Základy složitosti a vyčíslitelnosti

NTIN090

Petr Kučera

2021/22 (7. přednáška)

Savičova věta

Savičova věta

Věta (Savičova věta)

Pro každou funkci $f(n) \geq \log_2 n$ *platí*

$$\mathrm{NSPACE}(f(n)) \subseteq \mathrm{SPACE}(f^2(n))$$

Důsledek

$$\mathrm{PSPACE} = \mathrm{NPSPACE}$$

Savičova věta (začátek důkazu)

- Předpokládejme, že $L \in \mathrm{NSPACE}(f(n))$
- Existuje NTS M, který přijímá L v prostoru $O(f(n))$
- Popíšeme deterministický TS M', který rozhoduje L v prostoru $O(f^2(n))$

Zjednodušující předpoklad: M pracuje v prostoru $f(n)$

- Pokud M pracuje v prostoru $g(n) = O(f(n))$, pak

$$O(g^2(n)) = O(f^2(n))$$

- Tedy $\mathrm{SPACE}(g^2(n)) \subseteq \mathrm{SPACE}(f^2(n))$

Technické předpoklady

- M nepohne hlavou na pracovní pásce nalevo od počáteční pozice
- M má jednoznačnou přijímající konfiguraci C_{acc}
  - Jediný přijímající stav q_1
  - Hlava na vstupní pásce je nad nejlevějším symbolem vstupu
  - Hlava na pracovní pásce je nad nejlevější buňkou omezeného prostoru
  - Pracovní páska je prázdná
- C_0^x označuje počáteční konfiguraci výpočtů M se vstupem x
- $n = |x|$ označuje délku vstupu x

Prohledáváme graf

Idea: Se vstupem x, hledej cestu z C_0^x do C_{acc} v grafu konfigurací $G_{M,x} = (V,E)$

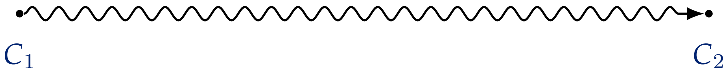
První návrh řešení: Použij DFS nebo BFS

- Oba algoritmy vyžadují paměť, která je přinejmenším lineární ve velikosti grafu
- $G_{M,x}$ může obsahovat až $2^{c_M f(n)}$ vrcholů pro nějakou konstantu c_M, která závisí na M
- DFS i BFS vyžadují prostor exponenciální v $f(n)$

Nemůžeme použít DFS ani BFS

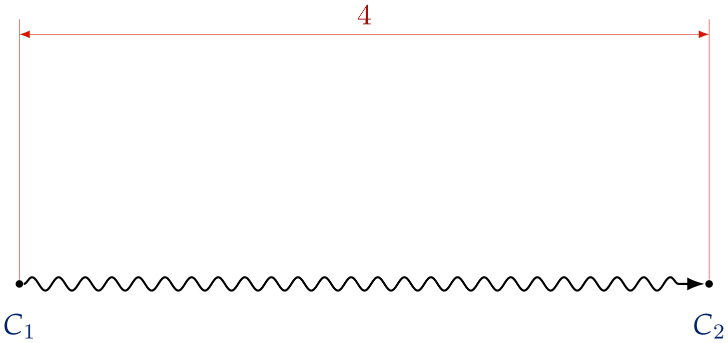
Rozděl a panuj

Existuje cesta z C_1 do C_2?



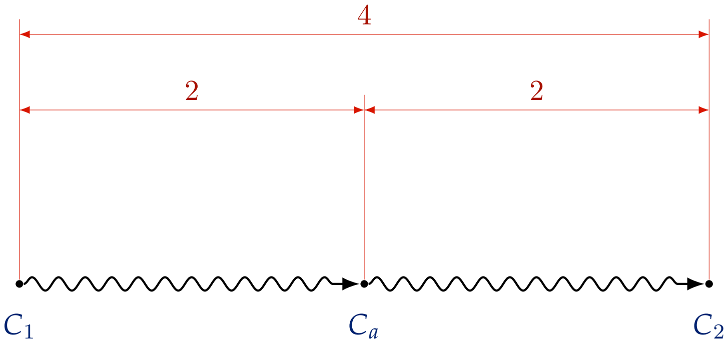
Rozděl a panuj

Existuje cesta z C_1 do C_2 délky nejvýš t?



