Složitost

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

Důkaz (idea):

- Zkonstruujeme 2-páskový DTS, který přijímá jazyk L — označme tento DTS jako M — v lineárním čase.
- M pracuje následovně:
  1. M přesune hlavu na 1. pásce na konec vstupu.
  2. M posouvá hlavu na 1. pásce zprava doleva a současně kopíruje (zleva doprava) čtený obsah na 2. pásku.
  3. M přesune hlavu na 2. pásce zpět na začátek.
  4. M synchronizovaně prochází obsah obou pásek a kontroluje, zda je stejný (a tedy, zda $w = w^R$). Pokud se obsah obou pásek shoduje, M přijme, jinak odmítne.

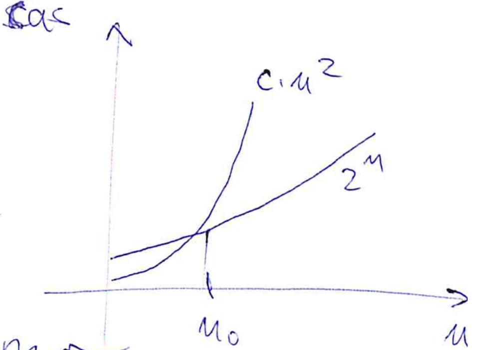
- Analyzujeme složitost M:
  - krok 1: sl. $n+1$
  - krok 2: sl. $O(n)$, přesněji v řádu $3n$.
  - krok 3: sl. $n+1$
  - krok 4: sl. $O(n)$, přesněji v řádu $2n$.

---

celkově tedy $(n+1) + O(n) + (n+1) + O(n) = O(n)$

- Dokažte, že $2^n \notin O(n^2)$.

<u>Důkaz sporem:</u>



- Předpokládejme, že $2^n \in O(n^2)$.
- Pak $\exists c \in \mathbb{R}^+ \exists n_0 \in \mathbb{N} : \forall n \geq n_0 : 2^n \leq c \cdot n^2$
- Můžeme krátit vzhledem k tomu, že $2^n \neq 0$ pro žádné n:

$$\exists c \in \mathbb{R}^+ \exists n_0 \in \mathbb{N} \; \forall n \geq n_0 : \frac{c \cdot n^2}{2^n} \geq 1$$

- Uvažme $\lim_{n \to \infty} \frac{c \cdot n^2}{2^n} = \left| \begin{array}{ll} \text{pro } n \to \infty: & \text{lze užít} \\ c \cdot n^2 \to \infty & \\ 2^n \to \infty, & \text{L'Hospitalovo} \\ & \text{pravidlo} \end{array} \right| =$

$$= \lim_{n\to\infty} \frac{\frac{d\, cn^2}{dn}}{\frac{d\, 2^n}{dn}} = \lim_{n\to\infty} \frac{2cn}{(\ln 2)\cdot 2^n} =$$

$$= \left| \begin{array}{ll} \text{pro } n\to\infty: & \text{uplatníme znova} \\ 2cn \to \infty, & \text{L'Hospitalovo} \\ (\ln 2) 2^n \to \infty, & \text{pravidlo} \end{array} \right| =$$

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

- Nemůže současně platit, že $\lim_{n\to\infty} \frac{c\cdot n^2}{2^n} = 0$

a $\forall n \geq n_0 : \frac{c\cdot n^2}{2^n} \geq 1$. Spor. □

Problém kliky velikosti k v neor. grafu G

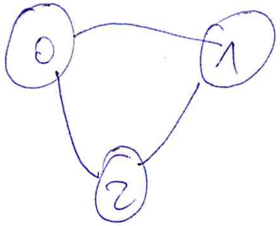
- klika v neor. grafu: podgraf, který je úplný.
- Problém kliky: Má daný graf G kliku velikosti k?
- Lze charakterizovat jazykem

KLIKA = { <(G, k)> | G je graf, který má kliku vel. k },

kde <.> je kódování dvojic graf, velikost

např. takto:

- graf můžeme zakódovat jako posloupnost čísel uzlů zapsaných v binární soustavě, následovaný seznamem hran zapsaných jako dvojice čísel uzlů, vše vhodně odděleno, např.



