Základy složitosti a vyčíslitelnosti

NTIN090

Petr Kučera

2022/23 (9. přednáška)

Vrcholové pokrytí
a další problémy

NP-úplnost VRCHOLOVÉHO POKRYTÍ

VRCHOLOVÉ POKRYTÍ

Instance: Neorientovaný graf $G = (V, E)$ a celé číslo $k \geq 0$.

Otázka: Existuje množina vrcholů $S \subseteq V$ velikosti nejvýš k, která obsahuje alespoň jeden vrchol z každé hrany $\{u, v\} \in E$ (tedy $\{u, v\} \cap S \neq \emptyset$)?

Věta

3-SAT je polynomiálně převoditelný na VRCHOLOVÉ POKRYTÍ.

- VRCHOLOVÉ POKRYTÍ patří do NP
- Polynomiální verifikátor ověřuje, zda množina vrcholů S tvoří vrcholové pokrytí velikosti nejvýš k

Důsledek

VRCHOLOVÉ POKRYTÍ je NP-úplné.

Problémy související s Vrcholovým pokrytím

Klika (Clique) Obsahuje G jako podgraf kliku, tj. úplný graf, na k vrcholech?

Nezávislá množina (Independent Set): Obsahuje G nezávislou množinu velikosti k?

- Množina vrcholů S je nezávislá, pokud mezi žádnými dvěma vrcholy z S nevede hrana.

Hranového pokrytí (Edge Cover) Existuje v G množina hran velikosti nejvýš k, která pokrývá všechny vrcholy?

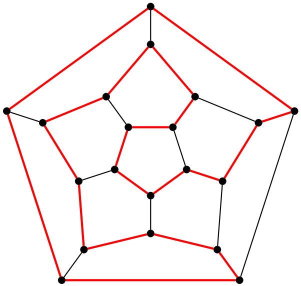
- Problémy Klika a Nezávislá množina jsou NP-úplné
- Problém Hranového pokrytí je řešitelný v polynomiálním čase

Hamiltonovská kružnice

Hamiltonovská kružnice (HK, Hamiltonian cycle)

Instance: Neorientovaný graf $G = (V, E)$.

Otázka: Existuje v grafu G cyklus vedoucí přes všechny vrcholy?



Věta (Bez důkazu)

Hamiltonovská kružnice je NP*-úplný problém.*

Třírozměrné párování

Třírozměrné párování (3DM, 3D Matching)

Instance: Množina $M \subseteq W \times X \times Y$, kde W, X a Y jsou množiny velikosti q.

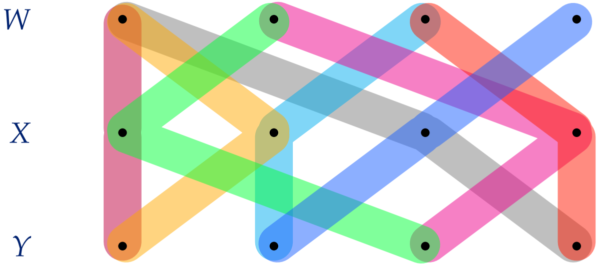
Otázka: Má M perfektní párování? Tj. lze v M vybrat q po dvou disjunktních trojic?

Věta (Bez důkazu)

Třírozměrné párování je NP*-úplný problém.*

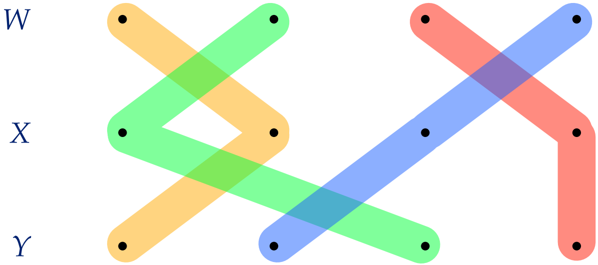
Příklad 3DM

Množina trojic $M \subseteq W \times X \times Y$



Příklad 3DM

Perfektní párování $M' \subseteq M$



Loupežníci

Loupežníci (Partition)

Instance: Množina předmětů A, s každým předmětem $a \in A$ asociované přirozené číslo $s(a)$ (váha, cena, velikost).

