Základy složitosti a vyčíslitelnosti

NTIN090

Petr Kučera

2022/23 (5. přednáška)

Základní třídy složitosti

Rozhodovací problémy

V rozhodovacím problému se ptáme, zda daná instance x splňuje danou podmínku.

- Odpověď je typu `ano`/`ne`.
- Rozhodovací problém formalizujeme jako jazyk $L \in \Sigma^*$ kladných instancí a otázku, zda $x \in L$.
- Příklady rozhodovacích problémů:
    - Je daný graf souvislý?
    - Má daná logická formule model?
    - Má daný lineární program přípustné řešení.
    - Je dané číslo prvočíslem?

Úlohy

V úloze pro danou instanci x hledáme y, které splňuje určitou podmínku

- Odpovědí je zde y nebo informace o tom, že žádné vhodné y neexistuje
- Úlohu formalizujeme jako relaci $R \subseteq \Sigma^* \times \Sigma^*$
  - K dané instanci x hledáme y tak, že $(x, y) \in R$
- Příklady úloh:
  - Nalezení silně souvislých komponent orientovaného grafu
  - Nalezení splňujícího ohodnocení logické formule
  - Nalezení přípustného řešení lineárního programu

Časová a prostorová složitost Turingova stroje

Definice

Nechť M je (deterministický) Turingův stroj a nechť $f : \mathbb{N} \to \mathbb{N}$ je funkce, která je definovaná pro každý vstup.

- M pracuje v čase $f(n)$, pokud výpočet M nad libovolným vstupem x délky $|x| = n$ skončí po provedení nejvýše $f(n)$ kroků.
- M pracuje v prostoru $f(n)$, pokud výpočet M nad libovolným vstupem x délky $|x| = n$ skončí a využije nejvýš $f(n)$ buněk pracovní pásky.

Základní deterministické třídy složitosti

Definice

Nechť $f : \mathbb{N} \to \mathbb{N}$ je funkce, potom definujeme třídy:

$\mathrm{TIME}(f(n))$ jazyky přijímané Turingovými stroji, které pracují v čase $O(f(n))$

$\mathrm{SPACE}(f(n))$ jazyky přijímané Turingovými stroji, které pracují v prostoru $O(f(n))$

$\mathrm{TIME}(f(n)) \subseteq \mathrm{SPACE}(f(n))$ pro každou funkci $f : \mathbb{N} \to \mathbb{N}$.

- V jednom kroku pohne Turingův stroj hlavou jen o jednu buňku vpravo nebo vlevo
- Stroj použije v každém kroku nejvýš jednu buňku navíc

Význačné deterministické třídy složitosti

- Třída problémů řešitelných v polynomiálním čase

$$\mathrm{P} = \bigcup_{k \in \mathbb{N}} \mathrm{TIME}(n^k)$$

- Třída problémů řešitelných v polynomiálním prostoru

$$\mathrm{PSPACE} = \bigcup_{k \in \mathbb{N}} \mathrm{SPACE}(n^k).$$

- Třída problémů řešitelných v exponenciálním čase

$$\mathrm{EXPTIME} = \bigcup_{k \in \mathbb{N}} \mathrm{TIME}(2^{n^k}).$$

Proč polynomy?

Silnější verze Churchovy-Turingovy teze

Reálné výpočetní modely lze simulovat na Turingovu stroji s polynomiálním zpomalením/nárůstem prostoru.

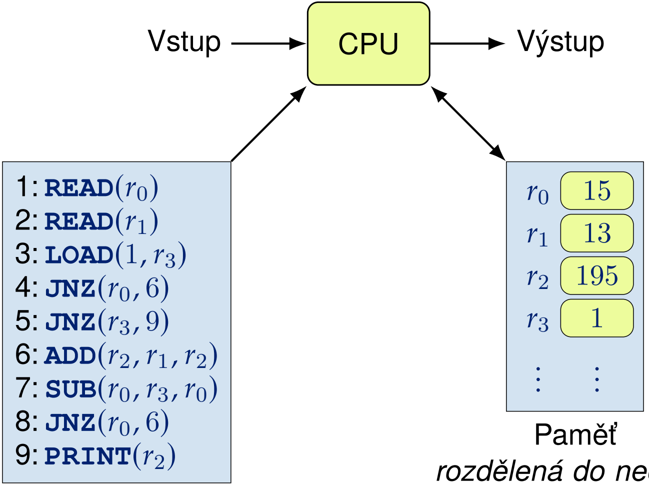
- Polynomy jsou uzavřeny na skládání.
- Polynomy (obvykle) nerostou příliš rychle.
- Definice třídy P nezávisí na zvoleném výpočetním modelu.