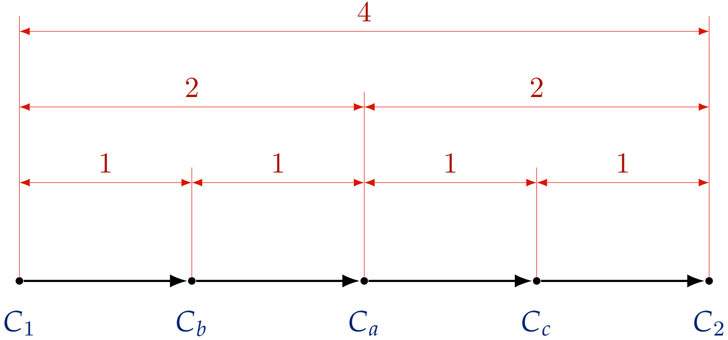
Rozděl a panuj

Existuje, pokud lze najít vrchol uprostřed



Rozděl a panuj

Půlení pokračuje dokud nedosáhneme délky nejvýš 1.



Hrany grafu, snadno ověřitelné

Prohledáváme graf s malou pamětí

- Předpokládejme (pro tuto chvíli), že M' může spočítat $f(n)$ pro vstup x
- Délka cesty z C_0^x do C_{acc} je nejvýš $2^{c_M f(n)}$

Rozděl a panuj

Cesta z C_1 do C_2 délky nejvýš 2^k existuje, právě když

1. $k = 0$ a $C_1 = C_2$ nebo $(C_1, C_2) \in E$, nebo
2. $k > 0$ a existuje prostřední vrchol C_m, pro který platí, že
   - existuje cesta z C_1 do C_m délky nejvýš 2^{k-1} a
   - existuje cesta z C_m do C_2 délky nejvýš 2^{k-1}

- Hloubka rekurze je $O(f(n))$
- Na každé úrovni rekurze je potřeba prostor $O(f(n))$ pro reprezentaci konfigurací C_1, C_2 a C_m
- Celkový prostor $O(f^2(n))$

Dosažitelnost

Funkce `Reachable(`C_1, C_2, k`)`

Vstup: Konfigurace C_1 a C_2, přirozené číslo k

Výstup: **true** pokud $G_{M,x}$ obsahuje cestu z C_1 do C_2 délky nejvýš 2^k, jinak **false**

if $k = 0$ **then**
  if $C_1 = C_2$ **or** $(C_1, C_2) \in E$ **then**
    return true
  else
    return false

foreach konfiguraci C_m využívající prostor $f(n)$ **do**
  if `Reachable(`C_1, C_m, $k-1$`)` **and** `Reachable(`C_m, C_2, $k-1$`)` **then**
    return true

return false

Volání `Reachable()`

- Pokud M' zná hodnotu $f(n)$, stačí mu zavolat

$$\texttt{Reachable}(C_0^x, C_{\mathrm{acc}}, c_M f(n))$$

- Jedna instance `Reachable()` používá prostor velikosti $O(f(n))$
  - konfigurace C_1, C_2, C_m používají prostor $f(n)$
  - $O(f(n))$ bitů stačí k reprezentaci těchto konfigurací
  - $O(f(n))$ bitů stačí k reprezentaci hodnoty k
- Hloubka rekurze je $O(f(n))$
- $O(f(n))$ instancí `Reachable()` je v každém okamžiku na zásobníku

Celkem je stačí prostor velikosti $O(f^2(n))$

Jsme tedy hotovi?

Jsme hotovi?

Co když M' nemůže spočítat hodnotu $f(n)$ pro vstup x?