Otázka: Existuje $A' \subseteq A$, pro kterou platí, že

$$\sum_{a \in A'} s(a) = \sum_{a \in A \setminus A'} s(a)?$$

Věta (Bez důkazu)

Loupežníci je NP-úplný problém.

Batoh

BATOH (KNAPSACK)

Instance: Množina předmětů A, s každým předmětem $a \in A$ asociovaná velikost $s(a) \in \mathbb{N}$ a cena $v(a) \in \mathbb{N}$, velikost batohu $B \in \mathbb{N}$ a limit na cenu $K \in \mathbb{N}$.

Otázka: Lze vybrat množinu předmětů $A' \subseteq A$ tak, aby platilo

$$\sum_{a \in A'} s(a) \leq B \text{ a } \sum_{a \in A'} v(a) \geq K?$$

Věta

BATOH je NP-úplný problém.

- NP-těžkost lze ukázat snadným převodem z LOUPEŽNÍKŮ.

Rozvrhování

Rozvrhování (Scheduling)

Instance: Množina úloh U, s každou úlohou $u \in U$ asociovaná doba zpracování $d(u) \in \mathbb{N}$, počet procesorů m, limit $D \in \mathbb{N}$.

Otázka: Lze úlohy U rozdělit na m procesorů tak, aby byly všechny úlohy zpracované v časovém limitu D?

Věta

Rozvrhování je NP-úplný problém.

- NP-těžkost lze ukázat snadným převodem z Loupežníků.

Třída co-NP

Tautologie

Tautologie výroková formule, která je splněna každým ohodnocením

Příklady tautologií

Sylogismus $[(x \implies y) \wedge (y \implies z)] \implies (x \implies z)$

Kontrapozice $(x \implies y) \iff (\neg y \implies \neg x)$

De Morganova pravidla $\neg(x \vee y) \iff (\neg x \wedge \neg y)$

Důkaz sporem $[(\neg x \implies y) \wedge (\neg x \implies \neg y)] \implies x$

Problém tautologie

TAUTOLOGIE (TAUT)

Instance: Výroková formule φ

Otázka: Je φ tautologie?

Patří TAUT do NP?

- Jak ověřit, zda daná formule je tautologie?
  - Zkontrolovat pravdivostní tabulku
  - Důkaz z axiomů výrokové logiky
  - …

Není znám polynomiální verifikátor pro TAUT.

TAUT jako jazyk

$$\mathrm{TAUT} = \{\langle\varphi\rangle \mid \varphi \text{ je tautologie}\}$$

- Předpokládejme, že každý řetězec kóduje nějakou výrokovou formuli
- Pak doplněk TAUT je definován takto

$$\overline{\mathrm{TAUT}} = \{\langle\varphi\rangle \mid \varphi \text{ není tautologie}\}$$

- φ není tautulogie $\iff \varphi(\mathbf{a}) = 0$ pro nějaké ohodnocení $\mathbf{a}$
- Polynomiální verifikátor $V(\varphi, \mathbf{a})$ pro $\overline{\mathrm{TAUT}}$ ověří, zda $\varphi(\mathbf{a}) = 0$

$\overline{\mathrm{TAUT}}$ patří do NP

Třída co-NP

Definice

$$\mathrm{co\text{-}NP} = \{A \mid \overline{A} \in \mathrm{NP}\}$$

- Jazyk A patří do $\mathrm{co\text{-}NP}$, právě když existuje polynomiální verifikátor V a polynom $p(n)$, pro něž platí

$$A = \left\{x \mid (\forall y \in \{0,1\}^{p(|x|)})\,[V(x,y) \text{ přijme}]\right\}.$$

TAUT patří do $\mathrm{co\text{-}NP}$

co-NP-úplnost

Definice

Jazyk B je

co-NP-**těžký** pokud pro každý jazyk $A \in \text{co-NP}$ platí $A \leq_m^P B$

co-NP-**úplný** pokud je co-NP-těžký a současně $B \in \text{co-NP}$

Věta

Jazyk A je co-NP-úplný, právě když $\overline{A}$ je NP-úplný

Důkaz.