00 | 01 | 10 | (00,01) | (01,10) | (10,00)

- velikost kliky zakódujeme binárním číslem a oddělíme od kódu grafu např. "#".

- KLIKA patří mezi NP-úplné jazyky. Ukažme.

I. Ukážeme, že KLIKA je v NP (idea důkazu)

Při rozhodování, zda daný řetězec $w \in$ KLIKA může NTS M postupovat následovně:

1. M ověří, zda w je platnou instancí problému kliky,

potud ne, odmítne. Ověření lze provést následovně:

a) M projde zleva doprava vstup a ověří správnost použití oddělovačů, což odpovídá testu na členství v reg. jazyce. Složitost tohoto kroku je O(n).

b) M ověří, že v seznamu ~~[illegible]~~ uzlů se žádný ~~[illegible]~~ uzel neopakuje. Každý uzel, těch je O(n), přepíše na pracovní pásku a ověří, zda se nevyskytují mezi ostatními uzly. Těch je také O(n), přičemž porovnání dvou čísel uzlů lze realizovat v O(n). Celkově tedy tento krok lze řešit v O(n3).

c) M ověří, že žádná hrana není duplikovaná a že žádná hrana není založena na neexistujícím uzlu. Za tím účelem si můžeme opět hrany a uzly přepsat na pracovní pásky a podobně jako výše dojdeme k tomu, že tento test je v $O(n^k)$ pro nějaké $k \in N$

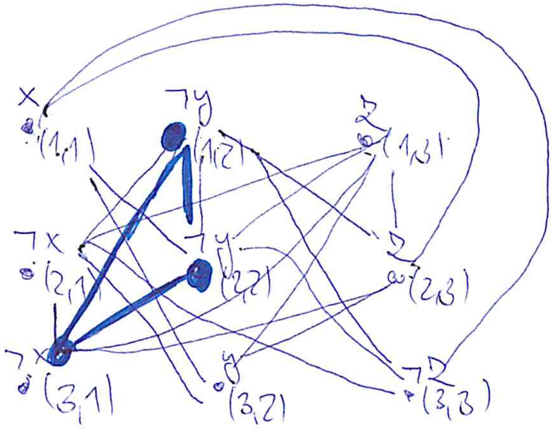
Test na správné zformování vstupní instance je tedy v $O(n^k)$.

2. M náhodně označí k uzlů grafu (např. jejich přepisem na pomocnou pásku) jako součást kliky.
Přepis 1 uzlu na pomocnou pásku je v O(n), přepíše O(n) uzlů, při přepisu bude dekrementovat čítač, který uvádí velikost kliky (původně má hodnotu k), který může mít na pomocné pásce.
Pokud je k větší než počet uzlů grafu, M odmítne.
Celkově má tento krok opět složitost $O(n^l)$ pro nějaké $l \in N$.

3. M ověří, zda k náhodně zvolených uzlů tvoří kliku.
Pokud ano, přijme, jinak odmítne. Tento krok může M realizovat tak, že pro každou dvojici zvolených uzlů (takových dvojic je $O(n^2)$) tuto dvojici přepíše na pomocnou pásku a projde seznam hran (těch je O(n)) a ověří, zda daná dvojice stavů odpovídá dané hraně (test v $O(n^1)$). Celkově tedy tento krok je v $O(n^4)$, neboli $O(n^l)$ pro $l \in N$.