- Funkce $f(n)$ nemusí být nutně algoritmicky vyčíslitelná
- $f(n)$ může být vyčíslitelná, ale k jejímu vyčíslení může být potřeba prostor $\omega(f^2(n))$
- I kdyby $f(n)$ byla vyčíslitelná v prostoru $O(f^2(n))$, funkce $f(n)$ může být neznámá, známe-li pouze M
- c_M je konstanta závisející na M
  - její hodnotu můžeme určit ze znalosti M

Je-li $f(n)$ neznámá

Idea

1. Zkoušej hodnoty $f(n) = 1, 2, 3, \ldots$
2. Zastav s hodnotou $f(n) = i$, pro niž
   - je nalezena cesta z C_0^x do C_{acc} nebo
   - z C_0^x není dosažitelná žádná konfigurace využívající prostor velikosti $i + 1$

Výpočet M'

Výpočet M' se vstupem x

```
1 i ← 1
  // Volání Reachable() předpokládají, že f(n) = i
2 if Reachable(C_0^x, C_acc, c_M · i) then přijmi
3 foreach konfiguraci C využívající prostor i + 1 do
4     if Reachable(C_0^x, C, c_M · i) then
5         i ← i + 1
6         goto 2
7 odmítni
```

> M' rozhoduje L v prostoru $O(f^2(n))$.

Více zdrojů, více síly

Prostor

Deterministická prostorová složitost

Připomenutí

Nechť $f : \mathbb{N} \to \mathbb{N}$ je funkce, která je definovaná pro každý vstup

- Deterministický Turingův stroj 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.
- $\mathrm{SPACE}(f(n))$ je třída jazyků přijímaných Turingovými stroji, které pracují v prostoru $O(f(n))$

Prostorová konstruovatelnost

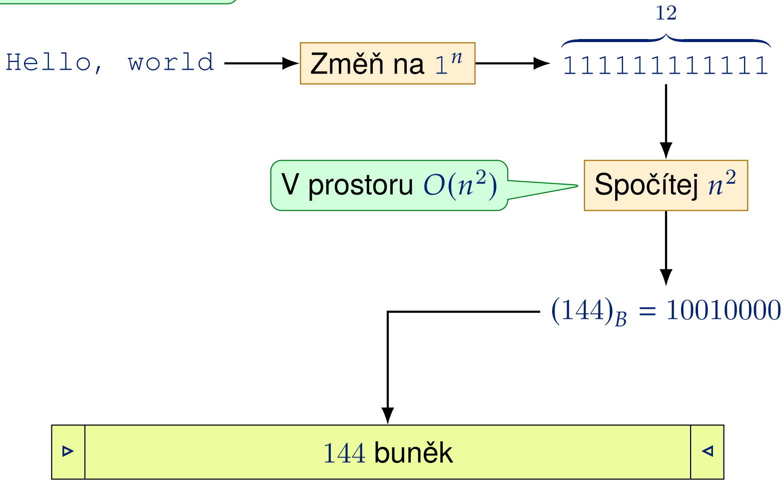
Definice

Funkci $f : \mathbb{N} \to \mathbb{N}$, kde $f(n) \geq \log n$, nazveme prostorově konstruovatelnou, je-li funkce, která zobrazuje 1^n na binární reprezentaci $f(n)$ vyčíslitelná v prostoru $O(f(n))$.

- Funkce obvykle používané pro měření prostorové složitosti jsou prostorově konstruovatelné, například
  - $\lceil \log_2 n \rceil$
  - $\lceil \sqrt{n} \rceil$
  - polynomy
  - $\lceil n \log_2 n \rceil$

Efektivní alokace paměti

Uvažme $f(n) = n^2$



Efektivní alokace paměti

Předpokládejme, že $f(n)$ je prostorově konstruovatelná strojem M_f

1. Se vstupem x délky $n = |x|$
2. Sestav řetězec $w = \mathtt{1}^n$
   - Každý znak x změň na $\mathtt{1}$
3. Vypočítej $k = f(n)$
   - Spusť $M_f(w)$
4. Vyznač k buněk na pásce

Využívá prostor $O(f(n))$, ne více, než potřebujeme alokovat.

Věta o deterministické prostorové hierarchii

Věta (o deterministické prostorové hierarchii)

Pro každou prostorově konstruovatelnou funkci $f : \mathbb{N} \to \mathbb{N}$ existuje jazyk A, který je rozhodnutelný v prostoru $O(f(n))$, nikoli však v prostoru $o(f(n))$.