Cobhamova-Edmondsova teze, 1965

P odpovídá třídě prakticky řešitelných problémů na počítači.

Složitost na RAMu

Random Access Machine (RAM)



Program

Paměť
rozdělená do neomezeného počtu registrů

Cena za vykonání instrukce

- Funkce $l(n)$ určuje cenu uložení čísla n v registru

Instrukce	Efekt	Cena
$\mathbf{LOAD}(C, r_i)$	$r_i \leftarrow C$	1
$\mathbf{ADD}(r_i, r_j, r_k)$	$r_k \leftarrow [r_i] + [r_j]$	$l([r_i]) + l([r_j])$
$\mathbf{SUB}(r_i, r_j, r_k)$	$r_k \leftarrow \max([r_i] - [r_j], 0)$	$l([r_i]) + l([r_j])$
$\mathbf{COPY}([r_p], r_d)$	$r_d \leftarrow [\![r_p]\!]$	$l([r_p]) + l([\![r_p]\!])$
$\mathbf{COPY}(r_s, [r_d])$	$r_{[r_d]} \leftarrow [r_s]$	$l([r_d]) + l([r_s])$
$\mathbf{JNZ}(r_i, I_z)$	$\mathbf{if}\ [r_i] > 0\ \mathbf{then\ goto}\ z$	$l([r_i])$
$\mathbf{READ}(r_i)$	$r_i \leftarrow$ input	l(input)
$\mathbf{PRINT}(r_i)$	output $\leftarrow [r_i]$	$l([r_i])$

Čas vykonání programu RAM

Přirozené volby funkce $l(n)$ ceny uložení čísla

Konstantní Každá instrukce je vykonána v konstantním čase

$$l(n) = 1$$

Logaritmická Počet bitů potřebných k reprezentaci hodnoty n

$$l(n) = \begin{cases} \lceil \log_2(n+1) \rceil & n \geq 2 \\ 1 & n < 2 \end{cases}$$

Definice

Nechť $f : \mathbb{N} \to \mathbb{N}$ je funkce.

- RAM R **pracuje v čase $f(n)$**, pokud pro každý vstup x s celkovou cenou n je součet cen vykonaných instrukcí nejvýš $f(n)$.

Složitost RAMu a Turingovy stroje

Věta (Cook and Reckhow, 1973)

Nechť L je jazyk rozhodnutelný RAMem R v čase $f(n)$. Pak L je rozhodnutelný vícepáskovým TS, který pracuje v čase

- $O(f^2(n))$, *je-li $l(n)$ logaritmická*
- $O(f^3(n))$, *je-li $l(n)$ konstantní*

Lemma

Je-li L rozhodnutelný vícepáskovým TS v čase $f(n)$, pak je rozhodnutelný jednopáskovým TS v čase $O(f^2(n))$.

Důsledek

P je třídou jazyků, které lze rozhodnout RAMem, který pracuje v polynomiálním čase.

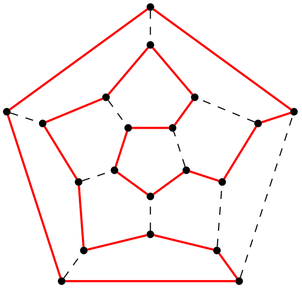
Problémy ověřitelné v polynomiálním čase

Hamiltonovská kružnice

Hamiltonovská kružnice (HK)

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

Otázka: Obsahuje G cyklus, který prochází všemi vrcholy?



