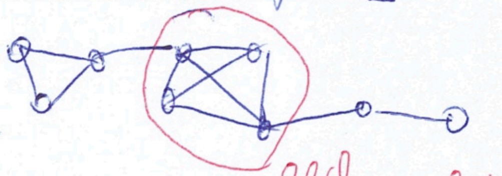
TIN – demo – 10/12/2021 – složitost

1. Problém kliky dané velikosti v neorientovaném grafu

– klika : úplný podgraf (např. )

klika vel. 4.

– problém kliky lze charakterizovat jazykem

KLIKA = { ⟨(G, k)⟩ | G je neor. graf, který má kliku vel. k }

Zde ⟨.⟩ jsou vhodná kódování (G, k), např.:

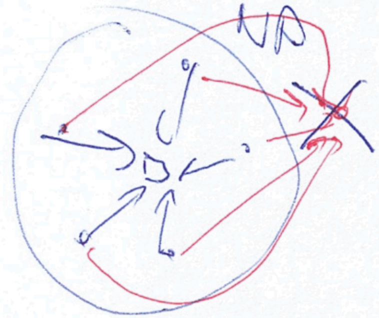
– G zakódujeme jako sekvenci čísel uzlů (bez opakování) následovanou sekvencí kódů hran (dvojice čísel uzlů, a to těch, které v grafu jsou), bez opakování!

– čísla zapíšeme binárně, vše oddělíme vhodnými oddělovači

– Např. ⟨  ⟩ = 00|01|10|(00,01)|(01,10)|(10,00)

– velikost kliky zakódujeme binárně.

– Dokažte, že KLIKA je NP-úplná.



Základní idea:

I. Klika $\in$ NP — zkonstruujeme NTS, který kliku přijímá v polynom. čase:

1. kontrola kódování
2. "guess" náhodnou kandidátní množinu kliky
3. "check": ověříme, zda zvolená množina tvoří kliku

II. Klika je NP-těžká

└ polynomiální redukce z nějakého NP-úplného problému na Kliku

Podrobněji:

ad I. Ukážeme, že KLIKA $\in$ NP

– Zkonstruujeme NTS M, kde $L(M)$ = KLIKA, a M pracuje v polynom. čase. M bude pracovat následovně:

1. M ověří, že na vstupu má korektní instanci problému kliky:

a) M projde pásku zleva doprava a ověří správnost použití oddělovačů. Cena: $O(n)$.

b) M projde všechny kódy uzlů (těch je $O(n)$) a postupně je přepíše na pomocnou pásku (cena: $O(n)$). Každý kód srovná s ostatními (těch je $O(n)$), přičemž cena srovnání je $O(n)$ a kontroluje výskyt duplicit. Celkové cena:

$$\underset{\substack{\uparrow \\ \text{vých. uzly}}}{O(n)} \cdot \big(\underset{\substack{\uparrow \\ \text{přepis}}}{O(n)} + \underset{\substack{\uparrow \\ \text{srovnávané uzly}}}{O(n)} \cdot O(n) \big) = O(n^3)$$

cena 1 srovnání

c) Analogicky v polynom. čase ověří kódování hran.

2. "Guess": M přepíše ze vstupní pásky k náhodně zvolených uzlů na pásku pomocnou. Za tím účelem nejprve přepíše na další pásku samotné k (cena $O(n)$). Pak postupně dekrementuje k (cena $O(n)$) a s každým dekrementem přepíše 1 uzel na pomocnou pásku (cena: $O(n)$). Pokud nastane nedostatek

uzlů – odmítne. Celkově: vybíráme $O(n)$ uzlů – na vstupu jich určitě není více. Každý element je v $O(n)$ a přepis také v $O(n)$. Celkově $O(n^2)$.

