# IB107 Vyčíslitelnost a složitost

## třída P, algoritmus C-Y-K, třída NP

Jan Strejček

Fakulta informatiky
Masarykova univerzita

# opakování

$\mathrm{P} = \{L \mid L$ je rozhodovaný nějakým deterministickým jednopáskovým (nebo vícepáskovým) TM $\mathcal{M}$ s časovou složitostí $T_{\mathcal{M}}(n) \in \mathcal{O}(n^k)$ pro $k \in \mathbb{N}\}$.

$\mathrm{NP} = \{L \mid L$ je rozhodovaný nějakým nedeterministickým jednopáskovým (nebo vícepáskovým) TM $\mathcal{M}$ s časovou složitostí $T_{\mathcal{M}}(n) \in \mathcal{O}(n^k)$ pro $k \in \mathbb{N}\}$.

- z definic plyne $\mathrm{P} \subseteq \mathrm{NP}$
- otevřený problém $\mathrm{P} \stackrel{?}{=} \mathrm{NP}$

# příslušnost problémů v P

- stačí ukázat, že problém je řešitelný v polynomiálním počtu kroků a že každý krok je proveditelný v polynomiálním čase
- kódování/dekódování objektů $O$ do/ze slov $\langle O \rangle$ musí být proveditelné v polynomiálním čase
- příklad vhodného kódování: reprezentace grafu maticí sousednosti

- příklad nevhodného kódování: reprezentace sekvence číslic unárním zápisem čísla

# problém existence cesty

## Definice (problém existence cesty)

*Problém existence cesty je problém rozhodnout, zda v daném orientovaném grafu $G$ existuje cesta z $s$ do $t$.*

$$PATH = \{\langle G, s, t\rangle \mid G \text{ je orientovaný graf obsahující cestu z } s \text{ do } t\}$$

## Věta 1.10

$PATH \in P$.

**Důkaz:** Postupně spočítáme uzly dosažitelné z $s$.

1. označ uzel $s$
2. dokud lze označit nový uzel opakuj: projdi všechny hrany v $G$ a označ každý uzel, do kterého vede hrana z označeného uzlu
3. je-li $t$ označeno, akceptuj; jinak zamítni

Celkem $\mathcal{O}(n)$ kroků ($n$ je počet uzlů v $G$), každý lze provést v polynomiálním čase. ∎

# uzávěrové vlastnosti P

**Věta**

*Třída P je uzavřená na sjednocení, průnik, komplement a zřetězení.*

**Důkaz:** Nechť $L_1, L_2 \in$ P. Předpokládáme, že $L_i$ je rozhodován jednopáskovým det. TM $\mathcal{M}_i$ s časovou složitostí v $T_{\mathcal{M}_i}(n) \in \mathcal{O}(n^{k_i})$.

∎

# bezkontextové jazyky a třída P

**Věta 1.12**

*Pro každý bezkontextový jazyk $L$ platí $L \in P$.*

**Důkaz:**

- každý bezkontexotvý jazyk $L \subseteq \Sigma^*$ lze popsat gramatikou v Chomského normální formě (CNF)
- pro pevně danou gramatiku $G$ v CNF a pro slovo $w \in \Sigma^*$ lze v čase $\mathcal{O}(|w|^3)$ pomocí algoritmu Cocke-Younger-Kasami rozhodnout, zda $w \in L(G)$ ∎

# intuice k algoritmu C-Y-K

$S \rightarrow AB \mid CD \mid EF$

Platí $S \Rightarrow^* w$ ?

# intuice k algoritmu C-Y-K

## myšlenka

Pro každé neprázdné podslovo $u$ slova $w$ spočítáme množinu

$T_u$ všech neterminálů, z kterých lze odvodit $u$.

- $u = a$
- $u = ab$
- $u = abc$

# příklad

$$
\begin{array}{lcl}
S & \rightarrow & AB \mid SS \mid a \\
A & \rightarrow & AA \mid BC \mid a \\
B & \rightarrow & AB \mid b \\
C & \rightarrow & SA \mid b
\end{array}
$$

Platí $S \Rightarrow^* abaa$ ?

# příklad

$$
\begin{array}{lcl}
S & \rightarrow & AB \mid SS \mid a \\
A & \rightarrow & AA \mid BC \mid a \\
B & \rightarrow & AB \mid b \\
C & \rightarrow & SA \mid b
\end{array}
$$

$$T_{i,j} = \{X \in N \mid X \Rightarrow^* w_i w_{i+1} \dots w_{i+j-1}\}$$

$$w = abaa$$

| | 1 | | | |
|---|---|---|---|---|
| 4 | | 2 | | |
| 3 | | | 3 | |
| 2 | | | | 4 |
| 1 | | | | |
| | $a$ | $b$ | $a$ | $a$ |

| | 1 | | | |
|---|---|---|---|---|
| 4 | | 2 | | |
| 3 | | | 3 | |
| 2 | | | | 4 |
| 1 | | | | |
| | $a$ | $b$ | $a$ | $a$ |

# algoritmus Cocke-Younger-Kasami

**Vstup:** gramatika $\mathcal{G} = (N, \Sigma, P, S)$ v CNF, slovo $w = w_1 \dots w_n \neq \varepsilon$
**Výstup:** množiny $T_{i,j} = \{X \in N \mid X \Rightarrow^* w_i \dots w_{i+j-1}\}$

```
for i ← 1 to n do
    T_{i,1} ← ∅
    for každé pravidlo tvaru (A → a) ∈ P do
        if a = w_i then T_{i,1} ← T_{i,1} ∪ {A}
    od
od
for j ← 2 to n do
    for i ← 1 to n − j + 1 do
        T_{i,j} ← ∅
        for k ← 1 to j − 1 do
            for každé pravidlo tvaru (A → BC) ∈ P do
                if B ∈ T_{i,k} ∧ C ∈ T_{i+k,j−k} then T_{i,j} ← T_{i,j} ∪ {A}
            od
        od
    od
od
```

