

Základy složitosti a vyčíslitelnosti

NTIN090

Petr Kučera

2024/25 (6. přednáška)

Dnešní plán

- Třída NP
- Nedeterministické Turingovy stroje
- Vztah času k prostoru
- Vztah prostoru k času
- Savičova věta

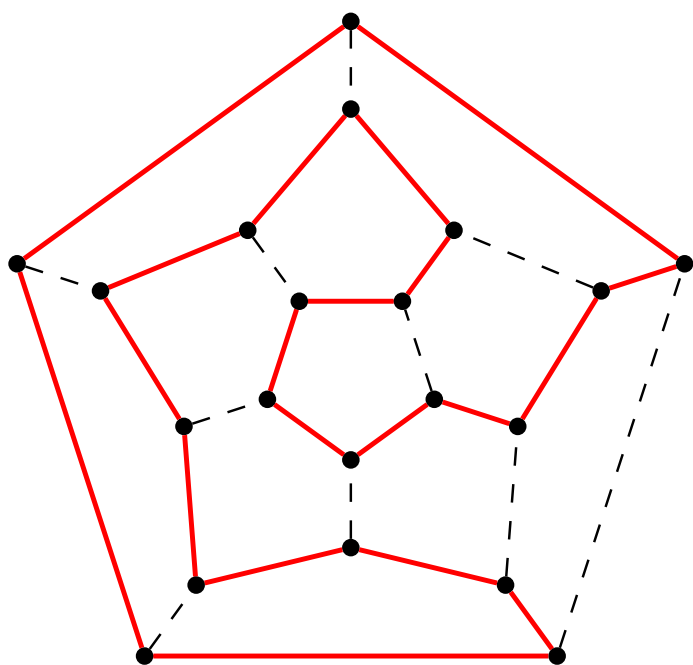
Problémy ověřitelné v polynomiálním čase

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 $|x|$

- Řetězec y zveme také **certifikátem** x
- **Polynomiální verifikátor** pracuje v čase $O(|x|^k)$ pro nějaké $k \in \mathbb{N}$
 - Přečte jen prefix y polynomiální délky
 - **Polynomiální certifikát** má délka 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 $G = (V, E)$, posloupnost vrcholů $\mathbf{v} = (v_1, \dots, v_\ell)$

- 1 **if** $\ell \neq |V|$ **then odmítni**
 - 2 **if** $v_i = v_j$ pro nějakou dvojici indexů $i \neq j$ **then odmítni**
 - 3 **for** $i = 1$ **to** $n - 1$ **do**
 - 4 **if** $\{v_i, v_{i+1}\} \notin E$ **then odmítni**
 - 5 **if** $\{v_n, v_1\} \notin E$ **then odmítni**
 - 6 **přijmi**
-

Definice

NP je třídou jazyků, které mají polynomiální verifikátory.

- Často odpovídají úlohám, u nichž jsme schopni v polynomiálním čase ověřit, zda daný řetězec polynomiální délky 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 $P \subseteq NP$

Otázka, zda $P = NP$ je otevřená.

Těžko ale ověřím, když mi NP problém říká, že nemá řešení.

Nedeterministický Turingův stroj

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

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

Roste exponenciálně s časem (2^n výpočtů po n krocích),

*takže těžko to spustím
všechno vyložene paralelně)*

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

Zajímá nás, jestli existuje nějaká přijímající větev.

- 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**

konfigurace tvoří strom všech možných výpočtů



Nejde o realistický výpočetní model

Nevychází nic silnějšího než přijímající TS. Je to jenom zjednodušení práce s existenčním kvantifikátorem.