II. Ukážeme, že KLIKA je NP-těžká, a to redukcí ze 3-SAT

- 3-SAT:
  - máme m ($m \geq 1$) booleovských proměnných $x_1, \dots, x_m$.
  - literály L: $x_i, \neg x_i$ pro $1 \leq i \leq m$.
  - 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: $C_1 \wedge \dots \wedge C_k$, $k \geq 1$.
  - ptáme se, zda daná formule F je splnitelná.
- 3-SAT je NP-úplný.
- K ukázání, že KLIKA je NP-těžká je nutno užít polynomiální redukci: redukující TS musí být DTS a pracovat polynomiálním čase.
- Princip redukce na příkladu:
  - $x = x_1$, $y = x_2$, $z = x_3$
  - $F = (x \vee \neg y \vee z) \wedge (\neg x \vee \neg y \vee z) \wedge (\neg x \vee y \vee \neg z)$ $\xrightarrow{3}$ $\downarrow k$
  - F lze splnit např. $x = y = z = false$.



- Idea důkazu
  - Libovolnou instanci 3-SAT, tedy formuli $F = C_1 \wedge \ldots \wedge C_k$, $k \geq 1$, lze převést na graf $G_F = (V, E)$, kde:
    - $V = \{ [i,j] \mid 1 \leq i \leq k \wedge 1 \leq j \leq 3 \}$,

      ↑ index klauzule ↖ index literálu
    - $E = \{ \{[i,j], [i',j']\} \mid i \neq i' \wedge l_{ij} \neq \neg l_{i'j'} \}$.

      ↑ nepropojujeme v rámci jedné klauzule

      nepropojujeme kolidující literály
  - Uvedenou redukci lze implementovat úplným DTS M pracujícím v polynomiálním čase pro vhodné kódování grafů, velikosti kliky a formule:
    - graf a velikost kliky lze kódovat jako v části I důkazu. Pro zjednodušení generování lze kód uzlu spojit s kódem odpovídajícího literálu
    - Pro kódování formulí postačí vhodně očíslovat (v bin. kódu) jednotlivé proměnné.

- M ověří průchodem ~~formuli, že se~~ vstupním řetězcem, že se jedná o platnou formuli (složitost O(n)). Pokud ne, vygeneruje ~~[illegible]~~ <({0,1},∅),2>.

- M vygeneruje seznam uzlů grafu GF, a to tak, že projde formuli a pro každý literál vygeneruje nový uzel, jehož číslo (resp. dvojici čísel klauzule, číslo literálu) si počítá na pomocných páskách v binárním kódu. Jednotlivé operace spojené s generováním kódu jednoho uzlu (přepis literálu, vygenerování čísla sloupce/řádku a jejich přepis) jsou v O(n). Uzlů je O(n). Celkově tedy tento krok realizuje v O(n²).

- M vygeneruje seznam hran. K tomu postačí, aby prošel všechny dvojice vygenerovaných uzlů, posoudil, zda odpovídající literály kolidují, a pokud ne, vytvoří hranu.

Dvojic je $O(n^2)$, jednotlivé operace spojené s generováním jedné hrany jsou v $O(n)$ a tedy celkově tento krok je v $O(n^3)$

- M za kód grafu vygeneruje hodnotu k v bin. zápisu, kde k je počet klauzulí (a to opět v $O(n^l)$, $l \in \mathbb{N}$).

- Zbývá ukázat, že $F = C_1 \wedge \dots \wedge C_k$ je splnitelná $\iff$ G_F má kliku velikosti k [redacted] ;

a) "$\Rightarrow$" – Je-li F splnitelná, lze splnit každou její klauzuli a tedy v každé klauzuli lze najít alespoň 1 literál tak, že si tyto literály navzájem neodporují.

- Uvažme uzly G_F odpovídající uvedeným literálům. Tyto uzly jsou z různých klauzulí a odpovídají neodporujícím literálům a jsou tedy všechny propojeny hranami.
- G_F tedy má kliku velikosti k (máme k klauzulí a z každé 1 literál).

b) "$\Leftarrow$" – Je-li v grafu G_F klika velikosti k, je tvořena uzly z různých řádků a tedy odpovídajících různým klauzulím.

- uvedené uzly jsou označeny nekolidujícími literály.
- Lze tedy literály spojené se všemi uzly v klice současně splnit.
- Tím dostaneme splňující ohodnocení pro danou formuli, protože splníme všechny klauzule.

□