Teoretická informatika – 3. domácí úloha

Ondřej Ondryáš (xondry02), 18. prosince 2022

Příklad 1

Důkaz provedeme redukcí z $co\text{-}HP$, který není ani částečně rozhodnutelný.

Navrhneme redukci $\sigma : \{0, 1, \#\}^* \to \{0, 1, \#\}^*$ z $co\text{-}HP$ na L: $co\text{-}HP \leq L$, kde

$$co\text{-}HP = \{\langle M\rangle\#\langle w\rangle \mid w \in \Sigma^* \wedge M \text{ je TS, který na } w \text{ nezastaví}\},$$
$$L = \{\langle M_1\rangle\#\langle M_2\rangle \mid M_1 \text{ a } M_2 \text{ jsou TS takové, že } L(M_1) \leq L(M_2)\}$$

přičemž $\langle\cdot\rangle$ značí standardní kódování TS nebo jeho vstupu.

Idea je následující:

1. $x \in co\text{-}HP$ („$co\text{-}HP$ odpoví YES“) $\Leftrightarrow$ M na w nezastaví $\rightsquigarrow L(M_1) \leq L(M_2)$,
   - což platí např. pro $L(M_1) = \emptyset$, $L(M_2) = \emptyset$ (viz níže).
2. $x \notin co\text{-}HP$ („$co\text{-}HP$ odpoví NO“) $\Leftrightarrow$ M na w zastaví $\rightsquigarrow L(M_1) \not\leq L(M_2)$,
   - což platí např. pro $L(M_1) = \Sigma^*$, $L(M_2) = \emptyset$ (viz níže).

σ přiřadí řetězci $x \in \{0, 1, \#\}^*$ řetězec $\langle M_1\rangle\#\langle M_2\rangle$, kde M_2 je TS, který nepřijímá žádný vstup ($L(M_2) = \emptyset$), a M_1 je TS pracující následovně:

1. M_1 smaže svůj vstup.
2. M_1 zapíše na pásku řetězec x.
3. M_1 ověří, zda x má strukturu $\langle M\rangle\#\langle w\rangle$ pro nějaký TS M a jeho vstup w. Pokud ne, M_1 přijme.
4. M_1 spustí simulaci M na w.
5. Pokud simulace doběhne, M_1 přijme. Pokud simulace cyklí, M_1 cyklí.

σ lze snadno realizovat pomocí úplného TS M_σ, který se v principu skládá z 5 sekvenčně propojených komponent, které vypisují konstantní řetězce nebo řetězce závisející na x:

1. Smazání vstupu: M_σ zapíše příslušný kód TS, který prochází vstup zleva doprava a nahrazuje symboly za Δ, po zjištění koncového Δ se vrací na začátek.
2. Zapsání řetězce $x = a_1a_2 \ldots a_n$ na vstupní pásku: M_σ zapíše kód TS, který pro $\forall i : 1 \leq i \leq n$ posouvá hlavu doprava a zapisuje a_i.
3. Test správného formátu: M_σ zapíše příslušný kód TS, který prochází vstupní řetězec a testuje jeho syntaxi.
4. M_σ zapíše vhodný kód univerzálního TS. V tento moment zapsal celý $\langle M_1\rangle$.
5. M_σ zapíše konstantní řetězec $\#\langle M_2\rangle$, kde $\langle M_2\rangle$ je kód předem zvoleného TS M_2, pro který $L(M_2) = \emptyset$.

Studujme jazyk TS M_1. Existují právě dva případy, neboť M_1 maže svůj vstup:

1. $L(M_1) = \emptyset \Leftrightarrow M$ na w nezastaví $\Leftrightarrow x \in co\text{-}HP$.
2. $L(M_1) = \Sigma^* \Leftrightarrow x$ nemá požadovanou strukturu $\vee$ M na w zastaví $\Leftrightarrow x \notin co\text{-}HP$.

Ukažme nyní zachování členství při použití redukční funkce σ:

- Nejprve nahlédněme, že $\emptyset \leq \emptyset$: Nechť $\sigma_a(w) = w$. Pak σ_a je pro lib. abecedu Σ zjevně totální, rekurzivně vyčíslitelná funkce $\sigma_a : \Sigma^* \to \Sigma^*$ a $\forall w \in \Sigma^* : w \in \emptyset \leftrightarrow \sigma_a(w) \in \emptyset$. σ_a je tedy redukcí $\emptyset \leq \emptyset$.