- Lze navštívit každé město na mapě právě jednou a vrátit se domů?
- Jednoduše ověřitelné zda posloupnost vrcholů určuje hamiltonovskou kružnici
- Obtížné zjistit, zda daný graf obsahuje hamiltonovskou kružnici

Verifikátor čili ověřovatel

Verifikátorem pro jazyk A je algoritmus V, pro který platí, že

$$A = \{x \mid (\exists y)[V(x,y) \text{ přijme}]\}$$

Časovou složitost verifikátoru měříme pouze vzhledem k $|x|$

- Polynomiální verifikátor pracuje v čase $O(|x|^k)$ pro nějaké $k \in \mathbb{N}$
- Řetězec y zveme také certifikátem x
- Pokud $V(x,y)$ přijme, přečte jen prefix y polynomiální délky
- Stačí uvažovat polynomiální certifikáty y, jejichž délka je polynomiální v $|x|$

Verifikátor pro hamiltonovskou kružnici

Hamiltonovská kružnice (HK)

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

Otázka: Obsahuje G cyklus, který prochází všemi vrcholy?

Verifikátor V pro hamiltonovskou kružnici

Vstup: Graf $G = (V, E)$ a posloupnost vrcholů $\mathbf{v} = (v_1, \ldots, v_\ell)$

```
if ℓ ≠ |V| then odmítni
if v_i = v_j pro nějakou dvojici indexů i ≠ j then
    odmítni
for i = 1 to n − 1 do
    if {v_i, v_{i+1}} ∉ E then odmítni
if {v_n, v_1} ∉ E then odmítni
přijmi
```

Třída NP

Definice

NP je třídou jazyků, které mají polynomiální verifikátory.

- Odpovídá třídě úloh, u nichž jsme schopni v polynomiálním čase ověřit, zda daný řetězec je řešením
- NP je třída jazyků, přijímaných nedeterministickými Turingovými stroji v polynomiálním čase
- Nedeterminismus zde odpovídá *hádání* správného certifikátu y pro vstup x

P vs. NP

P lze rychle rozhodnout, zda do dané slovo patří do jazyka

NP lze rychle ověřit, že dané slovo patří do jazyka

Triviálně $\mathrm{P} \subseteq \mathrm{NP}$

Otázka, zda $\mathrm{P} = \mathrm{NP}$ je otevřená.

Nedeterministický Turingův stroj

Nedeterministický Turingův stroj (NTS) je pětice $M = (Q, \Sigma, \delta, q_0, F)$

- Q, Σ, q_0, F mají týž význam jako u *obyčejného* deterministického Turingova stroje (DTS)
- Rozdíl oproti DTS je v přechodové funkci, nyní

$$\delta : Q \times \Sigma \mapsto \mathcal{P}(Q \times \Sigma \times \{L, N, R\})$$

- Možné pohledy
  - NTS M v každém kroku *uhodne* nebo *vybere* správnou instrukci
  - NTS M vykonává všechny možné instrukce současně a nachází se během výpočtu ve více konfiguracích současně

Nejde o realistický výpočetní model

Jazyk přijímaný NTS

Předpokládejme NTS M a vstup x

Výpočet M nad x je posloupnost konfigurací $C_0, C_1, C_2, \ldots$, kde

- C_0 je počáteční konfigurace se vstupem x a
- z C_i do C_{i+1} lze přejít pomocí přechodové funkce δ

Přijímající výpočet končí konfigurací v příjímajícím stavu

M přijme slovo x pokud existuje přijímající výpočet M nad x

$L(M)$ jazyk slov přijímaných M

Časová a prostorová složitost NTS

Definice

Nechť M je nedeterministický Turingův stroj a nechť $f : \mathbb{N} \mapsto \mathbb{N}$ je funkce.

- M pracuje v čase $f(n)$, pokud každý výpočet M nad libovolným vstupem x délky $|x| = n$ skončí po provedení nejvýše $f(n)$ kroků
- M pracuje v prostoru $f(n)$, pokud každý výpočet M nad libovolným vstupem x délky $|x| = n$ skončí a využije nejvýše $f(n)$ buněk pracovní pásky

Základní nedeterministické třídy složitosti

Definice

Nechť $f : \mathbb{N} \mapsto \mathbb{N}$ je funkce, potom definujeme třídy:

$\mathrm{NTIME}(f(n))$ jazyky přijímané nedeterministickými TS, které pracují v čase $O(f(n))$

$\mathrm{NSPACE}(f(n))$ jazyky přijímané nedeterministickými TS, které pracují v prostoru $O(f(n))$

Proposition

Pro každou funkci $f : \mathbb{N} \mapsto \mathbb{N}$ *platí*

$$\mathrm{TIME}(f(n)) \subseteq \mathrm{NTIME}(f(n))$$
$$\mathrm{SPACE}(f(n)) \subseteq \mathrm{NSPACE}(f(n))$$

NP = nedeterministicky polynomiální

Věta (Alternativní definice třídy NP)

$$\mathrm{NP} = \bigcup_{k \in \mathbb{N}} \mathrm{NTIME}(n^k)$$

Důkaz.

