Teorie algoritmů (B4M01TAL)

1 Třídy složitosti a typy úloh

- **Třída $\mathcal{P}$:** Rozhodovací úloha U leží ve třídě $\mathcal{P}$, jestliže existuje deterministický Turingův stroj M, který rozhoduje jazyk L_U a pracuje v polynomiálním čase; tj. existuje polynom $p(n)$ takový, že časová složitost $T(n)$ stroje M je $O(p(n))$.
- **Třída $\mathcal{NP}$:** Rozhodovací úloha U leží ve třídě $\mathcal{NP}$, jestliže existuje nedeterministický Turingův stroj M, který rozhoduje jazyk L_U a pracuje v polynomiálním čase; tj. existuje polynom $p(n)$ takový, že časová složitost $T(n)$ stroje M je $O(p(n))$.
- **Třída $\mathcal{NPC}$ (NP-úplné úlohy):** Rozhodovací úloha U je NP-úplná, jestliže jsou splněny obě následující podmínky:
  1. Úloha U leží ve třídě $\mathcal{NP}$.
  2. Pro každou rozhodovací úlohu X ze třídy $\mathcal{NP}$ platí, že X se polynomiálně redukuje na U ($X \leq_p U$).
- **Třída co-$\mathcal{NP}$:** Jazyk L patří do třídy co-$\mathcal{NP}$ právě tehdy, jestliže jeho doplněk $\overline{L}$ (množina všech slov nad danou abecedou, která nepatří do L) patří do třídy $\mathcal{NP}$.
- **Třída $\mathcal{PSPACE}$:** Jazyk L patří do třídy $\mathcal{PSPACE}$, jestliže existuje deterministický Turingův stroj M, který přijímá jazyk L a pracuje s polynomiální paměťovou složitostí; tj. existuje polynom $p(n)$ takový, že paměťová složitost $S(n)$ stroje M je $O(p(n))$.
- **Třída $\mathcal{NPSPACE}$:** Jazyk L patří do třídy $\mathcal{NPSPACE}$, jestliže existuje nedeterministický Turingův stroj M, který přijímá jazyk L a pracuje s polynomiální paměťovou složitostí.
- **Třída $\mathcal{R}$ (Rekursivní jazyky):** Jazyk L je rekursivní, jestliže existuje Turingův stroj M, který rozhoduje jazyk L. To znamená, že pro každé slovo $w \in L$ se M úspěšně zastaví v koncovém stavu a pro každé slovo $w \notin L$ se M neúspěšně zastaví.
- **Třída $\mathcal{RS}$ (Rekursivně spočetné jazyky):** Jazyk L je rekursivně spočetný, jestliže existuje Turingův stroj M, který tento jazyk přijímá. To znamená, že pro každé slovo $w \in L$ se M úspěšně zastaví v koncovém stavu, avšak pro slovo $w \notin L$ se M buď zastaví neúspěšně, nebo se nezastaví vůbec.
- **Třída $\mathcal{RP}$ (Randomized Polynomial):** Jazyk L patří do třídy $\mathcal{RP}$ právě tehdy, když existuje randomizovaný Turingův stroj M takový, že:
  - Pro každé slovo $w \notin L$ se stroj M zastaví v přijímajícím stavu q_f s pravděpodobností 0.
  - Pro každé slovo $w \in L$ se stroj M zastaví v koncovém stavu q_f s pravděpodobností alespoň rovnou $1/2$.
  - Existuje polynom $p(n)$ takový, že pro každý vstup délky n trvá každý běh stroje M maximálně $p(n)$ kroků.
- **Třída co-$\mathcal{RP}$:** Jazyk L patří do třídy co-$\mathcal{RP}$ právě tehdy, když jeho doplněk patří do třídy $\mathcal{RP}$.