- Dále nahlédněme, že $\Sigma^* \not\leq \emptyset$: Zřejmě pro žádnou abecedu Σ nemůže existovat totální funkce $\sigma_b(w)$ taková, že $\forall w \in \Sigma^* : w \in \Sigma^* \leftrightarrow \sigma_b(w) \in \emptyset$ (nehledě na volbu Σ je levá strana bikondicionálu vždy pravdivá, zatímco pravá je vždy nepravdivá).
- Jak bylo ukázáno výše, $\forall x \in \{0,1,\#\}^* : \sigma(x) \in L \Leftrightarrow \sigma(x) = \langle M_1 \rangle \# \langle M_2 \rangle$, kde M_1 a M_2 jsou TS takové, že $M_1 \leq M_2 \Leftrightarrow x$ má strukturu $\langle M \rangle \# \langle x \rangle$ a M na w nezastaví $\Leftrightarrow x \in co\text{-}HP$. □

Příklad 2

$$L_R = \{\langle M \rangle \# \langle k \rangle \mid k > 0 \wedge M \subseteq \mathbb{N} \wedge M \text{ je konečná} \wedge \exists R \subseteq 2^M : \\ (\exists \sim \subseteq M \times M : R = M_{/\sim}) \wedge |R| \leq k \wedge \forall r \in R : \forall i,j \in r : i \neq j \rightarrow NSD(i,j) = 1\}$$

Důkaz NP-úplnosti zadaného problému (formalizovaného jazykem L_R výše) provedeme tak, že ukážeme existenci polynomiální redukce ze známého NP-úplného problému na uvedený problém a dále ukážeme, že $L_R \in NP$.

Důkaz NP-těžkosti

Využijeme problému k-barvení grafu, což je známý[1] NP-úplný problém rozhodnutí, zda je možné daný neorientovaný graf *obarvit* maximálně k barvami, tedy jestli existuje funkce, která jednotlivým uzlům přiřazuje čísla od 1 do k tak, že žádným dvěma sousedním uzlům nepřiřadí stejné číslo:

$$K - COLOURING = \{\langle (V,E) \rangle \# \langle k \rangle \mid \exists (\phi : V \rightarrow \{1,2,\ldots,k\}) : \forall u,v \in V : \{u,v\} \in E \rightarrow \phi(u) \neq \phi(v)\}$$

Předpokládáme-li unární kódování v obou jazycích, hledáme funkci $\sigma : \{0,1,\#\}^* \rightarrow \{0,1,\#\}^*$, takovou, že:

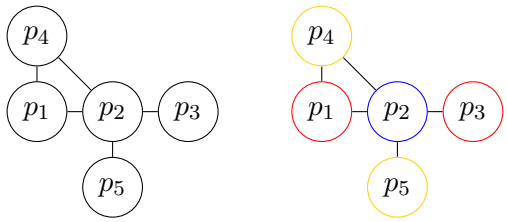
$$\forall w \in \{0,1,\#\}^* : w \in K - COLOURING \leftrightarrow \sigma(w) \in L_R$$

a σ je vyčíslitelná deterministickým Turingovým strojem v polynomiálním čase. Pokud taková existuje, platí:

$$K - COLOURING \leq_P^m L_R$$

Idea

Idea redukční funkce je takováto: Funkce „ohodnocuje“ uzly grafu čísly, která jistým způsobem postupně upravuje a nakonec je zapíše jako cílovou množinu M. Funkce pracuje s frontou prvočísel. Na počátku ohodnotí uzly postupně prvními $|V|$ prvočísly.



(a) před obarvením

(b) po jednom z možných 3-obarvení

Obrázek 1: Ukázkový graf, který je 3-obarvitelný. p_i označuje i-té prvočíslo.

[1] KARP, R. M. *Reducibility Among Combinatorial Problems*. New York: Plenum Press, 1972.

