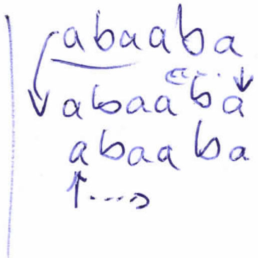
Složitost

$w = w^R$

- Ukažte, že $L = \{ w \in \Sigma^* \mid w \text{ je palindrom} \} \in DTime[n]$

Důkaz (idea):

- Zkonstruujeme 2-páskový DTS, který přijímá jazyk L v lineárním čase. Označme tento stroj M.
- M projde vstupní pásku (řetězec na vst. pásce) zleva doprava a zkopíruje ho na 2. pásku.
- M posune hlavu na 2. pásce zpět na začátek této pásky.
- M synchronizovaně posouvá obě hlavy protisměrně (na 1. pásce doleva, na 2. pásce doprava), dokud na obou páskách čte stejné symboly.



- Pokud M projde úspěšně celý řetězec, přijme, jinak odmítne.
- M tedy přijímá právě tehdy, když $w = w^R$, a tedy

$w \in L$.

- Složitost výpočtu je

$$O(n) + O(n) + O(n) = O(n).$$

složitost kontroly shody obsahu 1. p (reverzované) a 2. p.

↑ kopie 1. pásky na 2. pásku

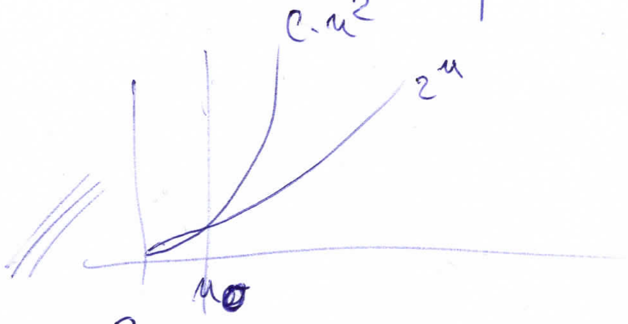
↑ posun zpět na začátek 2. pásky.

$\square$

- Ukažte, že $2^n \notin O(n^2)$.

Důkaz sporem:

důsledek předp.:



- Předpokládejme, že $2^n \in O(n^2)$.
- Pak dle def. platí, že
  $\exists c \in \mathbb{R}^+ \ \exists n_0 \in \mathbb{N} \ \forall n \geq n_0 : 2^n \leq c \cdot n^2$
- Pak ale platí, že $\exists c \in \mathbb{R}^+ \ \exists n_0 \in \mathbb{N} \ \forall n \geq n_0 : \frac{c \cdot n^2}{2^n} \geq 1$,
  neboť $2^n \neq 0$ pro žádné $n \geq 0$ a můžeme tedy krátit.
- Na druhou stranu ale platí:

$$\lim_{n \to \infty} \frac{c \cdot n^2}{2^n} = \left| \begin{array}{l} \text{pro } n \to \infty : \ cn^2 \to \infty, \ 2^n \to \infty \\ \text{lze užít L'Hospitalova pravidla} \end{array} \right| = \lim_{n \to \infty} \frac{\frac{dcn^2}{dn}}{\frac{d2^n}{dn}} =$$

$$= \lim_{n\to\infty} \frac{2\cdot cn}{(\ln 2)\cdot 2^n} = \left| \begin{array}{l} \text{pro } n\to\infty: 2\cdot cn \to \infty \\ \qquad (\ln 2)\cdot 2^n \to \infty \\ \text{a tedy opět uplatníme} \\ \text{L'Hospitalovo pr.} \end{array} \right| =$$

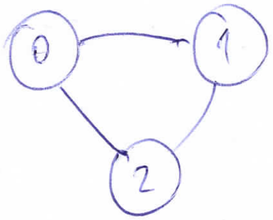
$$= \lim_{n\to\infty} \frac{2\cdot c}{(\ln 2)^2\cdot 2^n} = 0$$

- Nemůže současně platit $\forall n \geq n_0: \frac{c\cdot n^2}{2^n} \geq 1$ a současně $\lim_{n\to\infty} \frac{c\cdot n^2}{2^n} = 0$. Spor. $\square$

