Uveďte základní myšlenku polynomiální redukce z problému kliky na problém SAT.

Idea důkazu:

- Uvažme libovolnou instanci problému Klika (G, k), kde $G = (V, E)$ je neorientovaný graf a $k \in \mathbb{N}^+$.
- Uzly grafu můžeme očíslovat. Nechť $V = \{v_1, \dots, v_n\}$, $n \geq 1$.
- Rovněž můžeme očíslovat uzly kliky.
- Zavedeme booleovské proměnné x_j^i, kde $1 \leq i \leq n$ a $1 \leq j \leq k$, s následujícím významem:

$$\forall 1 \leq i \leq n \ \forall 1 \leq j \leq k: \quad x_j^i = \begin{cases} \text{true} & \text{pokud } v_i \text{ je } j\text{-tý uzel kliky} \\ \text{false} & \text{jinak.} \end{cases}$$

- (G, k) zredukujeme na formuli $\varphi_1 \wedge \varphi_2 \wedge \varphi_3$, kde:
  - $\varphi_1 = \bigwedge_{1 \leq j \leq k} \bigvee_{1 \leq i \leq n} x_j^i$ – každý uzel kliky odpovídá nějakému uzlu grafu
  - $\varphi_2 = \bigwedge_{1 \leq i \leq n} \bigwedge_{1 \leq j_1 < j_2 \leq k} \neg(x_{j_1}^i \wedge x_{j_2}^i) \equiv \bigwedge_{1 \leq i \leq n} \bigwedge_{1 \leq j_1 < j_2 \leq k} \neg x_{j_1}^i \vee \neg x_{j_2}^i$

- žádný uzel grafu nereprezentuje dva různé uzly kliky.

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

- Generovaná formule má $n\cdot k$ proměnných a velikost $O(n^2k^2)$.
- Snadno se nahlédne, že se daná redukce – pro vhodné kódování instancí daných problémů – dá realizovat úplným DTS pracujícím v polynomiálním čase.
- Dá se rovněž nahlédnout, že G má kliku velikosti k $\Leftrightarrow$ $\varphi_1 \wedge \varphi_2 \wedge \varphi_3$ je splnitelná $\square$.

Lineární programování

- optimalizační problém
  - máme proměnné $x_1 \dots x_n$, $n \ge 1$, které tvoří vektor $\overline{x} = (x_1, \dots, x_n)$.
  - Máme soustavu lineárních nerovnic $\overline{A}\overline{x} \le \overline{b}$.
  - Požadujeme $\overline{x} \ge 0$.

- min. / max. $\bar{c}^T \cdot \bar{x}$

- celočíselné lineární prog. – integer lin. progr. (ILP)
  - omezení LP na celočíselné hodnoty.
- rozhodovací verze ILP: ptáme se, zda soustava $\bar{A}\bar{x} \leq \bar{b}$ má řešení $\bar{x} \in \mathbb{Z}^n$
- ILP je NP-úplné.
- Členství v NP: založeno na principu uhádnutí polynomiálně velkého řešení a ověření, že se opravdu jedná o řešení. Je ovšem nutno ukázat, že je možné se omezit na polynomiálně velká řešení!
- NP-těžkost lze ukázat např. redukcí ze 3-SAT.

Základní myšlenka:

- Uvažujme 3-SAT nad proměnnými $v_1, \dots, v_n$, $n \geq 1$.
- Formule 3-SAT $F = C_1 \wedge \dots \wedge C_m$, $m \geq 1$, kde $\forall 1 \leq i \leq m$: $C_i = L_1^i \vee L_2^i \vee L_3^i$, kde $\forall 1 \leq j \leq 3 \; \exists 1 \leq k \leq n$: $L_j^i = v_k \vee L_j^i = \neg v_k$.
- F převedeme na soustavu lineárních

nerovnic nad proměnnými $x_1, \dots, x_n$ tvořenou následujícími nerovnicemi:

- $\forall\, 1 \le k \le n$: $0 \le x_k \le 1$
- Pro $\forall\, 1 \le i \le m$:

  $e_1^i + e_2^i + e_3^i \ge 1$, kde

  $\forall\, 1 \le j \le 3$: $e_j^i = \begin{cases} x_k & \text{pokud } L_j^i = v_k \\ (1-x_k) & \text{pokud } L_j^i = \neg v_k \end{cases}$

- Snadno se nahlédne, že uvedenou redukci je možno — po vhodném kódování — implementovat úplnou DTS pracující v polynomiálním čase a že F je splnitelná $\Leftrightarrow$ daný systém má řešení.

$\square$

Uzávěrové vlastnosti tříd $\mathbb{P}$ a $\mathbb{NP}$

1. $\forall L_1, L_2 \in \mathbb{P} : L_1 \cup L_2 \in \mathbb{P}$ - Uveďte základní myšlenku důkazu

- L_1, L_2 lze přijímat DTS s polynomiální složitostí

- ke každému DTS pracujícímu v polynom. čase a přijímajícímu jazyk L lze sestrojit úplný DTS, který L rozhoduje v polynom. čase.
- Můžeme tedy předpokládat existenci DTS M_1, M_2 rozhodujících jazyky L_1, L_2 v časech $P_1(n)$, $P_2(n)$, kde $P_1(n), P_2(n)$ jsou polynomy $\in O(n^k)$ pro $k \in N$.
- Sestrojíme úplný DTS M rozhodující $L_1 \cup L_2$. M může pracovat takto:
  1. M zkopíruje obsah vstupní pásky na pásku druhou.
  2. M odsimuluje M_1 na 1. pásce a pokud M_1 přijme, přijme.
  3. Jinak M odsimuluje M_2 na 2. pásce a pokud M_2 přijme, přijme, jinak odmítne.
- DTS M evidentně přijímá (rozhoduje) $L_1 \cup L_2$, a to v čase $\underbrace{O(n)}_{\text{kopie pásky}} + P_1(n) + P_2(n) \in O(n^k)$, $k \in N$. □

- Uzavřenost $\mathbb{P}$ vůči $\cap$ lze ukázat zcela analogicky.
- Základní myšlenka uzavřenosti $\mathbb{P}$ vůči konkatenaci:
    - Mějme jazyky $L_1, L_2 \in \mathbb{P}$. K nim existují úplné DTS M_1, M_2, které je rozhodují v polynomiálním čase $P_1(n)$ / $P_2(n)$.
    - DTS M rozhodující $L_1 \cdot L_2$ musí na vstupu $a_1 a_2 \dots a_n$ pracovat tak, že postupně přepíše na dvě ~~pomocné~~ pásky všechna možná rozdělení $a_1 \dots a_n$ na prefix w_1 a sufix w_2 ($a_1 \dots a_n = w_1 \cdot w_2$). Takových rozdělení je $(n+1)$. Pokud M_1 přijme w_1 a M_2 přijme w_2 pro některé z rozdělení, přijme, jinak odmítne.
    - M rozhoduje $L_1 \cdot L_2$ v čase $O(n \cdot (P_1(n) + P_2(n) + \underset{\text{kopírování}}{n})) \in O(n^k), k \in \mathbb{N}$ $\square$
- Uzavřenost $\mathbb{P}$ vůči $*$:
    - Mějme nějaké $L \in \mathbb{P}$ a tedy DTS M přijímající L

v polynomiálním čase.