- **Třída $\mathcal{ZPP}$ (Zero-error Probabilistic Polynomial):** Jazyk L patří do třídy $\mathcal{ZPP}$ právě tehdy, když existuje randomizovaný Turingův stroj M typu Las-Vegas takový, že:
  - Pro každé slovo $w \notin L$ se stroj M zastaví v přijímajícím stavu q_f s pravděpodobností 0.
  - Pro každé slovo $w \in L$ se stroj M zastaví v koncovém stavu q_f s pravděpodobností 1.
  - Existuje polynom $p(n)$ takový, že střední hodnota počtu kroků stroje M v jednom běhu pro vstup délky n je nejvýše $p(n)$.
- **NP-těžká úloha (NP-hard):** Rozhodovací úloha U je NP-těžká, jestliže existuje alespoň jedna NP-úplná úloha V taková, že V se polynomiálně redukuje na U ($V \leq_p U$).
- **Silně NP-úplná úloha:** Úloha U je silně NP-úplná, jestliže existuje polynom $p(n)$ takový, že úloha U zůstává NP-úplnou i v případě, že se omezíme pouze na množinu instancí I, pro které platí, že největší číslo v instanci $num(I) \leq p(n)$, kde n je velikost instance I.
- **Pseudopolynomiální algoritmus:** Algoritmus A je pseudopolynomiální, jestliže existuje polynom $p(x, y)$ takový, že pro každou instanci I algoritmus A řeší úlohu v čase $O(p(n, num(I)))$, kde n je velikost instance I a $num(I)$ je největší číslo v dané instanci I.
- **R-aproximační algoritmus:** Polynomiální algoritmus A se nazývá R-aproximační (pro $R > 1$), jestliže pro každou instanci I optimalizační úlohy algoritmus A nalezne přípustné řešení, jehož hodnota účelové funkce není horší než R-násobek hodnoty optimálního řešení $OPT(I)$.

2 Složitost a chování Turingových strojů

- **Časová složitost Turingova stroje $T(n)$:** Parciální zobrazení z množiny přirozených čísel do sebe. Jestliže existuje alespoň jeden vstup délky n, pro který se Turingův stroj (u NTM alespoň v jedné větvi výpočtu) nezastaví, pak $T(n)$ není definováno. V opačném případě je $T(n)$ rovno maximálnímu počtu kroků do zastavení stroje, kde maximum se bere přes všechny vstupy délky n (a přes všechny větve výpočtu).
- **Paměťová složitost Turingova stroje $S(n)$:** Parciální zobrazení z množiny přirozených čísel do sebe. Jestliže existuje alespoň jeden vstup délky n, pro který stroj použije nekonečnou část pásky, pak $S(n)$ není definováno. V opačném případě je $S(n)$ rovno největšímu rozdílu pořadových čísel polí pásky, která byla během výpočtu alespoň jednou navštívena hlavou, kde maximum se bere přes všechny vstupy w délky n.
- **Přijímání jazyka nedeterministickým TS (NTM):** Slovo $w \in \Sigma^*$ je přijímáno NTM M, jestliže existuje alespoň jedna konečná posloupnost výpočetních kroků (přijímající výpočet), po kterých se stroj dostane do koncového přijímajícího stavu $q_f \in F$. Jazyk $L(M)$ je množina všech takových slov w.
- **Rozhodování jazyka nedeterministickým TS:** NTM M rozhoduje jazyk L, jestliže M přijímá jazyk L, a navíc pro každé vstupní slovo $w \in \Sigma^*$ platí, že každá možná větev výpočtu stroje M vždy skončí po konečně mnoha krocích (stroj se nesmí na žádném vstupu v žádné větvi zacyklit).
- **Randomizovaný TS pracující v polynomiálním čase:** RTM M pracuje v polynomiálním čase, jestliže existuje polynom $p(n)$ takový, že pro každé vstupní slovo w délky n a pro každou možnou posloupnost náhodných bitů na druhé pásce se stroj M zastaví po nejvýše $p(n)$ krocích.

