal{M}}^* c_{\text{tak}}$, dla konfiguracji akceptującej c_{tak} .

Liczba konfiguracji $\leq 2^m$, $m = \mathcal{O}(f(n))$, więc wystarczy sprawdzić czy $c_0 \rightarrow_{\mathcal{M}}^{\leq 2^m} c_{\text{tak}}$.

Algorytm

```
bool p(c, c', k) {  
    if (k == 0) { return ((c==c') || c →ℳ c'); }  
    else {  
        for each c'' {  
            if (p(c, c'', k-1) && p(c'', c', k-1)) return true;  
        }  
        return false;  
    }  
}
```

Pamięć: $m \cdot f(n) = \mathcal{O}(f(n)^2)$.

Wniosek

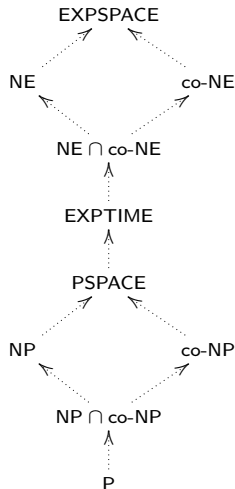
$$NPSPACE = PSPACE.$$

Ale nie wiemy, czy $NSPACE(n) = DSPACE(n)$.

$co-C$ = problemy, których dopełnienie należy do C

Wniosek

$$co-PSPACE = PSPACE.$$



Wiemy, że

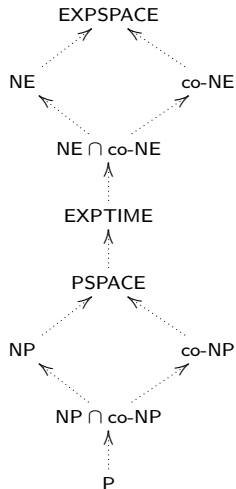
$P \neq \text{EXPTIME} \neq 2\text{-EXPTIME} \neq \dots$

$\text{NP} \neq \text{NEXPTIME} \neq 2\text{-NEXPTIME} \neq \dots$

$\text{PSPACE} \neq \text{EXPSPACE} \neq \dots$

$P \neq \text{NP} ?$

$P \neq \text{PSPACE} ?$



Problem spełnialności kwantyfikowanej formuły zdaniowej (QBF)

Dane: formuła postaci $\forall x_1 \exists x_2 \forall x_3 \dots \phi$.

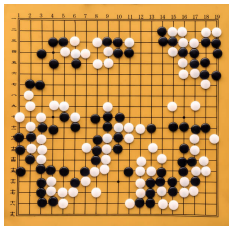
Wynik: czy formuła jest spełnialna (tautologią)?

np. $\forall x \exists y. (x \wedge y) \vee (x \wedge (\neg y \vee z))$

Kto wygrywa w Go?

Dane: pozycja w grze Go na dowolnie dużej planszy.

Wynik: czy białe wygrywają?



- Uniwersalność języka regularnego: $L = A^*$?
- Równość dwóch języków regularnych: $L_1 = L_2$?
- Niepustość przecięcia języków regularnych: $L_1 \cap \dots \cap L_k \neq \emptyset$?
- Niepustość wyrażenia regularnego z operacją przecięcia: $L \neq \emptyset$?
- ...

1 NP-zupełność

2 PSPACE

3 Języki kontekstowe

typ	języki	automaty	gramatyki
typ 0	częściowo rozstrzygalne	maszyny Turinga	dowolne
typ 1	kontekstowe	?	?
typ 2	bezkontekstowe	automaty ze stosem	bezkontekstowe
typ 3	regularne	automaty skończone	regularne

Gramatyka $\mathcal{G} = (A, N, S, \alpha)$ jest *kontekstowa* jeśli produkcje są postaci:

- $S \rightarrow \varepsilon$
- $uXv \rightarrow uwv \quad X \in N, \quad u, v, w \in (A \cup N)^*, \quad w \neq \varepsilon, \quad S \notin uwv$

Pytanie

Gramatyka $\mathcal{G}$:

$$\begin{array}{llll} A = \{a, b, c\} & S \rightarrow \varepsilon & X \rightarrow ABXc & BA \rightarrow AB \quad A \rightarrow a \\ N = \{S, X, A, B\} & S \rightarrow X & X \rightarrow Abc & Bb \rightarrow bb \end{array}$$

$$L(\mathcal{G}) = \{a^n b^n c^n : n \in \mathbb{N}\}$$

Jak przerobić tę gramatykę na gramatykę kontekstową?

Odpowiedź

Pomysł jest następujący:

$$\begin{array}{ll} BA \rightarrow B\bar{A} & \bar{B}\bar{A} \rightarrow \bar{B}B \\ B\bar{A} \rightarrow \bar{B}\bar{A} & \bar{B}B \rightarrow AB \end{array}$$

Gramatyka $\mathcal{G} = (A, N, S, \alpha)$ jest *kontekstowa* jeśli produkcje są postaci:

- $S \rightarrow \varepsilon$
- $uXv \rightarrow uwv \quad X \in N, \quad u, v, w \in (A \cup N)^*, \quad w \neq \varepsilon, \quad S \notin uwv$

Gramatyka $\mathcal{G} = (A, N, S, \alpha)$ jest *monotoniczna* jeśli produkcje są postaci:

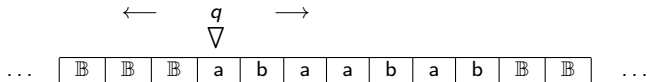
- $S \rightarrow \varepsilon$
- $v \rightarrow w \quad |v| \leq |w|, \quad S \notin w$

Twierdzenie

Gramatyki kontekstowe $\equiv$ gramatyki monotoniczne.

Dowód:

j.w.



Maszyna Turinga jest *liniowo ograniczona* jeśli nigdy nie pisze na pozycji taśmy zajętej przez symbol $\mathbb{B}$. Czyli zabraniamy przejść postaci:

$$(q, \mathbb{B}, q', a, x) \quad a \neq \mathbb{B}$$

(Automat dwukierunkowy, który może pisać po taśmie.)

Twierdzenie

Gramatyki monotoniczne $\equiv$ niedeterministyczne maszyny liniowo ograniczone.

Fakt

Problem słów dla gramatyk kontekstowych jest PSPACE-zupełny.

Problem stopu dla liniowo ograniczonych maszyn Turinga jest PSPACE-zupełny.

Twierdzenie

Problem (nie)pustości jest nierozstrzygalny dla języków kontekstowych.

Dowód:

Dla ustalonej instancji PCP, zbiór rozwiązań jest językiem kontekstowym.

funkcja obliczalna:	instancja PCP x	$\mapsto$	$\mathcal{G}_x$
poprawność:	x ma rozwiązanie	$\iff$	$L(\mathcal{G}_x) \neq \emptyset$

typ	języki	automaty	gramatyki	klasa złożoności
typ 0	częściowo rozstrzygalne	maszyny Turinga	dowolne	
typ 1	kontekstowe	maszyny liniowo ograniczone	kontekstowe / monotoniczne	$\text{NSPACE}(n)$
typ 2	bezkontekstowe	automaty ze stosem	bezkontekstowe	$\subseteq \text{PTIME}$
typ 3	regularne	automaty skończone	regularne	$\text{NSPACE}(1)/\text{DSPACE}(1)$

W następnym odcinku:

?