Jazyk přijímaný NTS

Předpokládejme NTS M a vstup x

Výpočet M **nad** x je posloupnost konfigurací $C_0, C_1, C_2, \dots$, 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 přijímajícím stavu

M **přijme slovo** x pokud existuje přijímající výpočet M nad x

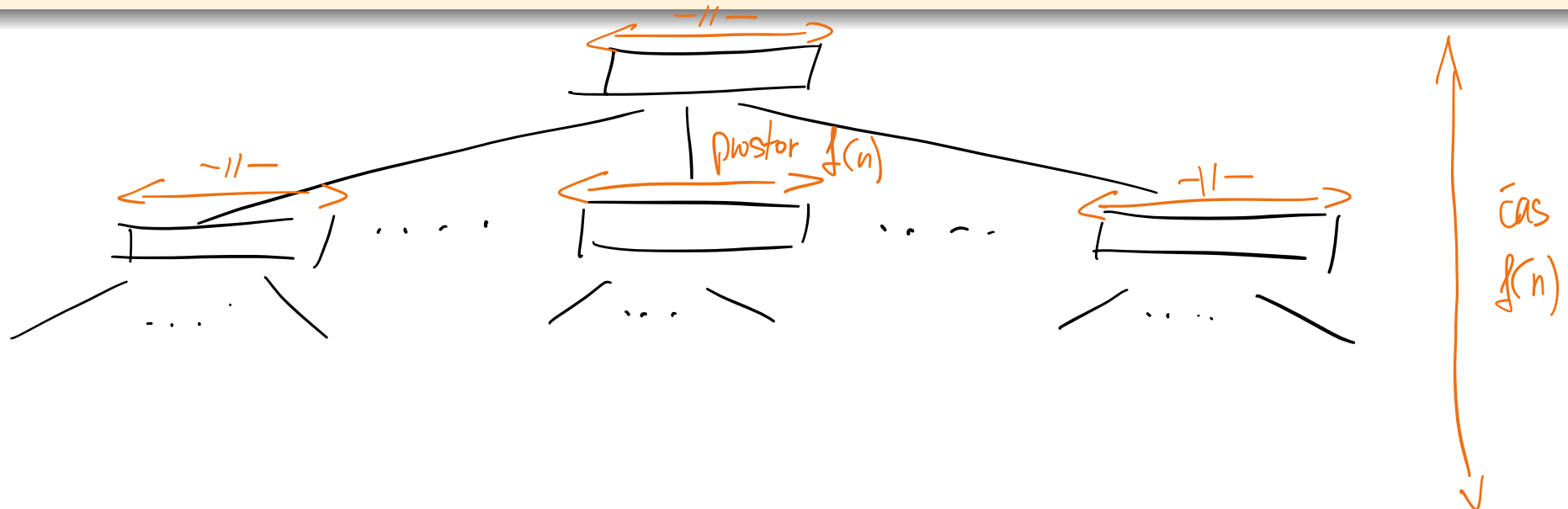
$L(M)$ jazyk slov přijímaných M

Časová a prostorová složitost NTS

Definice

Nechť M je nedeterministický Turingův stroj a nechť $f : \mathbb{N} \rightarrow \mathbb{N}$ 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 $f : \mathbb{N} \rightarrow \mathbb{N}$ je funkce, potom definujeme třídy:

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

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

Tvrzení

Pro každou funkci $f : \mathbb{N} \rightarrow \mathbb{N}$ platí $\hookrightarrow$ čistě proto, že každý TS je speciální NTS

$$\text{TIME}(f(n)) \subseteq \text{NTIME}(f(n))$$

$$\text{SPACE}(f(n)) \subseteq \text{NSPACE}(f(n))$$

NP = nedeterministicky polynomiální

Věta (Alternativní definice třídy NP)

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

— NP je třída jazyků přijímaných NTS v poly. čase

Důkaz.

Ve dvou krocích

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



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

- Nechť $L \in NP$
- Máme polynomiální verifikátor $V(x, y)$ pro L
- Následující NTS M přijímá 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**
-

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

M přijímá L v polynomiálním čase

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

- 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

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

a ověříme to právě tak, že ten vektor (v poly. čase) spustíme ←

- y_i — větev zvolená v kroku i , $i = 1, \dots, p(|x|)$
→ takže se můžeme zohledňovat do „indexů“, kterými vektor si při výpočtu zvolit“

Řetězec y je certifikátem toho, že M přijímá x .