3. "check": M projde všechny dvojice zvolených uzlů (těch je $O(n^2)$) a pro každou dvojici projde všechny hrany (těch je $O(n)$) a ověří, že každá dvojice je propojena hranou (vlastní kontrola: $O(n)$). Celkově: $O(n^4)$.

ad II. KLIKA je NP-těžká

- Důkaz polynomiální redukcí ze 3-SAT.
- 3-SAT značí splnitelnost formulí

$F = C_1 \wedge C_2 \wedge \dots \wedge C_k$, $k \geq 1$, kde

$\forall 1 \leq i \leq k: \quad C_i = L_1^i \vee L_2^i \vee L_3^i$, kde

$\forall 1 \leq j \leq 3: \quad L_j^i = x_e$ nebo $L_j^i = \neg x_e$, kde

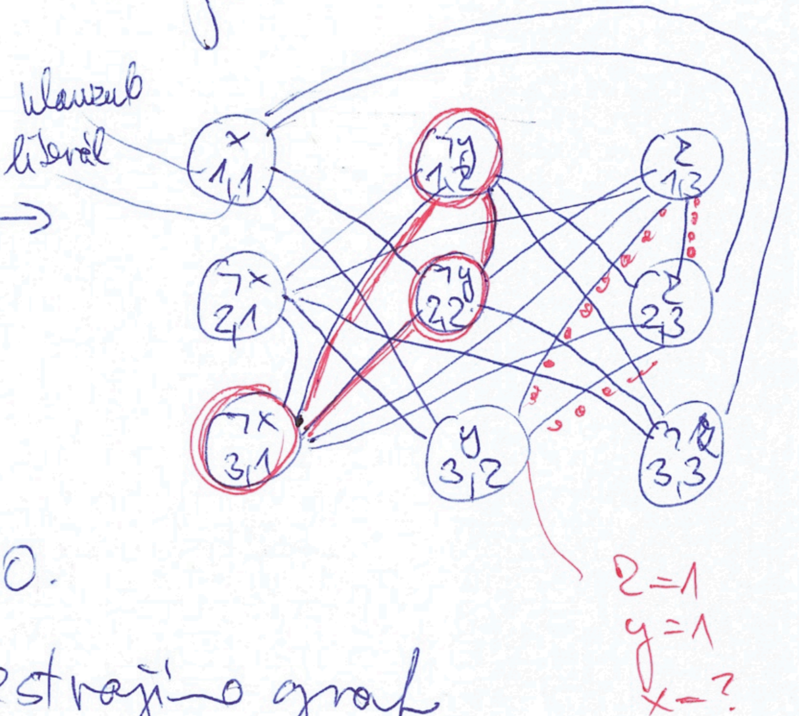
$x_e \in \{x_1, \dots, x_m\} = X$, $m \geq 1$, kde

X je mn-a booleovských proměnných.

- Idea redukce na příkladu:

$$F = \underbrace{(x \vee \neg y \vee z)}_{C_1} \wedge \underbrace{(\neg x \vee \neg y \vee z)}_{C_2} \wedge \underbrace{(\neg x \vee y \vee \neg z)}_{C_3}, \text{ kde}$$

$\longmapsto$



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

Lze splnit např. pro $x = y = z = 0$.

- Formálně k formuli F sestrojíme graf $G_F = (V_F, E_F)$, kde:
  - $V_F = \{ [i, j] \mid 1 \le i \le k \wedge 1 \le j \le 3 \}$

    i … # klauzule, j … # literálu

    (vlastní literály budeme číst z formule)
  - $E_F = \{ \{[i,j], [i',j']\} \mid i \ne i' \wedge L^i_j \ne \neg L^{i'}_{j'} \}$

    s využitím idempotence $\neg\neg x = x$
- ke grafu přidáme k (počet klauzulí).

– Uvedenou redukci lze implementovat úplným DTSM pracujícím v polynomiálním čase takto (opomineme tentokrát kontrolu správné struktury vstupní formule F):

1. M vygeneruje seznam uzlů tak, že si na pomocných páskách přečte pozice i a j a že každé dvojici i a j v rámci postupného průchodu F přiřadí příslušný literál L_{ij} ($1 \le i \le k$, $1 \le j \le 3$).