- <u>Problém kliky v neor. grafu G</u>
  - Klika v neor. grafu: podgraf, který je úplným grafem.
  - Problém kliky: Má daný graf kliku velikosti k?
  - KLIKA = {<(G, k)> | G je neor. graf s klikou vel. k}.
    - <_> reprezentuje kódování dvojic grafů a velikosti.
    - G lze kódovat tak, že nejprve uvedeme sekvenci čísel uzlů grafu a následně sekvenci dvojic čísel uzlů odpovídajících hranám grafu, vše vhodně odděleno. Čísla uzlů lze zapsat v bin. soustavě.

• Velikost Kliky k lze také zapsat binárně a vhodně oddělit

Např. graf



lze zakódovat jako 00|01|10|(00,01)|(01,10)|(10,00)

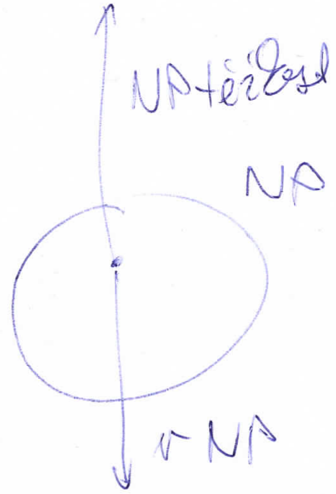
– Problém kliky je NP-úplný. DOKAŽTE.

I. Ukážeme, že KLIKA je v NP.

Ukážeme, že existuje nedeterministický vícepáskový TS, který přijímá jazyk KLIKA v polynom. čase. Označme ho M. M může pracovat takto:

1. M ověří zda má na vstupu správně zformovanou instanci problému kliky (jinak odmítne):

   a) Kontrola správného použití oddělovačů – lze realizovat jediným průchodem a tedy se složitostí O(n).



b) M ověří, že na vstupu nejsou duplikované uzly: Pro každý uzel (těch je $O(n)$) M postupuje tak, že daný uzel přepíše na pomocnou pásku (složitost $O(n)$), následně ho porovná s ostatními uzly (těch je $O(n)$), přičemž porovnání 2 uzlů má složitost $O(n)$. Celkově tedy tento krok má složitost $O(n) \cdot (O(n) + O(n) \cdot O(n)) = O(n^3)$. Tato složitost pokrývá i případy oddělovačů.

c) M následně ověří existenci mn-y hran (tedy neexistence duplikovaných hran, a to včetně možné reverze, a neexistenci hran nad neznámými uzly). Lze řešit analogicky jako výše v polynom. čase.

2. Označí náhodně k uzlů: Přepíše k na pomocnou pásku ($O(n)$), následně prochází jednotlivé uzly (těch je $O(n)$) a náhodně vybrané označuje, přičemž dekrementuje k ($O(n)$). Celkově tedy má tento krok složitost $O(n^2)$.

3. M ověří, zda vybrané uzly tvoří kliku. Projde všechny dvojice označených čísel (těch je $O(n^2)$). Každou dvojici zapíše na pomocnou pásku a srovná s hranami na vstupu. Těch je $O(n)$, přičemž jedno srovnání má složitost $O(n)$. Celkově má tento krok složitost $O(n^4)$.

4. M přijme, pokud zvolené uzly tvoří kliku, jinak odmítne.

KLIKA je NP-těžká

- Ukážeme pomocí polynomiální redukce z problému 3SAT, což je NP úplný problém.
- 3SAT:
  - m Booleovských proměnných $x_1, \dots, x_m$, $m \geq 1$.
  - literály L: $x_i \,/\, \neg x_i$ pro $i \in \{1, \dots, m\}$.
  - klauzule C: $L_{i_1} \vee L_{i_2} \vee L_{i_3}$ pro literály $L_{i_1}, L_{i_2}, L_{i_3}$.
  - formule F: $C_1 \wedge \dots \wedge C_k$, $k \geq 1$, pro klauzule C_i, $1 \leq i \leq k$.
  - ptáme se, zda daná formule je splnitelná.