Ve dvou krocích

1. $\mathrm{NP} \subseteq \bigcup_{k \in \mathbb{N}} \mathrm{NTIME}(n^k)$
2. $\bigcup_{k \in \mathbb{N}} \mathrm{NTIME}(n^k) \subseteq \mathrm{NP}$

□

Důkaz $\mathrm{NP} \subseteq \bigcup_{k\in\mathbb{N}} \mathrm{NTIME}(n^k)$

- Předpokládejme, že máme polynomiální verifikátor $V(x, y)$ pro L
- Popíšeme NTS M přijímající L v polynomiálním čase

Výpočet M se vstupem x

---

1 Simuluj verifikátor V
2 Nedeterministicky zvol znaky y, když je V potřebuje přečíst
3 **if** V přijme **then přijmi else odmítni**

$$\begin{aligned} x \in L &\iff (\exists y)[|y| \le p(|x|) \wedge V(x,y) \text{ přijme}] \\ &\iff \text{nějaký výpočet } M(x) \text{ přijme} \\ &\iff x \in L(M) \end{aligned}$$

M přijímá L v polynomiálním čase

Důkaz $\bigcup_{k\in\mathbb{N}} \mathrm{NTIME}(n^k) \subseteq \mathrm{NP}$

- Nechť L je přijímán nějakým NTS $M = (Q, \Sigma, \delta, q_0, F)$
- Předpokládejme, že M pracuje v polynomiálním čase $p(n)$
- Označme $r = \max_{q\in Q, a\in\Sigma} |\delta(q, a)|$
  - maximální počet větvení přechodu dle δ
  - $r \leq |Q| \cdot |\Sigma| \cdot 3$
  - r je konstanta, která závisí jen na M a nezávisí na vstupu
- Výpočet M se vstupem x lze popsat řetězcem

$$y \in \{1, \ldots, r\}^{p(|x|)}$$

- y_i — větev zvolená v kroku i, $i = 1, \ldots, p(|x|)$

Řetězec y je certifikátem toho, že M přijímá x.

Důkaz $\bigcup_{k\in\mathbb{N}} \mathrm{NTIME}(n^k) \subseteq \mathrm{NP}$ (dokončení)

Výpočet polynomiálního verifikátoru $V(x, y)$ pro jazyk L

Simuluj $M(x)$ s větvením podle y
if větev výpočtu $M(x)$ popsaná y přijme **then**
  přijmi
else
  odmítni

$$\begin{aligned} x \in L &\iff \text{nějaký výpočet } M(x) \text{ přijme} \\ &\iff (\exists y)[|y| \leq p(|x|) \text{ a simulace } M(x) \\ &\qquad\qquad \text{s větvením podle } y \text{ přijme}] \\ &\iff (\exists y)[|y| \leq p(|x|) \text{ a } V(x, y) \text{ přijme}] \end{aligned}$$

V je polynomiální verifikátor pro L, tedy $L \in \mathrm{NP}$.

Základní vztahy

Nedeterministický čas a deterministický prostor

Věta

Pro každou funkci $f : \mathbb{N} \mapsto \mathbb{N}$ platí, že

$$\mathrm{NTIME}(f(n)) \subseteq \mathrm{SPACE}(f(n))$$

- Nechť L je přijímán NTS M v čase $g(n) = O(f(n))$
- Předpokládejme, že $M = (Q, \Sigma, \delta, q_0, F)$
- Označme $r = \max_{q \in Q, a \in \Sigma} |\delta(q, a)|$
  - maximální počet větvení přechodu dle δ
  - $r \leq |Q| \cdot |\Sigma| \cdot 3$ je konstanta, která závisí jen na M
- Výpočet M se vstupem x lze popsat řetězcem $y \in \{1, \ldots, r\}^{g(|x|)}$
- y_i — větev zvolená v kroku i, $i = 1, \ldots, g(|x|)$
- Popíšeme TS M', který rozhoduje L v prostoru $O(f(n))$

Důkaz (část 2)

První nápad

Výpočet M' se vstupem x

forall $y \in \{1, \ldots, r\}^{g(|x|)}$ **do**
    Simuluj $M(x)$ s větvením podle y
    if větev výpočtu $M(x)$ popsaná y přijme **then**
        přijmi
odmítni

M' by musel znát hodnotu $g(|x|)$.

