

IB107 Vyčísitelnost a složitost

polynomiální redukce, SAT, 3SAT, CLIQUE, VERTEX-COVER

Jan Strejček

Fakulta informatiky
Masarykova univerzita

Definice (polynomiální redukce)

Nechť $A \subseteq \Sigma^*$ a $B \subseteq \Phi^*$ jsou jazyky. Řekneme, že A se *polynomiálně redukuje* na B , píšeme $A \leq_p B$, právě když existuje totální funkce $f : \Sigma^* \rightarrow \Phi^*$, která je vyčíslitelná deterministickým Turingovým strojem pracujícím v polynomiálním čase a taková, že

$$w \in A \iff f(w) \in B.$$

Funkci f nazveme *polynomiální redukcí* A na B .

Platí $A \leq_p B \implies \bar{A} \leq_p \bar{B}$.

Platí $A \leq_p B$ a $B \leq_p C \implies A \leq_p C$ (tj. \leq_p je tranzitivní).

Věta

Nechť $A \leq_p B$.

- $B \in P \implies A \in P$
- $B \in NP \implies A \in NP$

Důkaz: Nechť f je redukce A na B v polynomiálním čase a \mathcal{M}_B je TM rozhodující B . Stroj \mathcal{M}_A rozhodující A na vstupu w

1 spočítá $f(w)$

2 spustí \mathcal{M}_B na vstupu $f(w)$ a vrátí stejný výsledek jako \mathcal{M}_B

Je-li \mathcal{M}_B deterministický, pak je i \mathcal{M}_A deterministický. Krok 1 lze provést v polynomiálním čase vzhledem k $|w|$, krok 2 v polynomiálním čase vzhledem k $|f(w)|$, což je polynomiální i vzhledem k $|w|$. \mathcal{M}_A tedy pracuje v polynomiálním čase. ■

Definice (těžká a úplná množina ve složitostní třídě)

Nechť \mathbb{C} je složitostní třída splňující $NP \subseteq \mathbb{C}$. Jazyk L nazveme **\mathbb{C} -těžký**, právě když pro každý jazyk $L' \in \mathbb{C}$ platí $L' \leq_p L$. Je-li navíc $L \in \mathbb{C}$, pak L nazýváme **\mathbb{C} -úplný** nebo **úplný ve třídě \mathbb{C}** .

Věta (Cookova-Levinova věta)

$SAT = \{\langle \varphi \rangle \mid \varphi \text{ je splnitelná výroková formule}\}$ je NP-úplný.

Důkaz: $SAT \in NP$ jsme již dokázali.

Ukážeme, že SAT je NP-těžký, tj. $A \in NP \implies A \leq_p SAT$.

Nechť \mathcal{M} je nedeterministický TM rozhodující A v čase $p(n)$, kde p je polynom. Pro každé slovo w sestrojíme výrokovou formuli Φ , která je splnitelná, právě když stroj \mathcal{M} má akceptující výpočet na w .

SAT je NP-úplný

Každý výpočet stroje \mathcal{M} pracujícího v čase $p(n)$ na slově w délky n lze reprezentovat tabulkou:

Vytvoříme Φ , aby platilo:

Φ je splnitelné \iff existuje tabulka reprezentující akceptující výpočet na w

SAT je NP-úplný

$$\Phi = \Phi_{\text{cell}} \wedge \Phi_{\text{start}} \wedge \Phi_{\text{move}} \wedge \Phi_{\text{accept}}$$

Φ_{cell} = “každé $x_{i,j,s}$ platí \iff v tabulce na pozici i, j je symbol s ,
kde $s \in C = Q \cup \Gamma \cup \{\#\}$ ”

$$\Phi_{\text{cell}} = \bigwedge_{\substack{1 \leq i \leq p(n)+1 \\ 1 \leq j \leq p(n)+3}} \left[\left(\bigvee_{s \in C} x_{i,j,s} \right) \wedge \left(\bigwedge_{\substack{s, t \in C \\ s \neq t}} \neg(x_{i,j,s} \wedge x_{i,j,t}) \right) \right]$$

$$|\Phi_{\text{cell}}| \in \mathcal{O}(p^2(n))$$

SAT je NP-úplný

$$\Phi = \Phi_{\text{cell}} \wedge \Phi_{\text{start}} \wedge \Phi_{\text{move}} \wedge \Phi_{\text{accept}}$$

