NP-úplnost

Tomáš Matoušek

Definice.

• *Abeceda je libovolná konečná množina. Abecedu budeme značit Σ, její prvky nazveme symboly.*

• *Slovem nad abecedou Σ nazveme konečnou posloupnost jejích symbolů, píšeme $u = a_1 a_2 \dots a_n$. Délka slova je $|u| = n$. Posloupnost, která neobsahuje žádný symbol, nazveme prázdným slovem a označíme ji λ.*

• *Množinu všech slov nad abecedou Σ označíme Σ^*.*

• *Jazyk L nad abecedou Σ je libovolná podmnožina Σ^*.*

Příklad.
abeceda ..... $\Sigma = \{0, 1\}$
symboly ..... $0, 1$
množina slov..... $\Sigma^* = \{\lambda, 0, 1, 00, 01, \dots, 1000101010011, \dots\}$
konečný jazyk ..... $L_1 = \{\lambda, 1, 01, 001\}$
nekonečný jazyk .... $L_2 = \{0, 00, 000, \dots\}$

Definice. (Neformálně)

• *Úlohou myslíme nalezení řešení (výstupu) s danými vlastnostmi pro daný vstup.*

• *Instancí úlohy myslíme její konkrétní vstup.*

• *Optimalizační úloha je taková úloha, jejímž výstupem má být optimální řešení.*

• *Rozhodovací problém je úloha, jejímž výstupem je buď „ano“, nebo „ne“.*

Každá instance úlohy a každé její řešení je zakódováno do slova v abecedě $\Sigma = \{0, 1\}$. Úloha Q je zobrazením z jazyka $L(Q) \subseteq \Sigma^*$ do $\mathcal{P}(\Sigma^*)$, které danému vstupu přiřadí nějakou množinu řešení. Je-li třeba vybrat optimální řešení, lze nadefinovat funkci w, která ohodnotí jednotlivá řešení, a optimalizační úloha je pak zobrazení vstupu na řešení s minimální, resp. maximální hodnotou w. Rozhodovací problém zobrazuje vstupy do množiny $\{0, 1\}$.

Dále se budeme zabývat pouze rozhodovacími problémy (stručněji problémy).

Definice. (Jazyk problému) *Jazyk problému je množina $L(Q) = \operatorname{dom} Q$. Lze rozdělit na dva disjunktní jazyky:*

• *jazyk, na který dává problém odpověď „ano“, značíme $L(Q)_Y = Q^{-1}(\{1\})$,*

• *jazyk, na který dává problém odpověď „ne“, značíme $L(Q)_N = Q^{-1}(\{0\})$.*

Turingovy stroje

Nechť Q je problém. Vyřešit problém Q znamená pro každý vstup $x \in L(Q)$ rozhodnout, zda $x \in L(Q)_Y$ nebo $x \in L(Q)_N$. Pro posouzení složitosti tohoto rozhodnutí pro různé problémy použijeme Turingovy stroje.

Definice. *Deterministický Turingův stroj (DTS) je pětice $(Q, \Sigma_\varepsilon, \delta, q_0, F)$, kde*

- *Q je konečná neprázdná množina stavů,*
- *Σ_ε je nějaká abeceda Σ s přidaným speciálním symbolem ε,*
- *$\delta : \Omega \subseteq (Q \setminus F) \times \Sigma_\varepsilon \to Q \times \Sigma_\varepsilon \times \{-1, 0, 1\}$ je program DTS,*
- *$q_0 \in Q$ je počáteční stav,*
- *$F \subset Q$ je množina koncových stavů.*

Definice. (Instrukce) *Program δ je množina instrukcí. Instrukce je tedy pětice $(p, x, q, y, d) \in \delta$, což se pro přehlednost zapisuje obvykle jako $(p, x) \to (q, y, d)$.*

DTS si lze představit jako stroj, který má nekonečnou pásku a hlavu nad nějakým políčkem na této pásce. Tento stroj se v každém okamžiku nachází v právě jednom ze stavů Q a přechází mezi nimi v krocích podle svých instrukcí.

Stroj tedy pracuje v krocích. Na začátku je na pásce vstupní slovo, hlava se nachází na jeho prvním symbolu a stroj je ve stavu q_0. V každém kroku stroj přečte symbol x nacházející se pod hlavou a podle instrukce $(p, x) \to (q, y, d)$, kde p je aktuální stav, přejde do stavu q, zapíše na pásku pod hlavu symbol y a posune se o d políček vpravo.

Definice. (Výpočet TS) *Výpočet TS je posloupnost kroků TS. Výpočtem nad slovem w rozumíme výpočet, jehož první krok je prvním krokem stroje po jeho startu se vstupem w.*

Definice. (Přijímání slova DTS) *Slovo w je přijímáno DTS, pokud výpočet nad w skončí v nějakém koncovém stavu.*

Definice. *Nedeterministický Turingův stroj (NTS) je pětice $(Q, \Sigma_\varepsilon, \delta, q_0, F)$, kde Q, Σ_ε, q_0, F jsou definovány stejně jako u DTS, program δ je definován jako*

$$\delta : (Q \setminus F) \times \Sigma_\varepsilon \to \mathcal{P}(Q \times \Sigma_\varepsilon \times \{-1, 0, 1\}),$$

tj. stroj může mít v libovolném kroku možnost výběru mezi několika instrukcemi.