3 Redukce a speciální jazyky

- **Redukce rozhodovacích úloh ($U \leq V$):** Rozhodovací úloha U se redukuje na rozhodovací úlohu V, jestliže existuje algoritmus M, který se na každém vstupu zastaví a který pro každou instanci I úlohy U zkonstruuje instanci I' úlohy V tak, že platí: I je ANO-instance úlohy U právě tehdy, když I' je ANO-instance úlohy V.
- **Polynomiální redukce ($U \leq_p V$):** Rozhodovací úloha U se polynomiálně redukuje na rozhodovací úlohu V, jestliže se U redukuje na V a příslušný transformační algoritmus M pracuje v polynomiálním čase vzhledem k délce instance I.
- **Redukce jazyků ($L_1 \leq L_2$):** Jsou dány jazyky $L_1 \subseteq \Sigma^*$ a $L_2 \subseteq \Gamma^*$. Jazyk L_1 se redukuje na jazyk L_2, jestliže existuje algoritmus A, který pro každé slovo $w \in \Sigma^*$ zkonstruuje slovo $A(w) \in \Gamma^*$ tak, že: $w \in L_1 \iff A(w) \in L_2$.
- **Diagonální jazyk L_d:** Jazyk L_d je množina tvořená všemi binárními slovy w takovými, že Turingův stroj, jehož kód je w (M_w), nepřijímá slovo w (tj. $w \notin L(M_w)$). Pokud w není platným kódem stroje, uvažujeme, že reprezentuje stroj přijímající prázdný jazyk.
- **Univerzální jazyk L_{UN}:** Jazyk L_{UN} je množina všech binárních slov ve tvaru $\langle M \rangle w$, kde $\langle M \rangle$ je kód reprezentující deterministický TS M, a w je binární slovo takové, že $w \in L(M)$.

4 Grafové pojmy a instance

- **Rozhodovací úloha:** Úloha definovaná jako obecná specifikace vztahu zadání a řešení, přičemž pro každou platnou instanci I je správným řešením výhradně odpověď ANO, nebo odpověď NE.
- **Barevnost grafu (k-barevnost):** Neorientovaný graf $G = (V, E)$ bez smyček je k-barevný, jestliže existuje zobrazení $b : V \to B$ (kde $|B| = k$) takové, že pro každou hranu $\{u, v\} \in E$ platí $b(u) \neq b(v)$.
- **Klika:** Podmnožina vrcholů $K \subseteq V$ v neorientovaném grafu, pro kterou platí, že každé dva různé vrcholy $u, v \in K$ jsou spojeny hranou $\{u, v\} \in E$ (indukovaný podgraf je úplný).
- **Nezávislá množina:** Množina vrcholů $N \subseteq V$, jestliže pro žádnou dvojici vrcholů $u, v \in N$ neexistuje v grafu G hrana $\{u, v\} \in E$.
- **Vrcholové pokrytí:** Podmnožina vrcholů $C \subseteq V$, jestliže pro každou hranu $e = \{u, v\} \in E$ platí, že $u \in C$ nebo $v \in C$.
- **Hamiltonovská kružnice / cyklus:** Uzavřená cesta (v neorientovaném grafu kružnice, v orientovaném cyklus), která obsahuje každý vrchol z množiny V právě jednou.
- **Hamiltonovská cesta:** Cesta, která obsahuje každý vrchol z množiny V právě jednou.
- **Metrická instance TSP:** Instance TSP tvořená n městy a vzdálenostmi $d(i, j)$, která splňuje trojúhelníkovou nerovnost: $d(i, j) \leq d(i, k) + d(k, j)$ pro všechna města i, j, k.