# problém hamiltonovské cesty

hamiltonovská cesta = cesta procházející každým uzlem právě jednou

### Definice (problém hamiltonovské cesty)

*Problém hamiltonovské cesty je problém rozhodnout, zda v daném orientovaném grafu $G$ existuje hamiltonovská cesta z $s$ do $t$.*

$$\begin{aligned} HAMPATH = \{\langle G, s, t\rangle \mid\ & G \text{ je orientovaný graf obsahující} \\ & \text{hamiltonovskou cestu z } s \text{ do } t\} \end{aligned}$$

# problém hamiltonovské cesty

**Věta**

*HAMPATH* $\in$ *NP*.

**Důkaz:**

- hamiltonovská cesta v grafu $G$ s $n$ uzly má délku $n-1$
- hamiltonovskou cestu budeme nedeterministicky hádat
  1. začni budovat cestu z uzlu $s$
  2. $(n-1)$-krát opakuj: nedeterministicky vyber hranu vedoucí z posledního uzlu cesty a přidej ji na konec cesty
  3. je-li $t$ poslední uzel cesty a žádný uzel se neopakuje, akceptuj; jinak zamítni
- každý výpočet má $\mathcal{O}(n)$ polynomiálních kroků
- hamiltonovská cesta existuje $\iff$ existuje akceptující výpočet

$\blacksquare$

# problém složených čísel

## Definice (problém složených čísel)

*Problém složených čísel je problém rozhodnout, zda je dané číslo x složené, tedy součinem dvou čísel větších než 1.*

$$COMPOSITES = \{\langle x \rangle \mid \ x = pq \text{ pro nějaká přirozená čísla } p, q > 1\}$$

## Věta

$COMPOSITES \in NP$.

**Důkaz:** Nedeterministicky zvolíme číslo $p$ takové, že $1 < p < x$. Pokud $p$ je dělitelem $x$, akceptujeme, jinak zamítneme. ■

V roce 2002 bylo dokázáno, že $COMPOSITES \in$ P.

# problém splnitelnosti (SAT)

## Definice (problém splnitelnosti (SAT))

*Problém splnitelnosti (SAT) je problém rozhodnout, zda je daná výroková formule (využívající pouze operace $\wedge$, $\vee$ a $\neg$) splnitelná.*

$$SAT = \{\langle\varphi\rangle \mid \varphi \text{ je splnitelná výroková formule}\}$$

## Věta

*$SAT \in NP$.*

**Důkaz:**

$\blacksquare$

# polynomiální verifikátor

"Řešení" problému lze deterministickým TM v polynomiální čase

- nalézt, pokud je problém v P
- ověřit, pokud je problém v NP (když nám řešení někdo dodá)

**Definice (polynomiální verifikátor)**

*Polynomiální verifikátor pro jazyk L je deterministický TM $\mathcal{V}$ splňující*

$$w \in L \iff \textit{existuje řetězec } c \textit{ takový, že } \mathcal{V} \textit{ akceptuje } \langle w, c \rangle$$

*a pracující v polynomiálním čase vzhledem k $|w|$.*

- $c$ se nazývá svědek, důkaz nebo certifikát příslušnosti $w$ do $L$
- lze předpokládat, že velikost $c$ je polynomilální vzhledem k $|w|$, protože polynomiální verifikátor více znaků z $c$ nemůže přečíst

# polynomiální verifikátory a NP

**Věta**

*$L \in NP \iff$ existuje polynomiální verifikátor pro L.*

**Důkaz:**

$\Longrightarrow$ Nechť $\mathcal{M}$ je nedeterministický TM akceptující $L$ v polynomiálním čase. Verifikátor bude pro vstup $\langle w, c\rangle$ simulovat $\mathcal{M}$ na vstupu $w$ a $c$ bude používat k deterministickému výběru z možných přechodů.

$\Longleftarrow$ Nechť $\mathcal{V}$ je polynomiální verifikátor pro $L$ pracující na vstupech $\langle w, c\rangle$ v čase $\mathcal{O}(|w|^k)$. Nedeterministický stroj $\mathcal{M}$ nedeterministicky zvolí řetězec $c$ délky nejvýše $|w|^k$ a pak simuluje $\mathcal{V}$ na vstupu $\langle w, c\rangle$. ∎

# uzávěrové vlastnosti NP

**Věta**

*Třída NP je uzavřená na sjednocení, průnik a zřetězení.*

**Důkaz:**

- sjednocení a průnik: analogicky jako pro P
- zřetězení:

# uzavřenost NP na doplněk

- není známo, zda je třída NP uzavřená na doplněk
- například nevíme, zda $\overline{HAMPATH} \in \text{NP}$,
- tedy nevíme, jak polynomiálně verifikovat $\overline{HAMPATH}$
- lze definovat třídu

$$\text{coNP} = \{L \mid \overline{L} \in \text{NP}\}$$

- věří se, že platí $\text{NP} \neq \text{coNP}$, tj. NP není uzavřená na doplněk
- z $\text{NP} \neq \text{coNP}$ plyne $\text{P} \neq \text{NP}$
- existují problémy, které jsou v $\text{NP} \cap \text{coNP}$, ale není známo, zda jsou v P, například problém paritních her