Idea důkazu:

- A definujeme popsáním stroje D, který rozhoduje A
- Zajistíme, že
  - D pracuje v prostoru $O(f(n))$
  - Pro každý stroj M, který pracuje v prostoru $o(f(n))$ platí, že $L(M) \neq L(D)$
  - D toho dosahuje implementací diagonální metody

První nápad

První nápad konstrukce D

1. Se vstupem $x = \langle M \rangle$
2. Simuluj $M(\langle M \rangle)$ v prostoru $f(n)$
   - Pokud simulace potřebuje více prostoru, odmítni
3. Pokud M odmítl, přijmi, jinak odmítni

- Nechť M pracuje v prostoru $g(n) = o(f(n))$, pak

$$(\forall c \in \mathbb{R}^+)(\exists n_0 \in \mathbb{N})(\forall n \geq n_0)[cg(n) \leq f(n)]$$

- Prostor $f(n)$ postačuje k simulaci $M(x)$ se vstupy x, které jsou „dost dlouhé“
- Nezáleží na chování se stroji M, které nepracují v prostoru $o(f(n))$

Problém s asymptotikou

Není-li řetězec $\langle M\rangle$ dost dlouhý, prostor $f(n)$ nemusí stačit k simulaci $M(\langle M\rangle)$.

Řešení

- Uvážíme řetězce tvaru $\langle M\rangle\mathtt{10}^*$
- $x = \langle M\rangle\mathtt{10}^{n_0}$ je dost dlouhý pro nějaké n_0
- Prostor $f(n)$ stačí k simulaci $M(x)$
- $D(x)$ přijme, právě když $M(x)$ odmítne
- Tedy $L(M) \neq L(D)$

$\langle M\rangle\mathtt{1}$

$\langle M\rangle\mathtt{10}$

$\langle M\rangle\mathtt{100}$

$\langle M\rangle\mathtt{1000}$

$\langle M\rangle\mathtt{10000}$

$\langle M\rangle\mathtt{100000}$

$\vdots$

Problém se zastavením

I je-li prostor $f(n)$ dostatečný pro simulaci $M(\langle M \rangle)$, výpočet se může zacyklit.

Řešení

Zastav, pokud simulace vyžaduje více než $2^{f(n)}$ kroků.

- Nechť M pracuje v prostoru $g(n) = o(f(n))$
- Uvažme vstup x délky $n = |x|$
- Je-li $M(x)\downarrow$, pak výpočet skončí do $2^{c_M g(n)}$ kroků pro nějakou konstantu c_M
- Je-li n dost velké, simulace $M(x)$ skončí do $2^{f(n)}$ kroků

Stroj D

Výpočet D se vstupem x

$n \leftarrow |x|$
Vypočti $f(n)$ pomocí prostorové konstruovatelnosti
Označ $f(n)$ buněk na pracovní pásce
if v následujících krocích hlava opustí vyznačený prostor **then**
    odmítni
if x nemá tvar $\langle M \rangle \mathtt{1}\mathtt{0}^*$ **then** odmítni
Simuluj $M(x)$ s počítáním kroků simulace
if počet simulovaných kroků překročí $2^{f(n)}$ **then** odmítni
if M přijal **then** odmítni **else** přijmi

Definujeme $A = L(D)$

Prostor použitý strojem D

- Výpočet $f(n)$ vystačí s prostorem $O(f(n))$ díky prostorové konstruovatelnosti
- Poté hlava D zůstane v rámci $f(n)$ vyznačených buněk
- D tedy pracuje v prostoru $O(f(n))$
- Čítač kroků lze reprezentovat pomocí $f(n)$ bitů
  - Může být na další stopě

A je rozhodnutelný v prostoru $O(f(n))$.

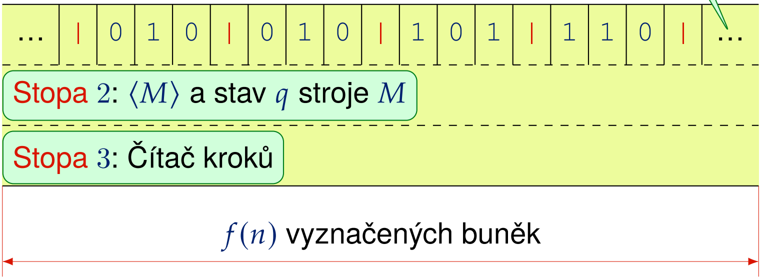
D simuluje M

Vstupní páska D (i M)