   Složitost: Máme $O(n)$ literálů. Jejich kopie $O(n)$. Inkrement a kopie indexů mají cenu $O(n)$. Celkově $O(n) \cdot (O(n) + O(n) + O(n)) = O(n^2)$

2. M vytvoří seznam hran tak, že projde přes všechny dvojice každých uzlů (těch je $O(n^2)$) a jsou-li kompatibilní (kontrola: $O(n)$)

"$\Leftarrow$" - Má-li G_F kliku vel. k, tato musí být tvořena uzly z různých skupin uzlů odpovídajících různým klauzulím. Navíc příslušné uzly musí být označeny nekolidujícími literály.

- všechny literály spojené s uzly v klice lze tedy splnit současně a tím najít splňující ohodnocení F, neboť splníme všechny klauzule současně.

$\square$

přepíše hranu na pozorovou pásku ($O(n)$).
celkově: $O(n^2) \cdot (O(n) + O(n)) = O(n^3)$.

3. M projde jistě jednou formuli a spočte v bin. řádu počet klauzulí k, který zapíše za graf: $O(n)$ – průchod, $O(n)$ zápis, celkově $O(n^2)$.

– Zbývá ukázat, že F je splnitelná $\iff$ G_F má kliku vel. k:

"$\Rightarrow$" – Je-li F splnitelná, pak lze v každé klauzuli C_i $(1 \leq i \leq k)$ zvolit literál L^i_j $(j \in \{1,2,3\})$ tak, že žádné dva zvolené literály nekolidují (tedy $L^i_j \neq \neg L^{i'}_{j'}$).

– Ovšem zvolené literály odpovídají uzlům grafu ze skupin odpovídajících různým klauzulím a jsou všechny propojeny hranami (protože literály nekolidují).

– Tyto uzly tedy tvoří kliku vel. k.

2. Uveďte alespoň základní ideu redukce z problému kliky na SAT.

- Základní idea:
  - Uvažujme graf $G = (V, E)$ a velikost kliky k.
  - Předpokládejme, že uzly jsou očíslované – $V = \{v_1, v_2, \dots, v_m\}$, $m \geq 1$.
  - Očíslujme také uzly kliky, a to $1, \dots, k$.
  - Zavedeme booleovské proměnné x^i_j, kde $1 \leq i \leq m$, $1 \leq j \leq k$, a to s následující sémantikou:

$$x^i_j = \begin{cases} \text{true} & \text{jestliže uzel } v_i \text{ je } j\text{-tým uzlem kliky} \\ \text{false} & \text{jinak} \end{cases}$$

  - Zformulujeme podmínky, které proměnné mohou být tedy splněny:

1. Každý uzel kliky musí odpovídat nějakému uzlu grafu. To vyjádříme formulí

$$\varphi_1 \equiv \underbrace{\bigwedge_{1 \le j \le k}}_{\text{„pro každý uzel kliky"}} \; \underbrace{\bigvee_{1 \le i \le m}}_{\text{„existuje uzel grafu"}} x^i_j$$

2. Žádné dva uzly kliky se ale nesmí namapovat na stejný uzel grafu.
(= pro každé dva různé uzly kliky odpovídající uzly grafu budou různé) — vyjádříme formulí

$$\varphi_2 : \bigwedge_{1 \le j_1 < j_2 \le k} \; \bigwedge_{1 \le i \le m} \left(\neg x^i_{j_1} \vee \neg x^i_{j_2} \right) \equiv \neg \left(x^i_{j_1} \wedge x^i_{j_2} \right)$$

3. Nesmí se dále stát, že dva různé uzly grafu, mezi kterými není hrana, budou oba vybrány do kliky — vyjádříme násl. form.:

$$\varphi_3 : \bigwedge_{\substack{1 \le i_1 < i_2 \le m \\ \wedge \{v_{i_1}, v_{i_2}\} \notin E}} \; \bigwedge_{1 \le j_1 < j_2 \le k} \left(\neg x^{i_1}_{j_1} \vee \neg x^{i_2}_{j_2} \right)$$

$$\equiv \neg \left(x^{i_1}_{j_1} \wedge x^{i_2}_{j_2} \right)$$

- Tedy dvojici (G, k) představující instanci problému Klika zredukujeme na formuli $\varphi_1 \wedge \varphi_2 \wedge \varphi_3$.
- Redukci lze implementovat DTS pracující v polynom. čase.
- Snadno él také nahlédneme, že G má kliku velikosti $k \iff \varphi_1 \wedge \varphi_2 \wedge \varphi_3$ je splnitelná. $\square$