Co když $g(|x|)$ není vyčíslitelná v prostoru $O(g(|x|))$?

Důkaz (část 3)

Výpočet M' se vstupem x

```
k ← 1
repeat
    forall y ∈ {1, ..., r}^k do
        Simuluj M(x) s větvením podle y
        if větev výpočtu M(x) popsaná y přijme then přijmi
    k ← k + 1
until všechny simulace M(x) větvící podle y odmítly
odmítni
```

- M pracuje v čase $g(n) = O(f(n))$

$\implies$ každý výpočet $M(x)$ skončí do $g(n)$ kroků

$\implies$ $M'(x)$ skončí výpočet s $k \leq g(|x|)$

Důkaz (část 4)

Prostorové nároky

- Vždy platí $k \leq g(|x|)$
- Prostor $O(g(n)) = O(f(n))$ je potřeba pro uložení y
- Prostor $O(g(n)) = O(f(n))$ je potřeba k simulaci $M(x)$ podle y
- Dohromady M' pracuje v prostoru $O(f(n))$

Důkaz (část 5)

$$L = L(M')$$

- Pokud $M(x)$ má přijímající výpočet
  - Simulace $M(x)$ přijme s nějakým y, $|y| \leq g(|x|)$
  - $M'(x)$ přijme
- Pokud $M(x)$ nemá přijímající výpočet
  - Pro hodnotu $k = g(|x|)$, simulace $M(x)$ s každým y odmítne
  - $M'(x)$ odmítne

M' rozhoduje L v prostoru $O(f(n))$

NP and PSPACE

Věta

Pro každou funkci $f : \mathbb{N} \mapsto \mathbb{N}$ *platí, že*

$$\mathrm{NTIME}(f(n)) \subseteq \mathrm{SPACE}(f(n))$$

Důsledek

$\mathrm{NP} \subseteq \mathrm{PSPACE}$

Model TS s menším než lineárním prostorem

Pro prostor menší než lineární uvažujeme vícepáskový TS:

Vstupní páska je jen pro čtení

Pracovní pásky jsou pro čtení i zápis

Výstupní páska je jen pro zápis a hlava se hýbe jen vpravo

- Do prostoru se počítá jen obsah pracovních pásek
- Součástí konfigurace je
  - stav
  - poloha hlavy na vstupní pásce
  - polohy hlav na pracovních páskách
  - obsah pracovních pásek
- Konfigurace neobsahuje vstupní slovo

Další prostorové třídy

- Třída problémů řešitelných v logaritmickém prostoru

$$\mathrm{L} = \mathrm{SPACE}(\log_2 n)$$

- Třída jazyků přijímaných NTS v logaritmickém prostoru

$$\mathrm{NL} = \mathrm{NSPACE}(\log_2 n)$$

- Třída jazyků přijímaných NTS v polynomiálním prostoru

$$\mathrm{NPSPACE} = \bigcup_{k \in \mathbb{N}} \mathrm{NSPACE}(n^k)$$

Vztah prostoru a času

Věta (Vztah prostoru a času)

Pro každou funkci $f(n) \geq \log_2 n$ a každý jazyk L platí, že

$$L \in \mathrm{NSPACE}(f(n)) \implies (\exists c_L \in \mathbb{N}) \left[L \in \mathrm{TIME}(2^{c_L f(n)}) \right].$$

Důsledek

Je-li $f(n)$ funkce, pro kterou platí $f(n) \geq \log_2 n$ a je-li $g(n)$ funkce, pro kterou platí $f(n) = o(g(n))$, pak

$$\mathrm{NSPACE}(f(n)) \subseteq \mathrm{TIME}(2^{g(n)}).$$

Vztah prostoru a času

Věta (Vztah prostoru a času)

Pro každou funkci $f(n) \geq \log_2 n$ a každý jazyk L platí, že

$$L \in \mathrm{NSPACE}(f(n)) \implies (\exists c_L \in \mathbb{N}) \left[L \in \mathrm{TIME}(2^{c_L f(n)}) \right].$$