Φ_{start} = “na prvním řádku je iniciální konfigurace pro $w = w_1 \dots w_n$ ”

$$\begin{aligned} \Phi_{\text{start}} = & x_{1,1,\#} \wedge x_{1,2,q_0} \wedge x_{1,3,\triangleright} \wedge \\ & x_{1,4,w_1} \wedge x_{1,5,w_2} \wedge \dots \wedge x_{1,n+3,w_n} \wedge \\ & x_{1,n+4,\sqcup} \wedge x_{1,n+5,\sqcup} \wedge \dots \wedge x_{1,p(n)+2,\sqcup} \wedge x_{1,p(n)+3,\#} \end{aligned}$$

$$|\Phi_{\text{start}}| \in \mathcal{O}(p(n))$$

SAT je NP-úplný

$$\Phi = \Phi_{\text{cell}} \wedge \Phi_{\text{start}} \wedge \Phi_{\text{move}} \wedge \Phi_{\text{accept}}$$

Φ_{move} = “každé dva po sobě jdoucí řádky odpovídají kroku výpočtu”

popíšeme pomocí “legálních oken” 2×3

příklady legálních oken pro $\delta(q_1, b) = \{(q_2, c, L), (q_2, a, R)\}$:

SAT je NP-úplný

$$\Phi = \Phi_{\text{cell}} \wedge \Phi_{\text{start}} \wedge \Phi_{\text{move}} \wedge \Phi_{\text{accept}}$$

Φ_{move} = “každé okno 2×3 v tabulce je legální” (okna se překrývají)

$$\Phi_{\text{move}} = \bigwedge_{\substack{1 \leq i < p(n)+1 \\ 1 < j < p(n)+3}} \left(\bigvee_{\text{legální okno}} \begin{array}{c} x_{i,j-1,a_1} \wedge x_{i,j,a_2} \wedge x_{i,j+1,a_3} \\ x_{i+1,j-1,a_4} \wedge x_{i+1,j,a_5} \wedge x_{i+1,j+1,a_6} \end{array} \right)$$

a_1	a_2	a_3
a_4	a_5	a_6

$$|\Phi_{\text{move}}| \in \mathcal{O}(p^2(n))$$

SAT je NP-úplný

$$\Phi = \Phi_{\text{cell}} \wedge \Phi_{\text{start}} \wedge \Phi_{\text{move}} \wedge \Phi_{\text{accept}}$$

$\Phi_{\text{accept}} =$ “v tabulce je stav q_{acc} ”

$$\Phi_{\text{accept}} = \bigvee_{\substack{1 < i \leq p(n)+1 \\ 1 < j < p(n)+3}} x_{i,j,q_{\text{acc}}}$$

$$|\Phi_{\text{accept}}| \in \mathcal{O}(p^2(n))$$

SAT je NP-úplný

$$\Phi = \Phi_{\text{cell}} \wedge \Phi_{\text{start}} \wedge \Phi_{\text{move}} \wedge \Phi_{\text{accept}}$$

$$|\Phi| = \mathcal{O}(p^2(n)) + \mathcal{O}(p(n)) + \mathcal{O}(p^2(n)) + \mathcal{O}(p^2(n)) = \mathcal{O}(p^2(n))$$

Počet proměnných $x_{i,j,s}$ závisí je $\mathcal{O}(p^2(n))$, tedy závisí na $n = |w|$.
Proměnnou lze zakódovat do binární abecedy $\mathcal{O}(\log n)$ znaky.
Tedy $|\langle \Phi \rangle| = \mathcal{O}(p^2(n) \cdot \log n)$, což znamená $|\langle \Phi \rangle| = \mathcal{O}(p^2(n) \cdot n)$.

$\langle \Phi \rangle$ má polynomiální délku vzhledem k $n = |w|$ a lze spočítat v polynomiálním čase. Tedy $A \leq_p SAT$. ■

konjunktivní normální forma (cnf) výrokových formulí

literál = je proměnná nebo její negace

klauzule = disjunkce literálů

formule v **cnf** = konjunkce klauzulí

formule v **3cnf** = formule v cnf, kde každá klauzule má 3 literály

Definice (problém 3SAT)

Problém 3SAT je problém rozhodnout, zda je daná výroková formule v 3cnf splnitelná.

$$3SAT = \{\langle \varphi \rangle \mid \varphi \text{ je splnitelná formule v 3cnf}\}$$

Věta

3SAT je NP-úplný.

Důkaz: $3SAT \in NP$:

3SAT je NP-úplný

3SAT je NP-těžký, tj. $A \in \text{NP} \implies A \leq_p \text{3SAT}$:

1 zkonstruujeme Φ jako v důkazu NP-těžkosti SATu

2 Φ převedeme na ekvivalentní Φ' v cnf pomocí ekvivalence

$$(\varphi \wedge \psi) \vee \rho \equiv (\varphi \vee \rho) \wedge (\psi \vee \rho)$$

3 Φ' převedeme na Φ'' v 3cnf pomocí vztahů

$$\begin{aligned} & l_1 \\ & l_1 \vee l_2 \\ & l_1 \vee l_2 \vee l_3 \\ & l_1 \vee l_2 \vee l_3 \vee l_4 \\ & \vdots \\ & l_1 \vee l_2 \vee \dots \vee l_m \end{aligned}$$

Φ'' je splnitelná $\iff \Phi'$ je splnitelná $\iff \Phi$ je splnitelná
 $|\Phi''| \in \mathcal{O}(p^2(n))$. Tedy $A \leq_p 3SAT$. ■

Lemma

Nechť \mathbb{C} je složitostní třída splňující $NP \subseteq \mathbb{C}$.