Lineární programování (LP)

- optimalizační problém
- máme vektor reálných proměnných $\overline{x} = (x_1, \dots, x_m)$, $m \ge 1$.
- Máme soustavu lineárních nerovnic $\overline{A}\,\overline{x} \le \overline{b}$ ($a_{11} x_1 + a_{12} x_2 + \cdots \le b_1$ $a_{21} x_1 + a_{22} x_2 + \cdots$)

– Hledáme řešení takové, že $\overline{x} \geq 0$, které navíc maximalizuje / minimalizuje hodnotu $\overline{c}^T \cdot \overline{x}$ pro nějaký vektor tak $\overline{c} = (c_1, \ldots, c_m)$.

Integer LP (ILP)

- hodnoty $x_i \in \mathbb{Z}$ pro $1 \leq i \leq m$.
- rozhodovací verze ILP se ptá pouze, zda existuje nějaké celočíselné řešení.
- rozhodovací verze ILP je NP-úplná!

3. Ukažte základní ideu důkazu NP-úplnosti rozh. verze ILP

I. Rozhodovací verze ILP je v NP

Idea: "guess & check" (+ kontrola správného zformování – v dalším přednášce)

- NTS uhádne kandidáta $\bar{x}$ na řešení a ověří, že $\bar{x}$ je opravdu řešení.
- POZOR: Je ale třeba ukázat, že stačí hádat polynomiálně velká řešení, tedy řešení omezená na nějaké $O(n^l)$, kde l je nějaká konstanta (viz literatura).

II. Rozhodovací verze ILP je NP-těžká

- Lze ukázat např. redukcí (polynomiální) ze 3-SAT.
- 3-SAT – splnitelnost formule

$$F = C_1 \wedge \dots \wedge C_k, \quad k \geq 1,$$

$\forall 1 \leq i \leq k$: $C_i = L_1^i \vee L_2^i \vee L_3^i$, kde

$\forall 1 \leq j \leq 3$: $L_j^i = x_e$ nebo $L_j^i = \neg x_e$,

kde $x_e \in \{x_1, \dots, x_m\}$, kde

x_e jsou booleovské prom.

- Lze redukovat na instanci rozhodovací verze ILP takto.

  - Každému x_ℓ přiřadíme celočíselnou proměnnou y_ℓ a budeme požadovat pro $1 \leq \ell \leq m$: $0 \leq y_\ell \leq 1$, neboli

$$\boxed{\begin{matrix} y_\ell \leq 1 \ \wedge \\ -y_\ell \leq 0 \end{matrix}} \quad \Psi_\ell$$

  - Dále vyjádříme, že je nutno splnit každou klausuli $C_i = L_1^i \vee L_2^i \vee L_3^i$ pro $1 \leq i \leq k$. To vyjádříme nerovnicí:

$$\text{"E"} \quad l_1^i + l_2^i + l_3^i \geq 1 \quad (\varphi_i), \text{ kde}$$

$$l_j^i = \begin{cases} y_\ell & \text{pokud } L_j^i = x_\ell \\ 1 - y_\ell & \text{pokud } L_j^i = \neg x_\ell \end{cases} \quad \text{"L"}$$

  - Formule F je konjunkce a ta se převede na soustavu, kde každé klausuli bude odpovídat nerovnice φ_i

a navíc bude mít hodnotu Ψ_l

- Dá se ukázat, že:
  a) Tuto transformaci lze řešit DTS v polynomiálním čase.
  b) Je zachováno "členství" v jazyce, tedy F je splnitelná $\Leftrightarrow$ vygenerovaná instance ILP $(\wedge \varphi_i) \wedge (\wedge \Psi_l)$ má řešení!

$\square$