Idea důkazu

- L je přijímán nějakým NTS M v prostoru $O(f(n))$
- Pro vstup x, definujeme graf konfigurací $G_{M,x}$

  Vrcholy možné konfigurace výpočtu $M(x)$

  Hrany možné přechody mezi konfiguracemi
- Popíšeme TS M', který rozhoduje L
  - M' se vstupem x hledá v $G_{M,x}$ cestu z počáteční konfigurace do nějaké přijímající konfigurace

Graf konfigurací

M NTS pracující v prostoru $f(n)$
x vstupní řetězec
C_0^x počáteční konfigurace výpočtu $M(x)$
C_F přijímající konfigurace M
- *bez újmy na obecnosti je jediná*

Graf konfigurací výpočtů M nad x

Orientovaný graf $G_{M,x} = (V, E)$, kde
- Vrcholy V reprezentují možné konfigurace výpočtů $M(x)$
- $(C_1, C_2) \in E$ je-li možné z C_1 do C_2 přejít přechodem dle δ

$$M(x) \iff G_{M,x} \text{ obsahuje cestu z } C_0^x \text{ do } C_F$$

Velikost grafu konfigurací

Lemma

- *Uvažme funkci* $f(n) \geq \log_2 n$
- *Uvažme NTS* $M = (Q, \Sigma, \delta, q_0, F)$*, který pracuje v prostoru* $f(n)$
- *Nechť* x *je vstup délky* $n = |x|$
- *Nechť* $G_{M,x} = (V, E)$ *je odpovídající graf konfigurací*

Pak $|V| \leq 2^{c_M f(n)}$ *pro nějakou konstantu* c_M

Předpoklad

M má jednu vstupní pásku jen pro čtení a jednu pracovní pásku.

Velikost $G_{M,x}$ (důkaz)

$$|V| \leq \overbrace{|Q|}^{\text{stav}} \cdot \overbrace{n}^{\substack{\text{hlava na}\\\text{vstupní pásce}}} \cdot \overbrace{f(n)}^{\substack{\text{hlava na}\\\text{pracovní pásce}}} \cdot \overbrace{|\Sigma|^{f(n)}}^{\substack{\text{slovo na}\\\text{pracovní pásce}}}$$

Konfigurace se skládá z následujících prvků

- stav
  - $|Q|$ různých stavů
- poloha hlavy na vstupní pásce
  - n možných poloh
  - včetně prázdných políček kolem vstupu
- poloha hlavy na pracovní pásce
  - $f(n)$ možných poloh
  - je-li označený první a poslední znak vstupu
- obsah pracovní pásky
  - slovo $w \in \Sigma^*$ na pásce má délku $|w| \leq f(n)$
  - $|\Sigma|^{f(n)}$ různých slov

Počet konfigurací

$$
\begin{aligned}
|V| &\leq |Q| \cdot n \cdot f(n) \cdot |\Sigma|^{f(n)} \\
&= 2^{\log_2 |Q|} \cdot 2^{\log_2 n} \cdot 2^{\log_2 f(n)} \cdot 2^{f(n) \log_2 |\Sigma|} \\
&= 2^{\log_2 |Q| + \log_2 n + \log_2 f(n) + f(n) \log_2 |\Sigma|} \\
&\leq 2^{f(n)(\log_2 |Q| + 1 + 1 + \log_2 |\Sigma|)} && (f(n) \geq \log_2 n)
\end{aligned}
$$

Položíme-li $c_M = \log_2 |Q| + 1 + 1 + \log_2 |\Sigma|$, pak

$$
\begin{aligned}
|V| &\leq 2^{c_M f(n)} \\
|E| &\leq |V|^2 \leq 2^{2 c_M f(n)}
\end{aligned}
$$

Vztah prostoru a času

Věta (Vztah prostoru a času)

Pro každou funkci $f(n) \geq \log_2 n$ *a každý jazyk* L *platí, že*

$$L \in \mathrm{NSPACE}(f(n)) \implies (\exists c_L \in \mathbb{N}) \left[L \in \mathrm{TIME}(2^{c_L f(n)}) \right].$$