– řetězec konfigurací vedoucích až vstupem do přijímacího stavu je sám o sobě certifikátem

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

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

- 1 Simuluj $M(x)$ s větvením podle y
 - 2 **if** větev výpočtu $M(x)$ popsaná y přijme **then**
 - 3 **přijmi**
 - 4 **else**
 - 5 **odmítni**
-

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

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

Základní vztahy

Nedeterministický čas a deterministický prostor

Věta

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

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

- Nechť L je přijímán NTS M v čase $\underline{g(n) = O(f(n))}$
- Předpokládejme, že $M = (Q, \Sigma, \delta, q_0, F)$ *čas toho stroje*
- 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

$$y \in \{1, \dots, r\}^{g(|x|)}$$

- y_i — větev zvolená v kroku i , $i = 1, \dots, g(|x|)$
- Popíšeme TS M' , který rozhoduje L v prostoru $O(f(n))$

Důkaz (část 2)

První nápad

Výpočet M' se vstupem x

```
1 forall  $y \in \{1, \dots, r\}^{g(|x|)}$  do  
2   Simuluj  $M(x)$  s větvením podle  $y$   
3   if větev výpočtu  $M(x)$  popsaná  $y$  přijme then  
4     přijmi  
5 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

```
1  $k \leftarrow 1$ 
2 repeat
3   forall  $y \in \{1, \dots, r\}^k$  do
4     Simuluj  $M(x)$  s větvením podle  $y$ 
5     if větev výpočtu  $M(x)$  popsaná  $y$  přijme then přijmi
6    $k \leftarrow k + 1$ 
7 until všechny simulace  $M(x)$  větvící podle  $y$  odmítly
8 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 $k \leq g(|x|)$

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 $f : \mathbb{N} \rightarrow \mathbb{N}$ platí, že

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

Důsledek

$$\text{NP} \subseteq \text{PSPACE}$$

Model TS s menším než lineárním prostorem

Pro prostor menší než lineární uvažujeme vícepáskový TS:

Vstupní páska je **jen pro čtení**

Pracovní pásy 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  protože vstupní musí být vždy alespoň lineární
 - 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

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

- Třída jazyků přijímaných NTS v logaritmickém prostoru

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

- Třída jazyků přijímaných NTS v polynomiálním prostoru

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

Vztah prostoru a času

Věta (Vztah prostoru a času)

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

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

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

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

Vztah prostoru a času

Věta (Vztah prostoru a času)

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

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

Idea důkazu

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

$x \in L \iff G_{M,x}$ obsahuje 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

$M(x)$ přijme $\iff G_{M,x}$ obsahuje cestu z C_0^x 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 $f(n)$
- Necht' x je vstup délky $n = |x|$
- Necht' $G_{M,x} = (V, E)$ 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)

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

Konfigurace se skládá z následujících prvků

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

Počet konfigurací

$$|V| \leq |Q| \cdot n \cdot f(n) \cdot |\Sigma|^{f(n)}$$

$$= 2^{\log_2 |Q|} \cdot 2^{\log_2 n} \cdot 2^{\log_2 f(n)} \cdot 2^{f(n) \log_2 |\Sigma|}$$

$$= 2^{\log_2 |Q| + \log_2 n + \log_2 f(n) + f(n) \log_2 |\Sigma|}$$

$$\leq 2^{f(n)(\log_2 |Q| + 1 + 1 + \log_2 |\Sigma|)} \quad (f(n) \geq \log_2 n)$$

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

$$|V| \leq 2^{c_M f(n)}$$

$$|E| \leq |V|^2 \leq 2^{2c_M f(n)}$$

Vztah prostoru a času

Věta (Vztah prostoru a času)