Plyne z faktu, že pro každé dva jazyky A a B

$$A \leq_m^P B \iff \overline{A} \leq_m^P \overline{B}$$

□

Dva co-NP-úplné problémy

- Jazyky splnitelných a nesplnitelných formulí

$$\mathrm{SAT} = \{\langle\varphi\rangle \mid \varphi \text{ je splnitelná}\}$$
$$\overline{\mathrm{SAT}} = \{\langle\varphi\rangle \mid \varphi \text{ je nesplnitelná}\}$$

- Problém SAT je NP-úplný
- Z toho plyne, že $\overline{\mathrm{SAT}}$ je $\mathrm{co\text{-}NP}$-úplný
- $\overline{\mathrm{SAT}} \le_m^P \mathrm{TAUT}$ funkcí $f(\langle\varphi\rangle) = \langle\neg\varphi\rangle$

Problémy $\overline{\mathrm{SAT}}$ a TAUT jsou $\mathrm{co\text{-}NP}$-úplné.

NP vs. $\mathrm{co}\text{-}\mathrm{NP}$

Věta

Pokud nějaký NP-úplný problém A patří do $\mathrm{co}\text{-}\mathrm{NP}$, pak $\mathrm{NP} = \mathrm{co}\text{-}\mathrm{NP}$.

- Protože $A \in \mathrm{co}\text{-}\mathrm{NP}$, platí $\overline{A} \in \mathrm{NP}$
- Uvážíme-li $B \in \mathrm{NP}$, pak $B \leq_m^P A$ (z NP-úplnosti A)

$\Longrightarrow$ $\overline{B} \leq_m^P \overline{A}$, tedy $\overline{B} \in \mathrm{NP}$

$\Longrightarrow$ $B \in \mathrm{co}\text{-}\mathrm{NP}$

- Z toho plyne $\mathrm{NP} \subseteq \mathrm{co}\text{-}\mathrm{NP}$
- Díky symetrii též $\mathrm{co}\text{-}\mathrm{NP} \subseteq \mathrm{NP}$

Nevíme, zda platí $\mathrm{NP} = \mathrm{co}\text{-}\mathrm{NP}$ nebo $\mathrm{NP} \neq \mathrm{co}\text{-}\mathrm{NP}$

P a rozhodnutelnost

Třída DEC

Jazyky rozhodnutelné Turingovými stroji

Třída P

Jazyky rozhodnutelné Turingovými stroji v polynomiálním čase

NP a částečná rozhodnutelnost

Třída PD (částečně rozhodnutelné)

- Přijímané Turingovými stroji
- Ověřitelné jazyky
- $A \in \mathrm{PD} \iff$ pro nějaký $B \in \mathrm{DEC}$

$$A = \{x \mid (\exists y \in \{0,1\}^*)[\langle x, y\rangle \in B]\}$$

Třída NP (nedeterministicky polynomiální)

- Přijímané nedeterministickými TS v polynomiálním čase
- Polynomiálně ověřitelné jazyky
- $A \in \mathrm{NP} \iff$ pro nějaký $B \in \mathrm{P}$ a polynom $p(n)$

$$A = \{x \mid (\exists y \in \{0,1\}^{p(|x|)})[\langle x, y\rangle \in B]\}$$

$\mathrm{co\text{-}NP}$ a částečná rozhodnutelnost

Třída $\mathrm{co\text{-}PD}$

- Doplňky částečně rozhodnutelných jazyků
- $A \in \mathrm{co\text{-}PD} \iff$ pro nějaký $B \in \mathrm{DEC}$

$$A = \{x \mid (\forall y \in \{0,1\}^*)[\langle x, y\rangle \in B]\}$$

Třída $\mathrm{co\text{-}NP}$

- Doplňky jazyků v NP
- $A \in \mathrm{co\text{-}NP} \iff$ pro nějaký $B \in \mathrm{P}$ a polynom $p(n)$