5 Postův korespondenční problém (PCP)

- **Definice PCP:** Jsou dány dva seznamy slov $A = (u_1, u_2, \ldots, u_k)$ a $B = (v_1, v_2, \ldots, v_k)$ nad abecedou Σ. Postův korespondenční problém hledá, zda existuje konečná posloupnost indexů $i_1, i_2, \ldots, i_m$ (kde $m \geq 1$ a $1 \leq i_j \leq k$) taková, že konkatenace slov ze seznamu A odpovídá konkatenaci slov ze seznamu B ve stejném pořadí:

$$u_{i_1} u_{i_2} \ldots u_{i_m} = v_{i_1} v_{i_2} \ldots v_{i_m}$$

- **ANO-instance PCP:** Instance tvořená dvojicí seznamů (A, B) je ANO-instancí právě tehdy, když pro ni existuje alespoň jedno řešení (tj. neprázdná posloupnost indexů splňující rovnost zřetězených slov).
- **NE-instance PCP:** Instance tvořená dvojicí seznamů (A, B) je NE-instancí právě tehdy, když pro ni neexistuje žádná neprázdná konečná posloupnost indexů splňující danou podmínku. PCP je **nerozhodnutelný problém**, což znamená, že neexistuje algoritmus, který by pro každou NE-instanci PCP v konečném čase potvrdil, že řešení neexistuje.
- **Modifikovaný PCP (MPCP):** Varianta PCP, kde je vynucen začátek řešení pomocí prvního páru slov ze seznamů ($i_1 = 1$). MPCP je rovněž nerozhodnutelný a lze jej redukovat na PCP.

6 Doplňující věty a koncepty

- **Riceova věta:** Nechť $\mathcal{S}$ je libovolná netriviální vlastnost jazyků třídy $\mathcal{RS}$ (tj. existuje aspoň jeden jazyk $L \in \mathcal{RS}$, který vlastnost $\mathcal{S}$ má, a aspoň jeden $L' \in \mathcal{RS}$, který ji nemá). Pak jazyk $L_{\mathcal{S}} = \{\langle M \rangle \mid L(M) \text{ má vlastnost } \mathcal{S}\}$ je nerozhodnutelný.
- **Cookova-Levinova věta:** Problém splnitelnosti booleovských formulí (SAT) je NP-úplný ($SAT \in \mathcal{NPC}$).
- **Redukce optimalizace na rozhodování:** Optimalizační úlohu (např. TSP) lze řešit v polynomiálním čase pomocí orákula (algoritmu) pro příslušnou rozhodovací úlohu užitím binárního vyhledávání přes rozsah možných hodnot účelové funkce.
- **Vztah tříd složitosti:** $\mathcal{P} \subseteq (\mathcal{NP} \cap co\text{-}\mathcal{NP}) \subseteq \mathcal{NP} \subseteq \mathcal{PSPACE} = \mathcal{NPSPACE} \subseteq \mathcal{R} \subseteq \mathcal{RS}$
- **Savitchova věta (detail):** Pro každou paměťovou funkci $S(n) \geq \log n$ platí $NPSPACE(S(n)) \subseteq PSPACE(S^2(n))$. Speciálně pro polynomiální omezení platí $\mathcal{PSPACE} = \mathcal{NPSPACE}$.

7 Stochastické algoritmy (Typy)

- **Monte Carlo:** Algoritmus, který pracuje v polynomiálním čase, ale může vrátit chybný výsledek s určitou (omezenou) pravděpodobností (třídy **RP**, **co-RP**, **BPP**).
- **Las Vegas:** Algoritmus, který nikdy nevrátí chybný výsledek. Jeho čas běhu je náhodná veličina s konečnou střední hodnotou (třída **ZPP**). Platí $\mathcal{ZPP} = \mathcal{RP} \cap co\text{-}\mathcal{RP}$.