Idea důkazu

- Předpokládejme, že L je přijímán nějakým M v prostoru $O(f(n))$
- Pro vstup x, definujeme graf konfigurací $G_{M,x}$

  Vrcholy možné konfigurace výpočtu $M(x)$

  Hrany možné přechody mezi konfiguracemi
- Popíšeme TS M', který rozhoduje L
  - M' se vstupem x hledá v $G_{M,x}$ cestu z počáteční konfigurace do nějaké přijímající konfigurace

Vztah prostoru a času (důkaz)

Výpočet M' se vstupem x

Sestav počáteční konfiguraci C_0^x výpočtu $M(x)$
Projdi graf $G_{M,x}$ prohledáváním do hloubky (DFS) počínaje ve vrcholu C_0^x
if DFS nalezne přijímající konfiguraci **then**
  přijmi
odmítni

- DFS nepotřebuje mít před spuštěním zkonstruovaný graf $G_{M,x}$
  - zná počáteční vrchol C_0^x
  - seznam sousedů $\Gamma(C)$ aktuálního vrcholu C lze zkonstruovat v každém kroku pomocí přechodové funkce δ stroje M

M' nepotřebuje znát hodnotu $f(|x|)$.

Vztah prostoru a času (důkaz)

- DFS lze provést v polynomiálním čase na RAMu i TS
- Turingův stroj M' pracuje v polynomiálním čase vzhledem k velikosti grafu $G_{M,x}$
- Pracuje v čase

$$O(|V|^k) = O(2^{kc_M f(n)}) = O(2^{c_L f(n)})$$

pro nějaké konstanty $k \geq 1$ a $c_L \geq kc_M$

Vztah prostoru a čase (poznámky)

Věta (Vztah prostoru a času)

Pro každou funkci $f(n) \geq \log_2 n$ a každý jazyk L platí, že

$$L \in \mathrm{NSPACE}(f(n)) \implies (\exists c_L \in \mathbb{N}) \left[L \in \mathrm{TIME}(2^{c_L f(n)}) \right].$$

- c_L závisí na jazyku L
- Různé jazyky mají různé konstanty

Z věty NEPLYNE

- $\mathrm{NSPACE}(f(n)) \subseteq \mathrm{TIME}(2^{f(n)})$
- $\mathrm{NSPACE}(f(n)) \subseteq \mathrm{TIME}(2^{c f(n)})$ pro nějakou konstantu c

$\mathrm{NL} \subseteq \mathrm{P}$

$\mathrm{NL} = \mathrm{NSPACE}(\log_2 n)$

$\mathrm{P} = \bigcup_{k \in \mathbb{N}} \mathrm{TIME}(n^k)$

Důsledek

$\mathrm{NL} \subseteq \mathrm{P}$

- Předpokládejme, že $L \in \mathrm{NSPACE}(\log_2 n)$
- Podle věty o vztahu prostoru a času existuje konstanta $c_L \in \mathbb{N}$, pro kterou platí

$$L \in \mathrm{TIME}(2^{c_L \log_2 n}) = \mathrm{TIME}(n^{c_L}) \subseteq \mathrm{P}$$

NPSPACE $\subseteq$ EXPTIME

$$\text{NPSPACE} = \bigcup_{k \in \mathbb{N}} \text{NSPACE}(n^k)$$

$$\text{EXPTIME} = \bigcup_{k \in \mathbb{N}} \text{TIME}(2^{n^k})$$

Důsledek

NPSPACE $\subseteq$ EXPTIME

- Předpokládejme, že $L \in \text{NPSPACE}$

$\Longrightarrow$ $L \in \text{NSPACE}(n^k)$ pro nějaké $k \in \mathbb{N}$

- Podle věty o vztahu prostoru a času existuje konstanta $c_L \in \mathbb{N}$, pro kterou platí

$$L \in \text{TIME}(2^{c_L n^k}) \subseteq \text{TIME}(2^{n^{k+1}}) \subseteq \text{EXPTIME}$$

Vztahy mezi třídami

Věta

Platí následující řetězec inkluzí:

$$\mathrm{L} \subseteq \mathrm{NL} \subseteq \mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE} \subseteq \mathrm{NPSPACE} \subseteq \mathrm{EXPTIME}$$