$\langle M \rangle 1000000000000$

Stopa 1: páska M
každý znak zakódován
$b = \lceil \log_2 |\Sigma| \rceil$ bity

Pracovní páska D má tři stopy



Menší prostor nestačí

- Nechť $M = (Q, \Sigma, \delta, q_0, F)$ je stroj, který pracuje v prostoru $g(n) = o(f(n))$
- Ukážeme, že $A \neq L(M)$

Pro nějakou konstantu c_M platí, že se vstupem x délky $n = |x|$

- $M(x)$ lze simulovat v prostoru $c_M g(n)$
- $M(x)$ skončí výpočet do $2^{c_M g(n)}$ kroků

- Připomeňme univerzální TS
- Existuje konstanta c_M taková, že M se vstupem x má nejvýš $2^{c_M g(n)}$ různých konfigurací
- Prostor $c_M g(n)$ stačí k reprezentaci jedné konfigurace
  - $\lceil \log_2 |\Sigma| \rceil$ bitů pro každé políčko pracovní pásky M
  - $\lceil \log_2 |Q| \rceil$ bitů pro reprezentaci stavu M

Menší prostor nestačí

$$g(n) = o(f(n)) \implies (\exists n_0 \in \mathbb{N})(\forall n \geq n_0)[c_M g(n) \leq f(n)]$$

Existují konstanty c_M a n_0 takové, že se vstupem x délky $n \geq n_0$

- $M(x)$ lze simulovat v prostoru $c_M g(n) \leq f(n)$
- $M(x)$ skončí výpočet do $2^{c_M g(n)} \leq 2^{f(n)}$ kroků

- Simulace M se vstupem $x = \langle M \rangle 10^{n_0}$ skončí a
- $D(x)$ přijme právě když $M(x)$ odmítne

$$L(D) \neq L(M)$$

Prostorová hierarchie

Věta (o deterministické prostorové hierarchii)

Pro každou prostorově konstruovatelnou funkci $f : \mathbb{N} \to \mathbb{N}$ *existuje jazyk* A, *který je rozhodnutelný v prostoru* $O(f(n))$, *nikoli však v prostoru* $o(f(n))$.

Důsledek

Jsou-li $f_1, f_2 : \mathbb{N} \to \mathbb{N}$ *funkce, pro které platí, že* $f_1(n) \in o(f_2(n))$ *a* f_2 *je prostorově konstruovatelná, potom*

$$\mathrm{SPACE}(f_1(n)) \subsetneq \mathrm{SPACE}(f_2(n))$$

Polynomy a související funkce

Důsledek

Pro každá dvě reálná čísla $0 \leq \epsilon_1 < \epsilon_2$ *platí, že*

$$\mathrm{SPACE}(n^{\epsilon_1}) \subsetneq \mathrm{SPACE}(n^{\epsilon_2})$$

- Je-li ϵ_2 racionální číslo, pak
  - n^{ϵ_2} je prostorově konstruovatelná
    - Lze jednoduše ukázat pro přirozená čísla
    - Lze ukázat i pro racionální čísla
  - Ostrá inkluze plyne z prostorové hierarchie
- Je-li ϵ_2 iracionální číslo
  - Racionální čísla jsou hustá v reálných číslech
  - Existuje racionální číslo ϵ splňující $\epsilon_1 < \epsilon < \epsilon_2$
  - Z prostorové hierarchie a prostorové konstruovatelnosti n^ϵ

$$\mathrm{SPACE}(n^{\epsilon_1}) \subsetneq \mathrm{SPACE}(n^{\epsilon}) \subseteq \mathrm{SPACE}(n^{\epsilon_2})$$

Logaritmický,polynomiální a exponenciální prostor

Důsledek

$\mathrm{NL} \subsetneq \mathrm{PSPACE} \subsetneq \mathrm{EXPSPACE} = \bigcup_{k\in\mathbb{N}} \mathrm{SPACE}(2^{n^k})$.

Definice

Savičova věta

Prostorová hierarchie

$$\mathrm{NL} = \mathrm{NSPACE}(\log_2 n) \subseteq \mathrm{SPACE}((\log_2 n)^2) \subsetneq \mathrm{PSPACE} \subsetneq \mathrm{EXPSPACE}$$