- DTS M_* přijímající L^* v polynomiálním čase nemůže pracovat tak, že uváží rozdělení vstupního řetězce na všechny možné počty podřetězců začínající a končící na všech možných místech řetězce: $O(n)$ podřetězců s $O(n)$ možnými posicemi začátku/konce každého z nich, což by vedlo na složitost $O(n)^{O(n)}$!
- Namísto výše uvedeného lze užít následující postup:
  - M vytvoří matici X rozměru $(n+1) \times (n+1)$ pro vstupní řetězec $w = a_1 \dots a_n$ délky n, kterou vyplní následovně:

$$\forall 1 \le i \le n+1 \ \forall 1 \le j \le n+1:$$

$$X[i,j] = \begin{cases} \text{true} & \text{pokud } i < j \text{ a } a_i \dots a_{j-1} \in L \\ \text{false} & \text{jinak} \end{cases}$$

  - Následně M spočte matici X^+ odpovídající tranzitivnímu uzávěru relace, kterou X reprezentuje.

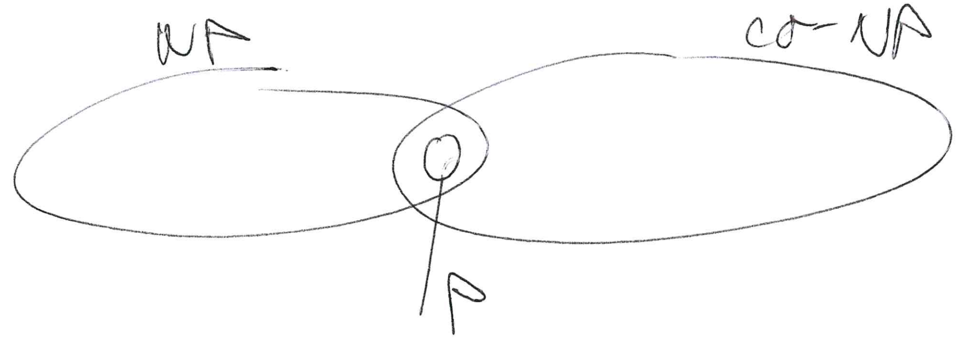
- Intuitivně $X[i,j] = true$ říká, že z indexu i na index $j-1$ je možné se v daném řetězci dostat přes řetězce z L.
- M ověří, zda $X^+[1, n+1] = true$, pokud ano, přijme, jinak odmítne. Intuitivně $X^+[1,n+1] = true$ znamená, že vstupní řetězec $a_1 \ldots a_n$ lze pokrýt několika na sebe navazujícími řetězci z L. (a tedy $a_1 \ldots a_n \in L^*$).
- M pracuje v čase

$$O\Big((n+1)^2 \cdot \underbrace{P(n)}_{\text{složitost rozhodování } L} + \overset{\text{výpočet tr. uz.}}{(n+1)^3} \Big) \in O(n^k)$$

$k \in \mathbb{N}$.

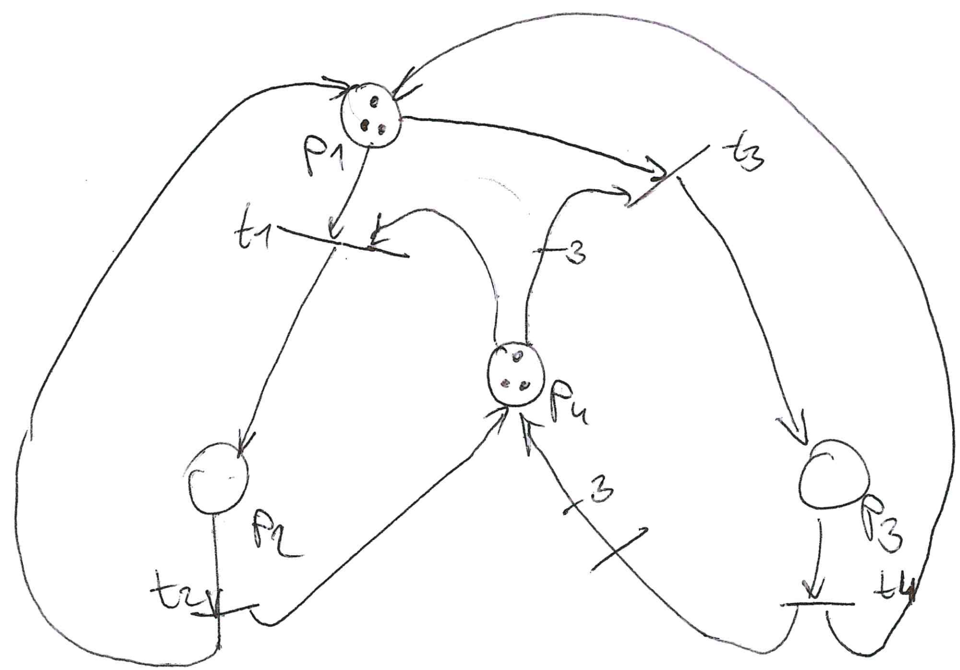
- <u>Uzavřenost NP vůči $\cup, \cap, \cdot, *$</u>
  - Lze ukázat analogicky jako u P s tím, že u $\cdot$ a $*$ je možno uhádnout rozdělení na podřetězce, není třeba systematicky procházet všechna možná rozdělení.