Následně postupně prochází uzly, pro každý uzel u z fronty vezme $\delta(u)$ prvočísel[2] a vynásobí jimi ohodnocení uzlu u. Dále vždy pouze jedním z těchto prvočísel postupně násobí každého následníka uzlu u. Například pro uzel s počátečním ohodnocením p_1 na obr. 1 by použila prvočísla p_6, p_7 a upravila by ohodnocení takto:

$$\begin{aligned} p_1 &\rightsquigarrow p_1 \cdot p_6 \cdot p_7 \\ p_4 &\rightsquigarrow p_4 \cdot p_6 \\ p_2 &\rightsquigarrow p_2 \quad\quad \cdot p_7 \end{aligned}$$

Tento proces se dá ekvivalentně popsat tak, že po počátečním přiřazení $|V|$ prvočísel je každé hraně v grafu přiřazeno unikátní prvočíslo, kterým se vynásobí ohodnocení obou uzlů, které tuto hranu tvoří. Je tedy vidět, že celkem bude užito prvních $|V| + |E|$ prvočísel. Protože hran může být maximálně $|V| \cdot (|V| - 1)/2$, počet generovaných prvočísel bude v $\mathcal{O}(|V|^2)$.

Konečný stav je zřejmě takový, že každé dva uzly, mezi kterými je hrana, mají soudělná ohodnocení, a zároveň každé dva uzly, mezi kterými není hrana, mají nesoudělná ohodnocení. Zároveň evidentně žádné uzly nemohou mít stejné ohodnocení, tedy pokud ohodnocení „vložíme“ do množiny M, bude $|V| = |M|$ (existuje bijekce mezi V a M). Z definice problému obarvitelnosti zároveň plyne, že pokud mezi uzly není hrana, mohou mít potenciálně stejné obarvení. Rozdělíme-li nyní V na třídy podle barvy uzlů, je zřejmé, že v každé třídě budou výhradně uzly se vzájemně nesoudělnými ohodnoceními, a zároveň tříd bude právě k. Obdobně tedy můžeme rozdělit M na právě k tříd, které budou obsahovat vždy pouze nesoudělená čísla: pokud ψ je popsaná bijekce $V \to M$, zavedeme funkci $\phi'(n) = \phi(\psi^{-1}(n))$ a relace ekvivalence vytvářející požadovaný rozklad M je pak $k \sim l \Leftrightarrow \phi'(k) = \phi'(l)$.

Implementace

Existenci polynomiální redukční funkce ukážeme konstrukcí příslušného DTS M_σ, který pracuje takto:

1. M_σ ověří, zda má jeho vstup validní tvar $\langle (V, E) \rangle \# \langle k \rangle$. Pokud ne, vrací řetězec nepatřící do L_R, např. $\langle \{2, 4\} \rangle \# \langle 1 \rangle$. Syntaktickou správnost je možné ověřit lineárním průchodem v $\mathcal{O}(n)$, sémantickou s vhodným kódováním také v polynomiálním čase: množinu uzlů je možné pro účely problému bez újmy na obecnosti reprezentovat pouze jako počet uzlů, pak v množině hran je nutné pro každý vyskytující se uzel ověřit, že je menší než $|V|$ (lze v $\mathcal{O}(n)$), a pro každou hranu ověřit, že se nerovná s jinou hranou (pro vyloučení multihran, lze cca v $\mathcal{O}(n^3)$).

2. M_σ spočítá počet uzlů $|V|$ a počet hran $|E|$ grafu zakódovaného ve vstupním řetězci a uloží si je na druhou pásku. Možné provést lineárním průchodem nad vstupem, čili v $\mathcal{O}(n)$ (konečně, mohl by to provést současně s ověřováním validity vstupu).

3. M_σ vypočítá prvních $|V| + |E|$ prvočísel p_i a uloží je na třetí pásku. Je možné ukázat, že toto je možné provést v polynomiálním čase[3].

4. M_σ postupně prochází prvočísla $p_{|V|+j}$ pro $1 \leq j \leq |E|$ a zároveň s nimi odpovídající j-tou hranu v kódu E na první pásce. Pro každou hranu určí indexy k, l uzlů spojených touto hranou a prvočíslem $p_{|V|+j}$ vynásobí čísla na pozicích k a l na třetí pásce.

   - Posun po pásce s prvočísly a zároveň po pásce se vstupem je v $\mathcal{O}(n)$.
   - Určení indexů je při vhodném kódování konstantní (index uzlu je přímo uložen v kódu hrany), může být vhodné je ale zkopírovat na pomocné pásky, což je v $\mathcal{O}(n)$.
   - Násobení je možné realizovat v subpolynomiálním čase[4].
   - Čísla na třetí pásce se budou zvětšovat, což vyžaduje posun celého obsahu pásky doprava. To je s využitím pomocné pásky možné realizovat v $\mathcal{O}(n)$.

[2] δ značí stupeň uzlu, tedy počet sousedů

[3] Viz např. `https://cs.stackexchange.com/a/9922`.