- Idea redukce na příkladu:

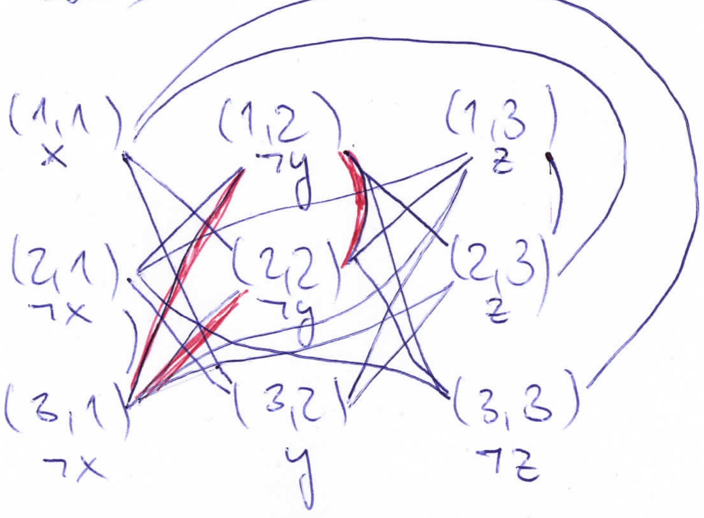
$x_1 = x$, $x_2 = y$, $x_3 = z$

a $F = (x \vee \neg y \vee z) \wedge$
$(\neg x \vee \neg y \vee z) \wedge$
$(\neg x \vee y \vee \neg z)$

$\rightsquigarrow$

$1 \le j \le 3$

$1 \le i \le k$



$\neg y \rightsquigarrow y = 0$

$\neg x \to x = 0$

z lze zvolit libovolně ($z = 0/1$)

- Formálněji:

Libovolnou formuli F představující instanci 3SAT (tedy $F = C_1 \wedge \dots \wedge C_k$, kde C_i ($1 \le i \le k$) jsou klauzule zavedené výše) lze převést na graf $G_F = (V, E)$:

$V = \{[\underbrace{i}_{\#\text{klauzule}}, \underbrace{j}_{\#\text{literálu}}] \mid 1 \le i \le k \wedge 1 \le j \le 3\}$

$E = \{\{[i,j],[i',j']\} \mid i \ne i' \wedge L_{ij} \ne \neg L_{i'j'}\}$

za předp., že $\neg\neg x = x$.

- Uvedenou redukci lze implementovat DTS M, který je úplný a pracuje v polynomiálním čase takto:

  - M bude generovat graf o $3k$ uzlech a méně než $(3k)^2$ hranách.
  - Pro kódování instance kliky lze užít kódování uvedené již dříve, jen si pro zjednodušení konstrukce hran doplníme ke každému číslu uzlu odpovídající literál
  - Formuli lze kódovat tak, že proměnné očíslujeme a čísla zapíšeme v binární soustavě.
  - M nejprve provede kontrolu, zda má na vstupu korektně zformulovanou formuli. Pokud ne, vygeneruje graf o jednom uzlu a přidá k němu velikost kliky 2. Toto lze evidentně realizovat v $O(n)$
  - M projde formuli a pro každý literál (těch je $O(n)$) vygeneruje na pomocnou pásku nové číslo uzlu (na další pozice si inkrementuje čítač uzlů $- O(n)$). Ke každému číslu uzlu

nalezení najde příslušný literál mezi vhodnými oddělovači (O(n)). Celkově tedy tento krok lze realizovat v $O(n^2)$.

- Pro vygenerování seznamu hran M projde seznam uzlů (těch je O(n)) a porovná jim odpovídající literál (cena O(n)) s literály ostatních uzlů z jiných klausulí (těch je O(n)). Pokud si literály neodporují, zapíše příslušnou hranu. Pro rozeznání uzlů ze stejné klausule musí doplnit vhodný oddělovač uzlů v klausulích. Tento krok má evidentně složitost $O(n^3)$.
- Celkově je tedy redukce uprovedena v polynomiálním čase.