Definice. (Přijímání slova NTS) *Slovo w je přijímáno NTS, pokud existuje výpočet nad w, který skončí v nějakém koncovém stavu.*

Definice. *Jazyk rozpoznávaný strojem T je $L(T) = \{w \in \Sigma^* \mid w \text{ je přijímáno } T\}$.*

Složitost problémů

Definice. *TS T řeší problém Q, pokud je $L(T) = L(Q)_Y$ a pokud se T pro každé slovo $w \in L(Q)$ zastaví.*

Definice. *Nechť Q je problém a T je DTS, který se zastaví na každém slově z $L(Q)$. Časová složitost programu T je funkce $t : \mathbb{N} \to \mathbb{N}$ definovaná předpisem*

$t(n) = \max\{0\} \cup \{ m \mid (\exists w \in L(Q), |w| = n)$ *výpočet T nad w skončí po m krocích*$\}$.

Existuje-li polynom p takový, že $t \leq p$, je složitost programu polynomiální.

Definice. *Řekneme, že problém Q je ve třídě P, existuje-li DTS řešící Q, jehož program má polynomální časovou složitost.*

Definice. *Nechť T je NTS. Řekneme, že T přijímá slovo $w \in \Sigma^*$ v čase $m \in \mathbb{N}$, pokud $w \in L(T)$ a nejkratší přijímající výpočet má m kroků. Časová složitost programu T je funkce $t : \mathbb{N} \to \mathbb{N}$ definovaná předpisem*

$$t(n) = \max\{0\} \cup \{ m \mid (\exists w \in \Sigma^*, |w| = n)\ T \text{přijímá } w \text{ v čase } m\}.$$

Existuje-li polynom p takový, že $t \leq p$, je složitost programu polynomiální.

Definice. *Řekneme, že problém Q je ve třídě NP, existuje-li NTS řešící Q, jehož program má polynomální časovou složitost.*

Poznámka. Pro každý NTS existuje NTS, jehož program se skládá ze dvou fází:

- Nedeterministické vygenerování slova $o \in \{0,1\}^*$ (tzv. *orákulum*).
- Deterministický běh původního programu řízený orákulem.

Definice. *Funkce $f : \Sigma^* \to \Sigma^*$ je polynomiálně vyčíslitelná, pokud existuje DTS, který každé slovo $w \in \Sigma^*$ délky n převede na slovo $f(w)$, přičemž vykoná nejvýše $p(n)$ kroků, kde p je polynom.*

Definice. *Jazyk $L_1 \subseteq \Sigma^*$ je polynomiálně převoditelný na jazyk $L_2 \subseteq \Sigma^*$ (značíme $L_1 \propto L_2$), pokud existuje polynomiálně vyčíslitelná funkce $f : \Sigma^* \to \Sigma^*$ taková, že*

$$(\forall w \in \Sigma^*)\ w \in L_1 \Leftrightarrow f(w) \in L_2.$$

Definice. *Problém Q je NP-těžký, pokud $(\forall Q' \in NP)\ L(Q')_Y \propto L(Q)_Y$.*

Definice. *Problém Q je NP-úplný, pokud je ve třídě NP a je NP-těžký.*

Důsledek. Nechť Q a Q' jsou rozhodovací problémy. Je-li problém Q NP-těžký a $L(Q)_Y \propto L(Q')_Y$, pak Q' je také NP-těžký.

NP-úplné problémy

Problém. (KACHL)

Instance problému:

• konečná množina barev B

• čtvercová síť S, strany čtverců tvořící obvod sítě jsou obarveny barvami z B

• konečná množina čtvercových kachlíků K, každý kachlík je rozdělen úhlopříčkami na čtyři trojúhelníky obarvené nějakými barvami z B

Problém:

Lze S vykachlíkovat pomocí kachlíků K (bez otáčení) tak, aby obarvení obvodu sítě odpovídalo barvám trojúhelníků přiléhajících k obvodu a aby každé dva sousední trojúhelníky měly stejnou barvu?

Problém. (SAT)

Instance problému: Formule VL v konjunktivním normálním tvaru, tj. formule tvaru

$$\varphi(x_1, \ldots, x_n) = \bigwedge_{i=1}^{m} \bigvee_{j=1}^{k_i} a_{ij},$$

kde $m, k_i \in \mathbb{N}$ a a_{ij} jsou výrokové proměnné $x_1, \ldots, x_n$ nebo jejich negace.

Problém: Je φ splnitelná?

Problém. (3-COLOR)

Instance problému: Graf G.

Problém: Jsou vrcholy G obarvitelné třemi barvami, aby nebyly spojeny hranou vrcholy stejné barvy?

Problém. (KLIKA)

Instance problému: Graf G, přirozené číslo k.

Problém: Existuje v G klika velikosti aspoň k, tj. úplný podgraf na aspoň k vrcholech?

Problém. (HK)

Instance problému: Graf G.

Problém: Existuje v G hamiltonovská kružnice?

Problém. (TSP)

Instance problému: Úplný graf G s hranami ohodnocenými přirozenými čísly, $k \in \mathbb{N}$.

Problém: Existuje v G hamiltonovská kružnice váhy nejvýše k?