[4] `https://en.wikipedia.org/wiki/Computational_complexity_of_mathematical_operations`

5. M_σ uschová na pomocné pásce kód $\langle k \rangle$ ze vstupu, zapíše na výstupní pásku prvních $|V|$ zakódovaných čísel ze třetí pásky a za ně okopíruje uschovaný kód $\langle k \rangle$. Obnáší pouze posuny a kopírování, tedy $\mathcal{O}(n)$.

Zachování příslušnosti

„$w \in K - COLOURING \rightarrow \sigma(w) \in L_R$“: idea důkazu popsána v posledním odstavci sekce *Idea* výše.

„$\sigma(w) \in L_R \rightarrow w \in K - COLOURING \Leftrightarrow w \notin K - COLOURING \rightarrow \sigma(w) \notin L_R$“: pokud graf není k-obarvitelný, znamená to, že při zavedení libovolné obarvovací funkce $\phi : V \rightarrow \{1, 2, \ldots, k\}$ existuje alespoň jedna dvojice uzlů (u, v), které jsou sousedné a zároveň $\phi(u) = \phi(v)$. Protože mezi uzly existuje hrana, po dokončení M_σ by ve výsledné množině odpovídala uzlům u, v čísla, která byla obě vynásobena prvočíslem přiřazeným této hraně, a tedy po provedení rozkladu podle $\sim$ uvedené výše (podle barev „zdrojových uzlů“) by v jedné třídě rozkladu určitě skončila dvojice čísel, která ve svém prvočíselném rozkladu mají stejné číslo, tedy jsou soudělná.

Důkaz příslušnosti do NP

Příslušnost $L_R \in NP$ ukážeme popisem konstrukce nedeterministického Turingova stroje M' takového, že $L(M') = L_R$ a M' pracuje v polynomiálním čase:

1. Očekávaným vstupem je řetězec ve tvaru $\langle M \rangle \# \langle k \rangle$, kde M je konečná podmnožina přirozených čísel, k je přirozené číslo a $\langle \cdot \rangle$ značí vhodné kódování, ve kterém jsou prvky M uvedeny za sebou v jistém pořadí (v podstatě je M uložena jako seznam).
2. Rozdělení množiny M na třídy je v uvedeném kódování možné charakterizovat posloupností $a_1 a_2 \ldots a_{|M|}$, kde $\forall 0 < i \leq |M| : 0 < a_i \leq k$, která značí, že prvek množiny na i-tém místě (vzhledem k užitému kódování) je součástí třídy rozkladu a_i.
3. M' ověří, zda má vstupní řetězec správný tvar (linárním průchodem, tedy v $\mathcal{O}(n)$). Pokud ne, **zamítá**.
4. M' **nedeterministicky zvolí** posloupnost $a_1 a_2 \ldots a_{|M|}$ a uloží ji na druhou pásku.
5. M' zleva postupuje seznamem prvků – kódem množiny M na první pásce a zároveň posloupností na pásce druhé.
6. Vždy si nejprve zkopíruje pořadí i a hodnotu aktuálního prvku m_i spolu s odpovídajícím a_i na třetí pásku (v $\mathcal{O}(n)$ krocích), následně prochází zbylé prvky seznamu.
7. Pro každý prvek na pořadí j zkontroluje, zda $a_i = a_j$, a pokud ano, ověří soudělnost m_i, m_j (to je možné provést v polynomiálním čase $\mathcal{O}(n^l)$ např. binárním NSD algoritmem[5]). Pokud jsou čísla soudělná, **zamítá**, jinak pokračuje dalším prvkem.
8. Pokud M' projde (ve vnější iteraci i) všechna čísla v M, **přijímá**.
9. Celkem takto pro $|M|$ prvků provede TS maximálně $|M| \cdot (|M| - 1)/2 \in \mathcal{O}(n^2)$ porovnání. V nejhorším případě pro každou dvojici prvků provede ověření soudělnosti v polynom. čase $\mathcal{O}(n^l)$, tedy celková časová složitost je také polynomiální.

Příklad 3

Jsem přesvědčen, že bych neměl problém tento příklad vypracovat, ale kvůli mému extrémnímu časovému vytížení způsobenému mj. organizací dne otevřených dveří na naší drahé fakultě jsem to v požadovaném čase bohužel nezvládl.

[5] `https://en.wikipedia.org/wiki/Binary_GCD_algorithm`