8 Vztahy mezi třídami a doplňky

8.1 Obecná hierarchie inkluzí

V teorii složitosti platí následující řetězec inkluzí (vztahy, o kterých víme, že platí, i když ne u všech víme, zda jsou ostré):

$$\mathcal{P} \subseteq (\mathcal{NP} \cap co\text{-}\mathcal{NP}) \subseteq \mathcal{NP} \subseteq \mathcal{PSPACE} = \mathcal{NPSPACE} \subseteq \mathcal{R} \subseteq \mathcal{RS}$$

8.2 Doplňky tříd (co-třídy)

Doplněk třídy $\mathcal{K}$ (značíme $co\text{-}\mathcal{K}$) obsahuje jazyky, jejichž doplněk patří do $\mathcal{K}$.

- **Třídy uzavřené na doplněk:** O těchto třídách víme, že $\mathcal{K} = co\text{-}\mathcal{K}$.
  - $\mathcal{P} = co\text{-}\mathcal{P}$ (Pokud umíme v polynomiálním čase říct ANO, umíme i NE – stačí prohodit stavy).
  - $\mathcal{PSPACE} = co\text{-}\mathcal{PSPACE}$ (Plyne z Immerman-Szelepcsényiho věty).

- $\mathcal{R} = co\text{-}\mathcal{R}$ (Pokud je jazyk rozhodnutelný, je rozhodnutelný i jeho doplněk).

- **Třídy u kterých rovnost s doplňkem není známa:**
  - $\mathcal{NP}$ vs. $co\text{-}\mathcal{NP}$: Obecně se věří, že $\mathcal{NP} \neq co\text{-}\mathcal{NP}$. Pokud by se rovnaly, znamenalo by to, že pro každou NP úlohu (např. SAT) existuje krátký certifikát i pro odpověď NE.
  - $\mathcal{RS}$ vs. $co\text{-}\mathcal{RS}$: Víme bezpečně, že $\mathcal{RS} \neq co\text{-}\mathcal{RS}$ (např. jazyk $L_d \notin \mathcal{RS}$, ale $\overline{L_d} \in \mathcal{RS}$).

8.3 Věty o průniku (Klíčové vztahy)

Tyto věty definují třídy pomocí průniku třídy a jejího doplňku:

- **Vztah rekurzivních a rekurzivně spočetných jazyků:**

$$\mathcal{R} = \mathcal{RS} \cap co\text{-}\mathcal{RS}$$

  Tedy: Jazyk je rozhodnutelný právě tehdy, když on i jeho doplněk jsou přijímány Turingovým strojem (pokud se jeden zastaví pro ANO a druhý pro NE, můžeme je pustit paralelně a vždy dostaneme odpověď).

- **Vztah pravděpodobnostních tříd (ZPP):**

$$\mathcal{ZPP} = \mathcal{RP} \cap co\text{-}\mathcal{RP}$$

  Třída Zero-error Probabilistic Polynomial (Las Vegas) je přesně průnikem tříd s jednostrannou chybou.

8.4 Randomizované třídy v hierarchii

Pro randomizované algoritmy (RTM) platí tento vztah k deterministickým a nedeterministickým třídám:

$$\mathcal{P} \subseteq \mathcal{ZPP} \subseteq \mathcal{RP} \subseteq \mathcal{NP}$$

$$\mathcal{P} \subseteq \mathcal{ZPP} \subseteq co\text{-}\mathcal{RP} \subseteq co\text{-}\mathcal{NP}$$

8.5 Shrnutí vztahů k NP-úplnosti

- Pokud libovolný jazyk $L \in \mathcal{NPC}$ patří do $\mathcal{P}$, pak $\mathcal{P} = \mathcal{NP}$.
- Pokud libovolný jazyk $L \in \mathcal{NPC}$ patří do $co\text{-}\mathcal{NP}$, pak $\mathcal{NP} = co\text{-}\mathcal{NP}$.
- Žádný NP-úplný jazyk nemůže být rekurzivně nerozhodnutelný (protože z definice musí být v NP, a NP je podmnožinou R).