- $\mathrm{L} \subseteq \mathrm{NL}$, $\mathrm{P} \subseteq \mathrm{NP}$, $\mathrm{PSPACE} \subseteq \mathrm{NPSPACE}$ plyne z
  - Pro každou funkci $f(n)$
    - $\mathrm{TIME}(f(n)) \subseteq \mathrm{NTIME}(f(n))$
    - $\mathrm{SPACE}(f(n)) \subseteq \mathrm{NSPACE}(f(n))$
- $\mathrm{NP} \subseteq \mathrm{PSPACE}$ plyne z
  - $\mathrm{NTIME}(f(n)) \subseteq \mathrm{SPACE}(f(n))$ pro každou funkci $f(n)$
- $\mathrm{NL} \subseteq \mathrm{P}$, $\mathrm{NPSPACE} \subseteq \mathrm{EXPTIME}$ plyne ze vztahu prostoru a času

Vztahy mezi třídami

Věta

Platí následující řetězec inkluzí:

$$\mathrm{L} \subseteq \mathrm{NL} \subseteq \mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE} \subseteq \mathrm{NPSPACE} \subseteq \mathrm{EXPTIME}$$

- $\mathrm{L} \subseteq \mathrm{NL}$, $\mathrm{P} \subseteq \mathrm{NP}$, $\mathrm{PSPACE} \subseteq \mathrm{NPSPACE}$ plyne z
  - Pro každou funkci $f(n)$
    - $\mathrm{TIME}(f(n)) \subseteq \mathrm{NTIME}(f(n))$
    - $\mathrm{SPACE}(f(n)) \subseteq \mathrm{NSPACE}(f(n))$
- $\mathrm{NP} \subseteq \mathrm{PSPACE}$ plyne z
  - $\mathrm{NTIME}(f(n)) \subseteq \mathrm{SPACE}(f(n))$ pro každou funkci $f(n)$
- $\mathrm{NL} \subseteq \mathrm{P}$, $\mathrm{NPSPACE} \subseteq \mathrm{EXPTIME}$ plyne ze vztahu prostoru a času

Vztahy mezi třídami

Věta

Platí následující řetězec inkluzí:

$$\mathrm{L} \subseteq \mathrm{NL} \subseteq \mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE} \subseteq \mathrm{NPSPACE} \subseteq \mathrm{EXPTIME}$$

- $\mathrm{L} \subseteq \mathrm{NL}$, $\mathrm{P} \subseteq \mathrm{NP}$, $\mathrm{PSPACE} \subseteq \mathrm{NPSPACE}$ plyne z
  - Pro každou funkci $f(n)$
    - $\mathrm{TIME}(f(n)) \subseteq \mathrm{NTIME}(f(n))$
    - $\mathrm{SPACE}(f(n)) \subseteq \mathrm{NSPACE}(f(n))$
- $\mathrm{NP} \subseteq \mathrm{PSPACE}$ plyne z
  - $\mathrm{NTIME}(f(n)) \subseteq \mathrm{SPACE}(f(n))$ pro každou funkci $f(n)$
- $\mathrm{NL} \subseteq \mathrm{P}$, $\mathrm{NPSPACE} \subseteq \mathrm{EXPTIME}$ plyne ze vztahu prostoru a času

Vztahy mezi třídami

Věta

Platí následující řetězec inkluzí:

$$\mathrm{L} \subseteq \mathrm{NL} \subseteq \mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE} \subseteq \mathrm{NPSPACE} \subseteq \mathrm{EXPTIME}$$

- $\mathrm{L} \subseteq \mathrm{NL}$, $\mathrm{P} \subseteq \mathrm{NP}$, $\mathrm{PSPACE} \subseteq \mathrm{NPSPACE}$ plyne z
  - Pro každou funkci $f(n)$
    - $\mathrm{TIME}(f(n)) \subseteq \mathrm{NTIME}(f(n))$
    - $\mathrm{SPACE}(f(n)) \subseteq \mathrm{NSPACE}(f(n))$
- $\mathrm{NP} \subseteq \mathrm{PSPACE}$ plyne z
  - $\mathrm{NTIME}(f(n)) \subseteq \mathrm{SPACE}(f(n))$ pro každou funkci $f(n)$
- $\mathrm{NL} \subseteq \mathrm{P}$, $\mathrm{NPSPACE} \subseteq \mathrm{EXPTIME}$ plyne ze vztahu prostoru a času