- Uzavřenost vůči doplňku je otevřená!



- NTS přijímající jazyk lze sestrojit, ale nelze snadno změnit režim přijímá / nepřijímá!

Petriho sítě

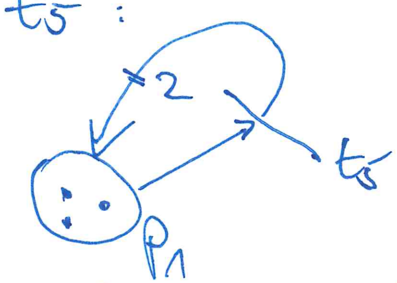


strom dosažitelných značení

(P_1, P_2, P_3, P_4)

- $(3, 0, 0, 3)$
  - $\xrightarrow{t_1}$ $(2, 1, 0, 2)$
    - $\xrightarrow{t_2}$ $\underline{(3, 0, 0, 3)}$
    - $\xrightarrow{t_1}$ $(1, 2, 0, 1)$
      - $\xrightarrow{t_2}$ $\underline{(2, 1, 0, 2)}$
      - $\xrightarrow{t_1}$ $(0, 3, 0, 0)$
        - $\xrightarrow{t_2}$ $\underline{(1, 2, 0, 1)}$
  - $\xrightarrow{t_3}$ $(2, 0, 1, 0)$
    - $\xrightarrow{t_4}$ $\underline{(3, 0, 0, 3)}$

Uvažme dále rozšíření uvedené P.s. o přechod t_5:



„t_5 generuje nové procesy (fork)"

vlastnosti:

- síť není bezpečná (více značek v místech)
- je omezená (počet značek nikde nepřesáhne nade všechny meze)
- striktně konzervativní není (součet počtu značek se mění)
- je konzervativní – vážený součet počtu značek zůstane zachován pro vektor vah $(1, 2, 4, 1)$
- je živá – každý přechod je vždy proveditelný po provedení určitého počtu dalších přechodů

strom. dosažitelnosti se změní na:

- $(3,0,0,3)$
  - $\xrightarrow{t_1}$ $(2,1,0,2)$
    - $\xrightarrow{t_2}$ $\underline{(3,0,0,3)}$
    - $\xrightarrow{t_1}$ $(1,2,0,1)$ …
    - $\xrightarrow{t_5}$ $(\omega,1,0,2)$ …
  - $\xrightarrow{t_3}$ $(2,0,1,0)$
  - $\xrightarrow{t_5}$ $[(\ldots 4,0,0,3) \geq (3,0,0,3)]$
    $(\omega,0,0,3)$ — počet roste nade všechny meze.
    - $\xrightarrow{t_1}$ $\underline{(\omega,1,0,2)}$
    - $\xrightarrow{t_3}$ $(\omega,0,1,0)$ …
    - $\xrightarrow{t_5}$ $\underline{(\omega,0,0,3)}$ …

– výsledná síť již <u>není</u> přesná ani konzervativní!