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

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

Idea důkazu

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

- 1 Sestav počáteční konfiguraci C_0^x výpočtu $M(x)$
 - 2 Projdi graf $G_{M,x}$ prohledáváním do hloubky (DFS) počínaje ve vrcholu C_0^x
 - 3 **if** DFS nalezne přijímající konfiguraci **then**
 - 4 **přijmi**
 - 5 **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 času (poznámky)

Věta (Vztah prostoru a času)

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

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

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

Z věty **NEPLYNE**

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

$$\text{NL} \subseteq \text{P}$$

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

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

Důsledek

$$\text{NL} \subseteq \text{P}$$

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

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

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

$$\text{NPSPACE} \subseteq \text{EXPTIME}$$

- Předpokládejme, že $L \in \text{NPSPACE}$
 $\Rightarrow L \in \text{NSPACE}(n^k)$ pro nějaké $k \in \mathbb{N}$
- Podle věty o vztahu prostoru a času existuje konstanta $c_L \in \mathbb{N}$, pro kterou platí

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

Věta

Platí následující řetězec inkluzí:

$$L \subseteq NL \subseteq P \subseteq NP \subseteq PSPACE \subseteq NPSPACE \subseteq EXPTIME$$

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

Vztahy mezi třídami

Věta

Platí následující řetězec inkluzí:

$$L \subseteq NL \subseteq P \subseteq NP \subseteq PSPACE \subseteq NPSPACE \subseteq EXPTIME$$

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

Věta

Platí následující řetězec inkluzí:

$$L \subseteq NL \subseteq P \subseteq NP \subseteq PSPACE \subseteq NPSPACE \subseteq EXPTIME$$

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

Vztahy mezi třídami

Věta

Platí následující řetězec inkluzí:

$$L \subseteq NL \subseteq P \subseteq NP \subseteq PSPACE \subseteq NPSPACE \subseteq EXPTIME$$

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

Savičova věta

Savičova věta

Věta (Savičova věta)

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

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

Důsledek

$$\text{PSPACE} = \text{NPSPACE}$$

Savičova věta (začátek důkazu)

- Předpokládejme, že $L \in \text{NSPACE}(f(n))$
- 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 $\text{SPACE}(g^2(n)) \subseteq \text{SPACE}(f^2(n))$

Technické předpoklady

- M nepohne hlavou na pracovní pásce nalevo od počáteční pozice
- M má jednoznačnou přijímající konfiguraci C_F
 - 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

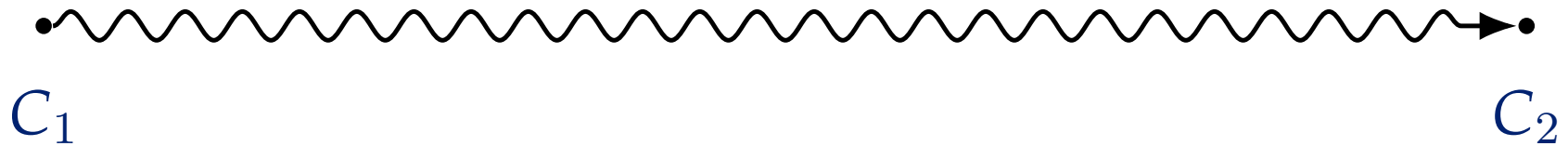
Se vstupem x , hledej cestu z C_0^x do C_F v grafu konfigurací $G_{M,x} = (V, E)$