$$A = \{x \mid (\forall y \in \{0,1\}^{p(|x|)})[\langle x, y\rangle \in B]\}$$

Souvislost s Postovou větou

Dle Postovy věty

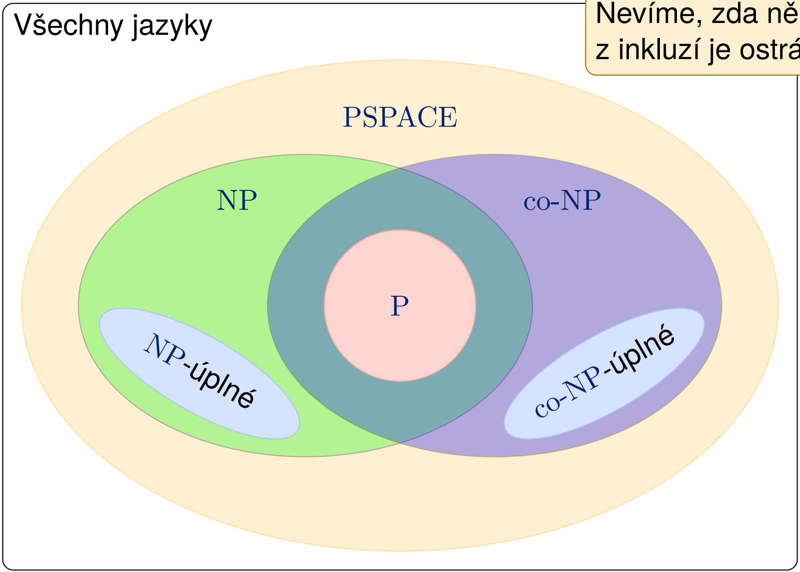
$$\boxed{\mathrm{DEC} = \mathrm{PD} \cap \text{co-PD}}$$

zatímco víme pouze

$$\boxed{\mathrm{P} \subseteq \mathrm{NP} \cap \text{co-NP}}$$

- Obvykle se předpokládá, že inkluze je ostrá
- Nicméně, neumíme vyloučit, že $\mathrm{P} = \mathrm{NP} = \text{co-NP}$

Vztahy mezi třídami



Nevíme, zda některá z inkluzí je ostrá

Třída $\#\mathrm{P}$

Počítání modelů

$\#\mathrm{SAT}$

Instance: Formule φ v KNF

Hodnota: Počet modelů φ

> Pokud $\#\mathrm{SAT}$ lze vyčíslit v polynomiálním čase, pak $\mathrm{P} = \mathrm{NP}$

- Polynomiální algoritmus pro $\#\mathrm{SAT}$ bychom mohli použít k rozhodnutí SAT

$$\varphi \text{ je splnitelná} \iff \#\mathrm{SAT}(\varphi) > 0$$

- Těžkost $\#\mathrm{SAT}$ je způsobená těžkostí SAT

Počítání cyklů

$\#\mathrm{CYCLE}$

Instance: Orientovaný graf $G = (V, E)$

Hodnota: Počet cyklů G

- Snadné ověřit, zda G obsahuje cyklus
- Těžké určit počet cyklů v G

Věta

Pokud $\#\mathrm{CYCLE}$ lze vyčíslit v polynomiálním čase, pak $\mathrm{P} = \mathrm{NP}$

- Ukážeme, že algoritmus vyčíslující $\#\mathrm{CYCLE}$ lze použít k rozhodnutí problému Hamiltonovské kružnice
- Problém Hamiltonovské kružnice NP-úplný

Orientovaná Hamiltonovská kružnice

ORIENTOVANÁ HAMILTONOVSKÁ KRUŽNICE (OHK)

Instance: Orientovaný graf $G = (V, E)$.

Otázka: Obsahuje G cyklus délky $n = |V|$?

Věta (Bez důkazu)

Problém OHK je NP-úplný.

Rozhodnutí OHK počítáním cyklů

- Předpokládejme, že $\#\mathrm{CYCLE}$ lze vyčíslit v polynomiálním čase
- Ukážeme, že OHK lze rozhodnout v polynomiálním čase
- Popíšeme polynomiální převod orientovaného grafu G na orientovaný graf G'

Orientovaný graf G → převedeme → Orientovaný graf G'

$G = (V, E)$ má Hamiltonovskou kružnici $\iff$ $\#\mathrm{CYCLE}(G') \geq n^{n^2}$ kde $n = |V|$

Počet vrcholů G

Předpoklad: $n = 2^k$ pro nějaké $k \in \mathbb{N}$

V opačném případě upravíme G takto:

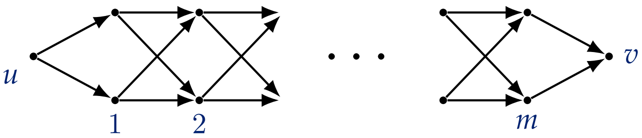
- Položme $k = \lceil \log_2 n \rceil$
- Vybereme vrchol $s \in V$
- Nahradíme s dvěma vrcholy s_{in} a s_{out} s hranami

$$E_{\mathrm{in}} = \{(u, s_{\mathrm{in}}) \mid (u, s) \in E\}$$
$$E_{\mathrm{out}} = \{(s_{\mathrm{out}}, u) \mid (s, u) \in E\}$$

- Spojíme s_{in} s s_{out} cestou procházející $2^k - n - 1$ nově přidanými vrcholy
- Výsledný graf
  - má 2^k vrcholů a
  - obsahuje hamiltonovskou kružnici, právě když ji obsahuje G

Konstrukce G'

Každou hranu (u, v) nahradíme následujícím podgrafem



- Podgraf má $m = n \log_2 n$ úrovní
- Podgraf je acyklický
  - Cykly G' odpovídají cyklům G
- Podgraf obsahuje 2^m orientovaných cest z u do v

Cyklus v G délky ℓ dává vzniknout $(2^m)^\ell$ cyklům v G'.

Použití G' pro rozhodnutí OHK

- Nechť G obsahuje hamiltonovskou kružnici
  - Počet cyklů v G' je alespoň

$$(2^m)^n = (2^{n \log_2 n})^n = n^{n^2}$$

- Nechť G neobsahuje hamiltonovskou kružnici
  - Každý cyklus v G má délku nejvýš $n - 1$
  - G obsahuje nejvýš n^{n-1} cyklů
  - Počet cyklů v G' je tedy nejvýš

$$(2^m)^{n-1} \cdot n^{n-1} = \left(2^{n \log_2 n} \cdot n\right)^{n-1} = n^{(n+1)(n-1)} = n^{n^2-1} < n^{n^2}$$

G obsahuje hamiltonovskou kružnici $\iff \#\mathrm{CYCLE}(G') \geq n^{n^2}$.

Třída $\#\mathrm{P}$

Definice

Funkce $f : \Sigma^* \to \mathbb{N}$ patří do třídy $\#\mathrm{P}$, pokud existuje polynomiální verifikátor V a polynom $p(n)$ takové, že pro každý $x \in \Sigma^*$ platí

$$f(x) = |\{y \in \{0,1\}^{p(|x|)} \mid V(x,y) \text{ přijme}\}|.$$

- Alternativně, f patří do $\#\mathrm{P}$, pokud pro nějaký polynomiální NTS M platí

$$f(x) = \text{počet přijímajících výpočtů } M(x)$$

Třídy NP a $\#\mathrm{P}$

Třída NP zachycuje problémy, v nichž se ptáme po existenci polynomiálního certifikátu

- Existuje polynomiální certifikát pro daný vstup?

Třída $\#\mathrm{P}$ zachycuje problémy, v nichž chceme určit počet certifikátů

- Kolik certifikátů pro daný vstup existuje?

> Počítání certifikátů může být těžší než hledání jednoho

Příklad

Uvažme rozdíl mezi

- ověřováním acykličnosti orientovaného grafu (lehké) a
- počítáním cyklů v orientovaného grafu (těžké)

Pozitivita funkce f

Pozitivita f

Instance: $x \in \Sigma^*$

Otázka: $f(x) > 0$?

Věta

Je-li $f \in \#\mathrm{P}$, pak Pozitivita f patří do NP.

- Nechť V je polynomiální verifikátor a $p(n)$ polynom, které splňují

$$f(x) = |\{y \in \{0,1\}^{p(|x|)} \mid V(x,y) \text{ přijme}\}|$$

- Pak V je polynomiální verifikátor pro Pozitivita f
- Pozitivita f tedy patří do NP.

Počítání pomocí rozhodování

Věta

Pro funkci $f \in \#\mathrm{P}$ je otázka náležení do množiny $S_f = \{(x, N) \mid f(x) \geq N\}$ výpočetně ekvivalentní (až na polynomiální faktor) výpočtu f.