- Zbývá ukázat, že dobře zformovaná formule $F = C_1 \wedge \ldots \wedge C_k$ se redukuje na G_F, který má kliku velikosti k, tehdy a jen tehdy, když F je splnitelná. Případ nesprávně zformovaných formulí je již výše vyřešen.

a) "$\Rightarrow$" - Je-li v G_F klika velikosti k, znamená tom, že v této klice je jeden uzel z každé trojice uzlů odpovídající jedné klauzuli — koneckonců mezi uzly téže klauzule nejsou hrany.

- Uzly kliky jsou všechny propojeny hranami a hranami propojujeme pouze ty uzly, jejichž literály nekolidují, tj. je možné je současně splnit ($\neg x \leadsto x=0$, $x \leadsto x=1$).
- Tím splníme každou klauzuli a tedy i celou formuli.

b) „$\Leftarrow$“ – Je-li F splnitelná, musí být možno splnit v každé klauzuli alespoň jeden literál. Tyto literály si tedy neodporují a jsou z jiných klauzulí, a proto jim odpovídající uzly budou propojeny hranami.

- V G tedy určitě bude klika velikosti k propojující výše uvedené uzly.

$\square$

- Ukažte, že třída $\mathbb{P}$ je uzavřena vůči $\cup, \cap$.

a) Ukážeme, že $\forall L_1, L_2 \in \mathbb{P}: L_1 \cup L_2 \in \mathbb{P}$

Důkaz (idea):

- Uvažme libovolné $L_1, L_2 \in \mathbb{P}$.
- K L_1, L_2 existují DTS přijímající L_1, L_2 a dokonce rozhodující L_1, L_2 v polynomiálním čase.
- Označme tyto stroje M_1 a M_2 a předpokládejme, že pracují v čase $P_1(n)$ a $P_2(n)$.
- Zkonstruujeme DTS M rozhodující $L_1 \cup L_2$:
  1. M zkopíruje obsah vstupní pásky na 2. pásku.
  2. M odsimuluje na 1. pásce M_1. Pokud přijme.
  3. Jinak M odsimuluje na 2. pásce M_2. Pokud přijme, přijme, jinak odmítne.
- M zřejmě rozhoduje $L_1 \cup L_2$, a to v čase

kopie $\rightarrow O(n) + P_1(n) + P_2(n) + C$ ← konstanta - režie předávání řízení

Tedy M pracuje v polynomiálním čase. $\square$

b) Uzavřenost vůči $\cap$ je analogická. Stroj M se liší pouze tím, že pokud M_1 přijme, pak M předá řízení M_2, jinak odmítne. $\square$

c) Ukážeme, že $\forall L_1, L_2 \in \mathbb{P} : L_1 \cdot L_2 \in \mathbb{P}$.

Důkaz (idea)

- Podobně jako je zmíněno již výše, pro libovolné L_1 a $L_2 \in \mathbb{P}$ můžeme předpokládat existenci DTS M_1, M_2, které tyto jazyky rozhodují v polynomiálních časech $P_1(n)$ a $P_2(n)$.
- Sestrojíme DTS M, který rozhoduje $L_1 \cdot L_2$ v polynomiálním čase:
  - M pro libovolný vstup $w = a_1 a_2 \dots a_n$ zkouší $n+1$ možných rozdělení $w = w_1 \cdot w_2$:
    - $w_1 = \varepsilon$ / $w_2 = a_1 \dots a_n$
    - $w_1 = a_1$ / $w_2 = a_2 \dots a_n$
    - ...

– $w_1 = a_1 \dots a_n$ / $w_2 = \varepsilon$

– Pro každé rozdělení M zkopíruje w_1 na 2. pásku a w_2 na 3. pásku.

– Na 2. pásce odsimuluje M_1 a 3. pásce M_2.

– Pokud M_1 a M_2 přijmou pro nějaké rozdělení, M přijme, jinak odmítne.

– M evidentně rozhoduje $L_1 \cdot L_2$, a to v čase

$$(n+1) \cdot \big(O(n) + P_1(n) + P_2(n) \big) \in O(n^k)$$

↑ mazání, kopírování apod.

pro nějaké k. $\square$

– <u>Uzavřenost P vůči *.</u>

POZOR! Nelze řešit pomocí systematického členění vstupní řetězce na 1, 2, 3, ... podřetězce a zkoumání všech takových rozdělení! Vede na exponenciální složitost!