9 Strom polynomiálních redukcí ($\mathcal{NP}$-úplnost)

Tento strom znázorňuje posloupnost důkazů NP-úplnosti. Redukce $A \leq_p B$ znamená: „Pokud umíme efektivně řešit B, umíme efektivně řešit i A.“

Základní uzel: SAT (Problém splnitelnosti booleovských formulí)
Daná formule v konjunktivní normální formě (CNF). Otázka: Existuje ohodnocení proměnných 0/1 takové, že formule je pravdivá?
Význam: První dokázaný NP-úplný problém (Cook-Levinova věta).

9.1 Větve redukcí od SAT

SAT $\leq_p$ 3-SAT Každá klauzule s k literály se rozbije na několik klauzulí s právě 3 literály pomocí pomocných proměnných. *Příklad:* $(l_1 \vee l_2 \vee l_3 \vee l_4)$ se změní na $(l_1 \vee l_2 \vee z) \wedge (\neg z \vee l_3 \vee l_4)$.

3-SAT $\leq_p$ Klika (Clique) Pro každý literál v každé klauzuli vytvoříme vrchol. Hrany vedou mezi všemi vrcholy, které nejsou ve stejné klauzuli a zároveň si neodporují (x vs $\neg x$). Hledáme kliku velikosti k = počet klauzulí.

Klika $\leq_p$ Nezávislá množina Provede se doplňkový graf (kde hrana byla, tam ji smažeme, a naopak). Klika v původním grafu je nezávislou množinou v doplňkovém grafu.

Nezávislá množina $\leq_p$ Vrcholové pokrytí Množina S je nezávislá právě tehdy, když $V \setminus S$ je vrcholové pokrytí. Pokud existuje nezávislá množina velikosti k, existuje vrcholové pokrytí velikosti $|V| - k$.

3-SAT $\leq_p$ 3-Barevnost Vytvoří se speciální grafové komponenty (gadgety). Jeden gadget fixuje 3 barvy (Zelená/Pravda, Červená/Nepravda, Modrá/Pomocná). Další gadgety modelují klauzule tak, aby byly obarvitelné pouze tehdy, když je aspoň jeden literál "zelený".

3-SAT $\leq_p$ Součet podmnožiny Vytvoří se velká čísla v tabulce. Horní část tabulky hlídá proměnné (vybereme buď x, nebo $\neg x$), dolní část hlídá splnění klauzulí (aspoň jeden literál musí být vybrán). Cílový součet K je nastaven tak, aby vynutil správnou strukturu výběru.

Součet podmnožiny $\leq_p$ Batoh (Knapsack) Subset Sum je speciální případ batohu, kde váha předmětu je rovna jeho ceně a kapacita batohu je rovna cílovému součtu K.

Hamiltonovská kružnice $\leq_p$ TSP Graf G pro Hamiltonovskou kružnici převedeme na úplný graf s ohodnocením hran: hrany z G mají cenu 1, hrany neexistující v G mají cenu 2 (nebo $r \cdot n + 1$). Hamiltonovská kružnice existuje, pokud existuje cesta obchodního cestujícího s cenou n.

10 Slovník NP-úplných úloh

- **3-Barevnost:** Lze vrcholy grafu obarvit 3 barvami tak, aby žádní sousedé neměli stejnou barvu?
- **Klika:** Existuje v grafu podgraf o K vrcholech, kde je každý s každým spojen hranou?
- **Nezávislá množina:** Existuje v grafu K vrcholů, mezi kterými nevede ani jedna hrana?
- **Vrcholové pokrytí:** Existuje K vrcholů takových, že každá hrana v grafu má aspoň jeden konec v této množině?
- **Hamiltonovská kružnice:** Existuje cesta, která projde každým vrcholem právě jednou a vrátí se do startu?
- **Součet podmnožiny:** Lze z dané množiny čísel vybrat taková, jejichž součet je přesně K?