A je \mathbb{C} -těžký a $A \leq_p B \implies B$ je \mathbb{C} -těžký

Důkaz:



Důsledek

Je-li A NP-úplný, $A \leq_p B$ a $B \in NP$, pak B je také NP-úplný.

Definice (k -klika)

Neorientovaný graf G má k -kliku, pokud v něm existuje úplný podgraf s k vrcholy.

Definice (problém CLIQUE)

Problém CLIQUE je problém rozhodnout, zda daný graf G má k -kliku pro dané k .

$$CLIQUE = \{ \langle G, k \rangle \mid G \text{ je graf s } k\text{-klikou} \}$$

Věta

CLIQUE je NP-úplný.

Důkaz: *CLIQUE* \in NP:

CLIQUE je NP-těžký: $3SAT \leq_p CLIQUE$

CLIQUE je NP-úplný

Nechť $\varphi = (a_1 \vee b_1 \vee c_1) \wedge (a_2 \vee b_2 \vee c_2) \wedge \dots \wedge (a_k \vee b_k \vee c_k)$.

Zkonstruujeme graf $G = (V, E)$, kde

- $V = \{a_1, b_1, c_1, a_2, b_2, c_2, \dots, a_k, b_k, c_k\}$ a
- $E = \{\{v_1, v_2\} \mid v_1, v_2 \text{ nejsou ze stejné klauzule a } v_1 \text{ není negací } v_2 \text{ nebo naopak}\}$.

$$\varphi = (x \vee y \vee z) \wedge (\neg x \vee \neg y \vee \neg z) \wedge (x \vee \neg y \vee \neg y)$$

φ je splnitelná \iff existuje splňující přiřazení \iff G má k -kliku
Zároveň $|G| = \mathcal{O}(|\varphi|^2)$ a tedy $3SAT \leq_p CLIQUE$.

další NP-úplné problémy

- problém hamiltonovské cesty *HAMPATH*
- problém obchodního cestujícího
(rozhodovací problém: existuje cesta do dané velikosti?)
- problém okachličkování (tiling problem)



Eternity II

- 256 dílků (16×16)
- 2.000.000 dolarů za řešení
- během 3 a půl roku nenalezeno

Lemma

Pokud je nějaký NP-úplný problém v P , pak $P = NP$.

Definice (vrcholové pokrytí)

*Nechť G je neorientovaný graf. Podmnožina jeho vrcholů se nazývá **vrcholové pokrytí** grafu G , pokud obsahuje alespoň jeden vrchol každé hrany grafu G .*

Definice (problém VERTEX-COVER)

***Problém VERTEX-COVER** je problém rozhodnout, zda daný graf G má vrcholové pokrytí obsahující právě daný počet vrcholů k .*

$$\text{VERTEX-COVER} = \{ \langle G, k \rangle \mid G \text{ je graf s vrcholovým pokrytím obsahujícím právě } k \text{ vrcholů} \}$$

VERTEX-COVER je NP-úplný

Věta

VERTEX-COVER je NP-úplný.

Důkaz: *VERTEX-COVER* \in NP:

VERTEX-COVER je NP-těžký: $3SAT \leq_p VERTEX-COVER$

VERTEX-COVER je NP-úplný

Nechť $\varphi = (a_1 \vee b_1 \vee c_1) \wedge (a_2 \vee b_2 \vee c_2) \wedge \dots \wedge (a_l \vee b_l \vee c_l)$ používá množinu proměnných X . Zkonstruujeme graf $G = (V, E)$, kde

- $V = \{a_1, b_1, c_1, a_2, b_2, c_2, \dots, a_l, b_l, c_l\} \cup \{x, \neg x \mid x \in X\}$ a
- $E = \{\{a_i, b_i\}, \{a_i, c_i\}, \{b_i, c_i\} \mid 1 \leq i \leq l\} \cup \{\{x, \neg x\} \mid x \in X\} \cup$
 \cup “hrany mezi literálem a shodným vrcholem x nebo $\neg x$ ”

$$\varphi = (x \vee y \vee z) \wedge (\neg x \vee \neg y \vee \neg z) \wedge (x \vee \neg y \vee \neg y)$$

φ je splnitelná \iff G má vrcholové pokrytí s $k = 2l + |X|$ vrcholy
Zároveň $|G| = \mathcal{O}(|\varphi|)$ a tedy $3SAT \leq_p VERTEX-COVER$. ■