První návrh řešení: Použij DFS nebo BFS

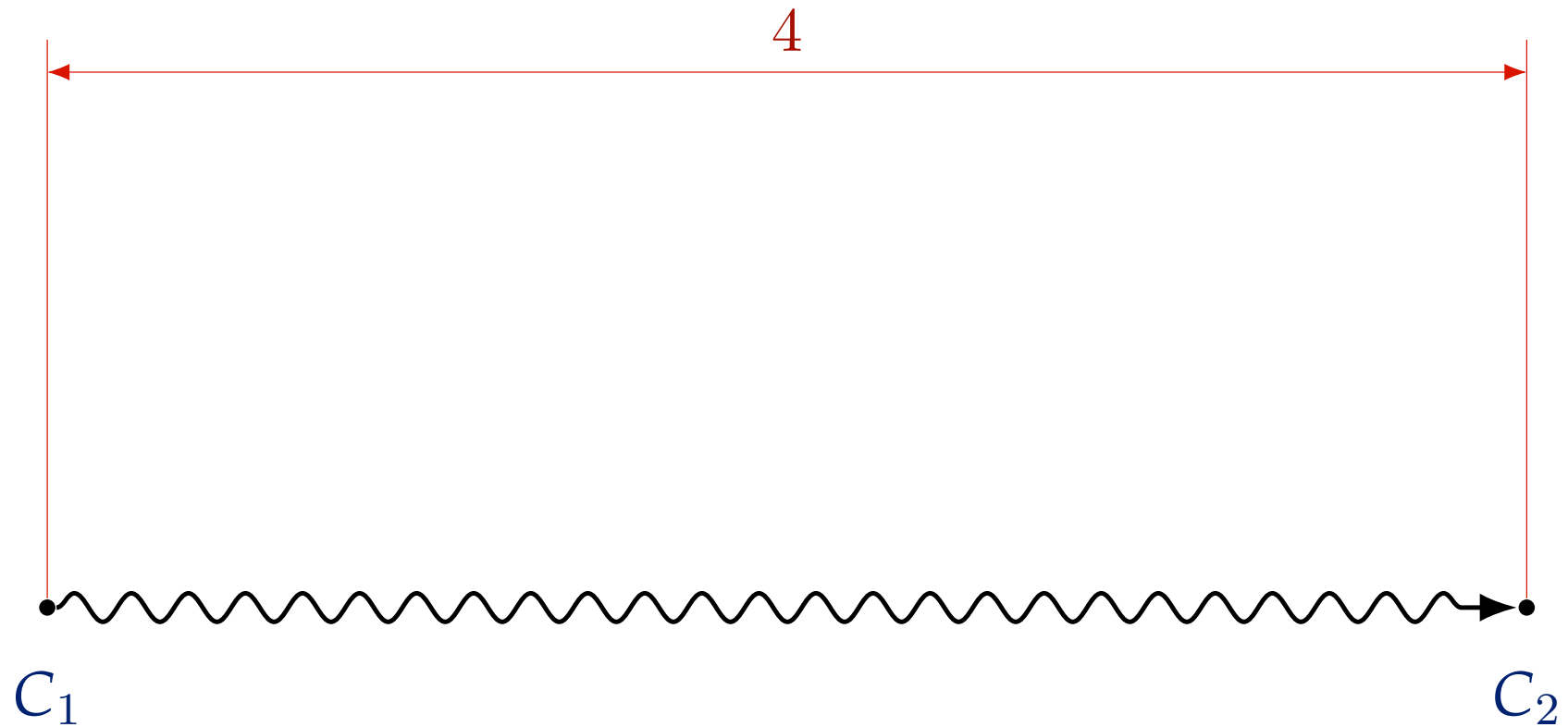
- Oba algoritmy vyžadují paměť, která je přinejmenším lineární ve velikosti grafu
- $G_{M,x}$ může obsahovat až $2^{c_M f(n)}$ vrcholů pro nějakou konstantu c_M , která závisí na M
- DFS i BFS vyžadují prostor exponenciální v $f(n)$

