

Języki, automaty i obliczenia

Wykład 13: Wielomianowy czas i pamięć

Sławomir Lasota

Uniwersytet Warszawski

8 czerwca 2016

- 1 Klasy złożoności
- 2 Redukcje wielomianowe
- 3 Problemy NP-zupełne
- 4 Problemy PSPACE-zupełne

Złożoność problemów z dokładnością do wielomianu:

$$\begin{array}{l}
 \dots \\
 \text{EXPSPACE} = \bigcup_c \text{DSPACE}(2^{n^c}) \quad = \bigcup_c \text{NSPACE}(2^{n^c}) \\
 \text{NEXPTIME} = \bigcup_c \text{NTIME}(2^{n^c}) \\
 \text{EXPTIME} = \bigcup_c \text{DTIME}(2^{n^c}) \\
 \text{PSPACE} = \bigcup_c \text{DSPACE}(n^c) \quad = \bigcup_c \text{NSPACE}(n^c) \\
 NP = \quad \quad \quad \text{NPTIME} = \bigcup_c \text{NTIME}(n^c) \\
 P = \quad \quad \quad \text{PTIME} = \bigcup_c \text{DTIME}(n^c) \\
 NL = \quad \quad \quad \text{NLOGSPACE} = \text{NSPACE}(\log(n)) \\
 L = \quad \quad \quad \text{LOGSPACE} = \text{DSPACE}(\log(n))
 \end{array}$$

Twierdzenie (Savitch 1970)

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

założenie!

Dowód:

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

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

Algorytm

```
bool p(c, c', k) {  
    if (k == 0) { return ((c==c') || c  $\rightarrow_{\mathcal{M}}$  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

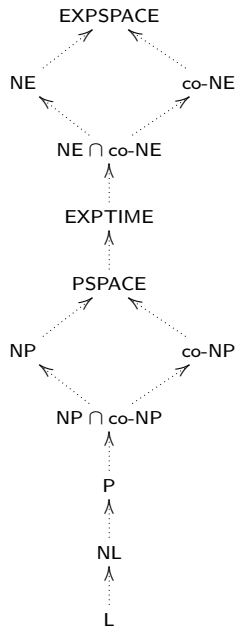
Wniosek

 $co-NPSPACE = NPSPACE$.

Twierdzenie (Immerman, Szelepcsényi 1987)

 $co-NL = NL$.

Wniosek

 $co-NSPACE(n) = NSPACE(n)$.

Wiemy, że

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

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

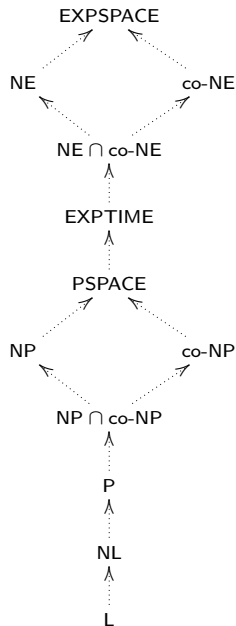
$\text{NL} \neq \text{PSPACE} \neq \text{EXSPACE} \neq \dots$

Czy $P = \text{NP}$?

Pytanie

Dlaczego to pytanie jest takie ważne?

Czy $P = \text{PSPACE}$?



...

$$\text{EXPSpace} = \bigcup_c \text{DSpace}(2^{n^c}) = \bigcup_c \text{NSpace}(2^{n^c}) = \text{AEXPTIME}$$

$$\text{NEXPTIME} = \bigcup_c \text{NTIME}(2^{n^c})$$

$$\text{EXPTIME} = \bigcup_c \text{DTIME}(2^{n^c}) = \text{APSPACE}$$

$$\text{PSPACE} = \bigcup_c \text{DSpace}(n^c) = \bigcup_c \text{NSpace}(n^c) = \text{APTIME}$$

$$NP = \text{NPTIME} = \bigcup_c \text{NTIME}(n^c)$$

$$P = \text{PTIME} = \bigcup_c \text{DTIME}(n^c) = \text{ALOGSPACE}$$

$$NL = \text{NLOGSPACE} = \text{NSpace}(\log(n))$$

$$L = \text{LOGSPACE} = \text{DSpace}(\log(n))$$

- 1 Klasy złożoności
- 2 Redukcje wielomianowe**
- 3 Problemy NP-zupełne
- 4 Problemy PSPACE-zupełne

Problem $K \subseteq A^*$ *redukuje się wielomianowo* do problemu $L \subseteq B^*$ (ozn. $K \leq_p L$)
jeśli istnieje funkcja obliczalna w czasie wielomianowym

$$f : A^* \rightarrow B^*$$

taka, że

$$w \in K \iff f(w) \in L, \quad \text{dla każdego } w \in A^*.$$

Fakt

Niech \mathcal{C} – klasa złożoności z listy poniżej. Jeśli $K \leq_p L$ i $L \in \mathcal{C}$ to $K \in \mathcal{C}$.

$$\begin{array}{rcl}
 \dots & & \\
 \text{EXPSpace} & = \bigcup_c \text{DSpace}(2^{n^c}) & = \bigcup_c \text{NSpace}(2^{n^c}) \\
 \text{NEXPTIME} & = \bigcup_c \text{NTIME}(2^{n^c}) & \\
 \text{EXPTIME} & = \bigcup_c \text{DTIME}(2^{n^c}) & \\
 \text{PSPACE} & = \bigcup_c \text{DSpace}(n^c) & = \bigcup_c \text{NSpace}(n^c) \\
 NP & = & \text{NPTIME} = \bigcup_c \text{NTIME}(n^c) \\
 P & = & \text{PTIME} = \bigcup_c \text{DTIME}(n^c)
 \end{array}$$

(W przypadku klas NL, L trzeba stosować redukcje w pamięci logarytmicznej.)

Problem L jest *\mathcal{C} -trudny* jeśli każdy problem $K \in \mathcal{C}$ redukuje się wielomianowo do L .

Problem L jest *\mathcal{C} -zupełny* jeśli jest \mathcal{C} -trudny i należy do \mathcal{C} .

Fakt

Jeśli $K \leq_p L$ i K jest \mathcal{C} -trudny to L jest \mathcal{C} -trudny.

$$\begin{aligned} & \dots \\ \text{EXPSPACE} &= \bigcup_c \text{DSPACE}(2^{n^c}) &= \bigcup_c \text{NSPACE}(2^{n^c}) \\ \text{NEXPTIME} &= \bigcup_c \text{NTIME}(2^{n^c}) \\ \text{EXPTIME} &= \bigcup_c \text{DTIME}(2^{n^c}) \\ \text{PSPACE} &= \bigcup_c \text{DSPACE}(n^c) &= \bigcup_c \text{NSPACE}(n^c) \\ \text{NP} &= \text{NPTIME} = \bigcup_c \text{NTIME}(n^c) \end{aligned}$$

(W przypadku klas P, NL trzeba stosować redukcje w pamięci logarytmicznej.)

- 1 Klasy złożoności
- 2 Redukcje wielomianowe
- 3 Problemy NP-zupełne**
- 4 Problemy PSPACE-zupełne

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

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

Fakt

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

Problem spełnialności formuły zdaniowej (SAT)

Dane: formuła zdaniowa ϕ

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

Wynik: czy ϕ jest spełnialna?

Twierdzenie (Cook 1971, Levin 1973)

SAT jest NP-zupełny.

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)

$\bigvee_i s^{i,q_F}$ (akceptacja)

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

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

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

3-kolorowalność

Dane: graf nieskierowany G .

Wynik: czy da się pokolorować wierzchołki G trzema kolorami tak, żeby kolory sąsiadów były różne?

Problem plecakowy (szczególny przypadek)

Dane: zbiór liczb $\{n_1, \dots, n_k\}$ i liczba m , reprezentowane binarnie.

Wynik: czy istnieje podzbiór $\{n_{i_1}, \dots, n_{i_l}\}$ taki, że $n_{i_1} + \dots + n_{i_l} = m$?

PCP z ograniczeniem

Dane: ciąg par słów $(w_1, v_1), \dots, (w_n, v_n)$ i liczba k reprezentowana unarnie.

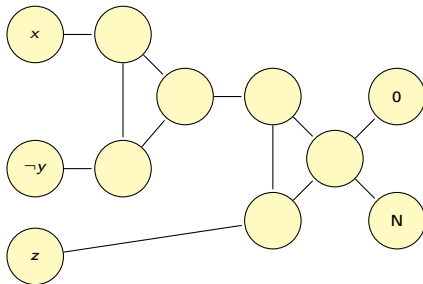
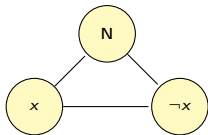
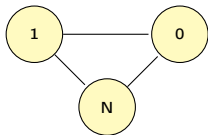
Wynik: czy istnieje niepusty ciąg (i_1, \dots, i_m) , $m \leq k$, t. że $w_{i_1} \dots w_{i_m} = v_{i_1} \dots v_{i_m}$?

Problem 3-kolorowalności jest NP-zupełny (dowód)

Redukcja 3-SAT \leq_p 3-kolorowalność.

funkcja obliczalna w czasie wiel.: ϕ w postaci 3-CNF $\mapsto G_\phi$
poprawność: ϕ spełnialna $\iff G_\phi$ 3-kolorowalny

$$\phi \equiv (x \vee \neg y \vee z) \wedge (\neg x \vee \neg z \vee u) \wedge \dots$$



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:

- gracze: Parzysty, Nieparzysty
- wierzchołki grafu etykietowane liczbami
- gramy do pierwszej powtórki (cyklu)
- zwycięzca określony przez parzystość największej liczby na cyklu

Kto wygrywa grę parzystości?

Dane: graf gry parzystości

Wynik: czy Parzysty ma strategię wygrywającą?

Problem należy do $NP \cap \text{co-NP}$.

Pierwszość

Dane: liczba naturalna $n \in \mathbb{N}$, reprezentowana binarnie.

Wynik: czy n jest liczbą pierwszą?

Twierdzenie (Agrawal, Kayal, Saxena 2004)

Problem pierwszości jest w P.

- 1 Klasy złożoności
- 2 Redukcje wielomianowe
- 3 Problemy NP-zupełne
- 4 Problemy PSPACE-zupełne**

Problem spełnialności kwantyfikowanej formuły zdaniowej (QBF)

Dane: formuła postaci $\forall x_1 \exists x_2 \forall x_3 \dots \phi$ np. $\forall x \exists y. (x \wedge y) \vee (x \wedge (\neg y \vee z))$.

Wynik: czy ϕ jest spełnialna (tautologią)?

Kto wygrywa w Go (bez ko)?

Dane: pozycja w grze Go na dowolnie dużej planszy.

Wynik: czy białe wygrywają?

A gdy ko jest dozwolone?

- Uniwersalność wyrażenia regularnego (albo automatu niedet.).
- Równoważność dwóch wyrażen regularnych (albo automatów niedet.).
- Niepustość uogólnionego wyrażenia regularnego (z operacją dopełnienia).
- Niepustość przecięcia wyrażen regularnych (albo automatów niedet.).
- Czy dane wyrażenie regularne jest równoważne wyrażeniu bezgwiazdkowemu (z operacją dopełnienia).
- Problem stopu dla maszyn liniowo ograniczonych.
- Problem słów dla gramatyk kontekstowych.
- ...