1. $(x, N) \in S_f$ lze jednoduše rozhodnout se znalostí hodnoty $f(x)$
2. $f(x)$ lze určit pomocí polynomiálního počtu dotazů typu „$(x, N) \in S_f$?“
   - Nechť V je polynomiální verifikátor a $p(n)$ polynom, které splňují

$$f(x) = |\{y \in \{0,1\}^{p(|x|)} \mid V(x, y) \text{ přijme}\}|$$

   - Tedy $0 \leq f(x) \leq 2^{p(|x|)}$
   - Použijeme binární vyhledávání k určení hodnoty $f(x)$
   - Stačí $O(p(|x|))$ dotazů

Prostorová složitost $\#\mathrm{P}$

Věta

Hodnotu $f \in \#\mathrm{P}$ lze vyčíslit v polynomiálním prostoru

- Nechť V je polynomiální verifikátor a $p(n)$ polynom, které splňují

$$f(x) = |\{y \in \{0,1\}^{p(|x|)} \mid V(x,y) \text{ přijme}\}|$$

- K určení hodnoty $f(x)$ stačí pustit $V(x,y)$ pro každý řetězec $y \in \{0,1\}^{p(|x|)}$
- $f(x)$ je pak počet přijatých certifikátů

Polynomiální převoditelnost funkcí

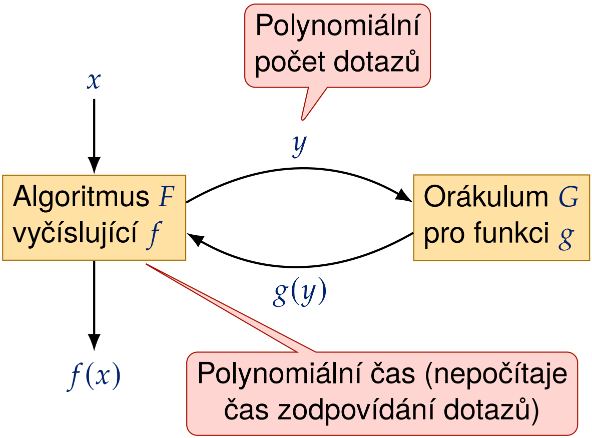
Definice

Funkce f je polynomiálně převoditelná na funkci g ($f \leq_P g$), pokud lze f vyčíslit v polynomiálním čase algoritmem, který má přístup k orákulu funkce g.

- Algoritmus vyčíslující $f(x)$
  - Má přístup k orákulu, které určí $g(y)$ pro libovolný řetězce y
  - Orákulu může položit polynomiální počet dotazů
  - Algoritmus pracuje v polynomiálním čase, za předpokladu, že orákulum odpovídá okamžitě

Převoditelnost funkcí (princip)

$f \leq_P g$ = „pomocí g umíme spočítat f v polynomiálním čase“



#P-úplnost

Definice

Funkce $g \in \#\mathrm{P}$ je $\#\mathrm{P}$-úplná, pokud $f \leq_P g$ pro každou funkci $f \in \#\mathrm{P}$.

Věta

Funkce $\#\mathrm{SAT}$ je $\#\mathrm{P}$-úplná.

- Plyne z převodu, kterým jsme ukazovali NP-úplnost SAT
- Podívejme se na to podrobněji

#P-úplnost #SAT

- Uvažme $f \in \#\mathrm{P}$
- Nechť V je polynomiální verifikátor a $p(n)$ polynom, které splňují

$$f(x) = |\{y \in \{0,1\}^{p(|x|)} \mid V(x,y) \text{ přijme}\}|$$

- Uvažme následující nedeterministický TS M

Výpočet M se vstupem x
Nedeterministicky vyber $y \in \{0,1\}^{p(\|x\|)}$
Pusť $V(x,y)$
if $V(x,y)$ přijal **then** přijmi **else** odmítni

$$f(x) = \text{počet přijímajících výpočtů } M(x)$$

Vyčíslení f pomocí $\#\mathrm{SAT}$

- V důkazu Cookovy-Levinovy věty jsme popsali konstrukci KNF φ, která s každým hodnocením $\mathbf{a}$ splňuje

$$\varphi(\mathbf{a}) = 1 \iff \mathbf{a} \text{ reprezentuje přijímající výpočet } M(x)$$

- Navíc platí, že mezi modely $\mathbf{a}$ formule φ a přijímajícími výpočty $M(x)$ je bijekce
- Platí tedy $f(x) = \#\mathrm{SAT}(\varphi)$
- Následující algoritmus vyčísluje $f(x)$:
  1. V polynomiálním čase zkonstruuj φ
  2. Vrať $\#\mathrm{SAT}(\varphi)$

$$\boxed{f \leq_P \#\mathrm{SAT}}$$

Parsimonious převod

Definice

Polynomiální převod z problému A na problém B je parsimonious, pokud zachovává počet certifikátů.