Nemůžeme použít DFS ani BFS

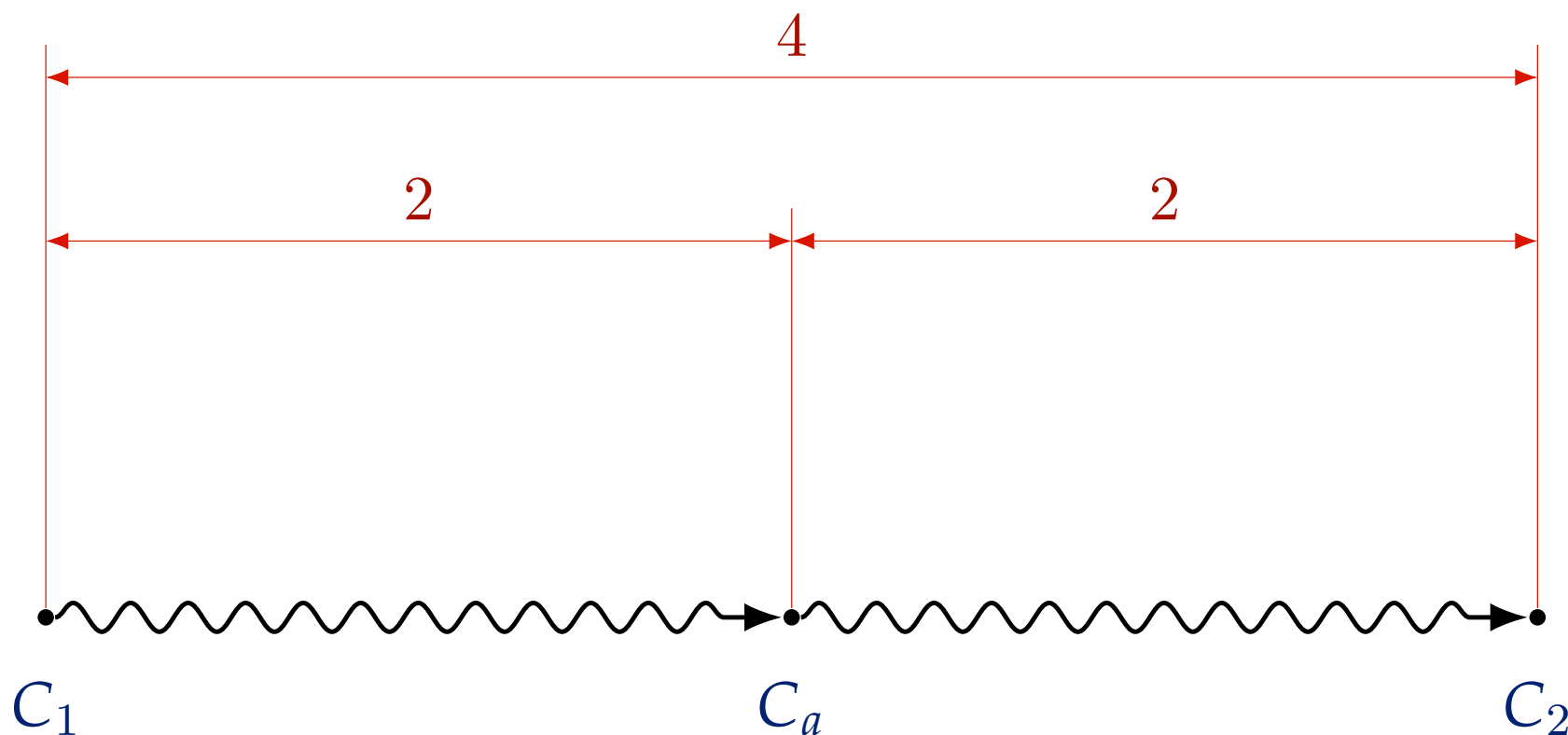
Existuje cesta z C_1 do C_2 ?