- Je-li A převoditelný na B parsimonious převodem, pak $\#A \leq_P \#B$
- Převod z $A \in \mathrm{NP}$ do SAT, který jsme popsali v důkazu Cookovy-Levinovy věty, je parsimonious
- funkce $\#\mathrm{SAT}$ je proto $\#\mathrm{P}$-úplná

Disjunktivní normální forma

Literál výroková proměnná x nebo její negace $\neg x$

Term konjunkce literálů

- Například $x \wedge \neg y \wedge \neg z$
- Prázdný term je splněný ($\top$)

DNF formule je v disjunktivní normální formě, pokud jde o disjunkci termů

Příklad

Následující formule je v DNF

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

Splnitelnost DNF

- V polynomiálním čase lze rozhodnout, zda DNF ψ je splnitelná
  - Ověříme, zda je splnitelný jeden z termů ψ
  - Term T je splnitelný $\iff$ T neobsahuje $x \wedge \neg x$ pro žádnou proměnnou x
- Pro danou KNF φ lze jednoduše zkonstruovat DNF $\psi \equiv \neg\varphi$
  - Použijeme De Morganova pravidla

$$\neg(a \wedge b) \equiv \neg a \vee \neg b \quad \text{and} \quad \neg(a \vee b) \equiv \neg a \wedge \neg b$$

Uvažme KNF

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

Aplikací De Morganových pravidel na $\neg\varphi$ dostaneme DNF $\psi \equiv \neg\varphi$

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

Počítání modelů DNF

- Definujeme funkci $\#\,\mathrm{DNF\text{-}SAT}$, která pro každou DNF ψ splňuje

$$\#\,\mathrm{DNF\text{-}SAT}(\psi) = |\{\mathbf{a} \mid \psi(\mathbf{a}) = 1\}|$$

- $\#\,\mathrm{DNF\text{-}SAT}$ patří do $\#\mathrm{P}$

Věta

$\#\mathrm{SAT} \leq_P \#\,\mathrm{DNF\text{-}SAT}$, *tedy funkce* $\#\,\mathrm{DNF\text{-}SAT}$ *je* $\#\mathrm{P}$*-úplná.*

Výpočet $\#\mathrm{SAT}(\varphi)$ s pomocí $\#\,\mathrm{DNF\text{-}SAT}$

Vstup: KNF φ

$n \leftarrow$ počet proměnných v φ

$\psi \leftarrow$ DNF ekvivalentní $\neg\varphi$

return $2^n - \#\,\mathrm{DNF\text{-}SAT}(\psi)$

Počet perfektních párování v bipartitním grafu

Perfektní párování v bipartitním grafu (BPM)

Instance: Bipartitní graf $G = (V = A \cup B, E \subseteq A \times B)$, kde $|A| = |B|$.

Otázka: Existuje v G párování velikosti $|A| = |B|$?

Věta (Bez důkazu)

Funkce $\#\mathrm{BPM}$ *je* $\#\mathrm{P}$*-úplná.*

Permanent matice

Definice

Je-li A matice typu $n \times n$ definujeme permanent A jako

$$\mathrm{perm}(A) = \sum_{\pi \in S(n)} \prod_{i=1}^{n} a_{i,\pi(i)},$$

kde $S(n)$ je množina permutací množiny $\{1, \ldots, n\}$.

- „Determinant“, kde neuvažujeme znaménko permutace.
- Je-li A matice sousednosti bipartitního grafu G, pak $\mathrm{perm}(A)$ určuje počet perfektních párování G.

Věta (Bez důkazu)

Funkce perm *je* $\#\mathrm{P}$*-úplná.*