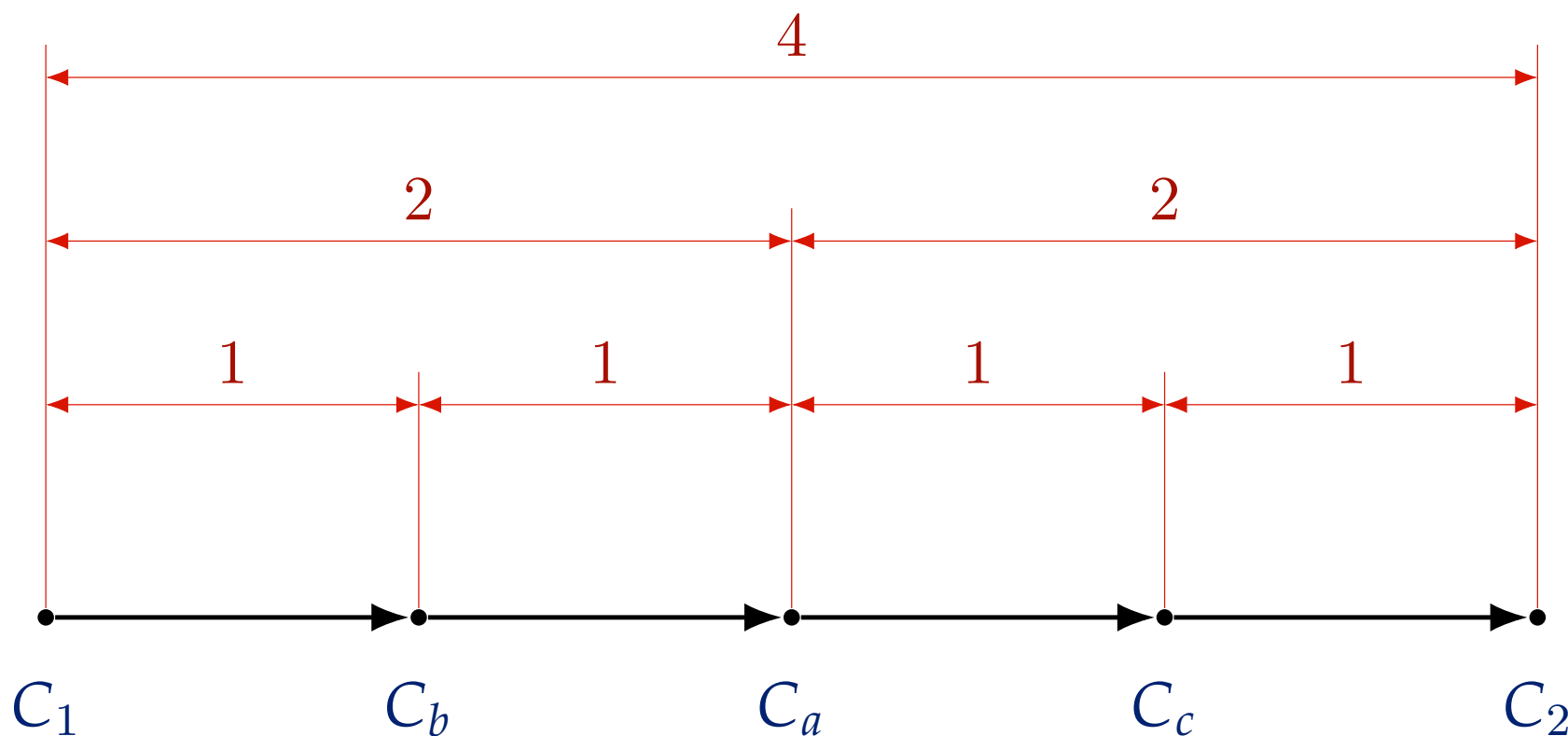
Existuje cesta z C_1 do C_2 délky nejvýš t ?



Existuje, pokud lze najít vrchol uprostřed



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_F 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

`Reachable( $C_0^x$ ,  $C_F$ ,  $c_M f(n)$ )`

- 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 závisající na M
 - její hodnotu můžeme určit ze znalosti M

Je-li $f(n)$ neznámá

Idea

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

Výpočet M'

Výpočet M' se vstupem x

1 $i \leftarrow 1$

// Volání `Reachable()` předpokládají, že $f(n) = i$

2 **if** `Reachable`(C_0^x , C_F , $c_M \cdot i$) **then přijmi**

3 **foreach** konfiguraci C využívající prostor $i + 1$ **do**

4 **if** `Reachable`(C_0^x , C , $c_M \cdot i$) **then**

5 $i \leftarrow i + 1$

6 **goto** 2

7 **odmítni**

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