

# Přednáška Pa NP

Třída P: Řekneme, že rozhodovací úloha  $U$  leží ve třídě P, jestliže existuje deterministický TM, který rozhodne jazyk  $L_U$  a pracuje v polynomickém čase; tj. fce  $T(n)$  je  $O(p(n))$  pro nějaký polynom  $p(n)$ .

Jazyk úlohy: Jazyk úlohy  $U$  znacíme  $L_U$ , je množina všech instance úlohy, pro které je správné rozhodnutí AND. TM rozhodne NE pro všechna slova  $\in \Sigma^*$   $\sim L_U$ .

Formulace instance úlohy jako slova - například SAT:  $\varphi = (x_1 \vee x_2 \vee \neg x_3) \wedge (x_2 \vee \neg x_3 \vee x_4)$   
 $\Sigma = \{0, 1, X, (,), \vee, \neg\}$ , čísla  $x_i$  převodíme na binární, ostatní znaky opětne  
 $w = "(x_1 \vee x_2 \wedge \neg x_3) \vee (\neg x_2 \vee \neg x_3 \vee x_4)"$

Příklady problémů v třídě P:

- existuje v grafu kostra s cestou nejdéle k?: nejmenší kostra
- existuje vnitrobarevná cesta z u do v s cestou nejdéle k (v acyklickém grafu)? nejkratší cesta
- existuje v síti případný lok?: existuje tam lok velikosti aspoň k?: maximální lok
- existuje v síti řez, který má cenu nejdéle k?: minimální řez

Třída NP: Rozhodovací úloha  $U$  patří do třídy NP, právě když existuje nedeterministický TM, který příjmí  $L_U$  a pracuje v polynomickém čase.

Příklady problémů v třídě NP: (rozhodovací verze)

- SAT problem - splnitelnost formulí
- existence Hamiltonovského cyklu / cesty / hranice
- úloha řešení
- problem kličky - hledání co největšího podgrafu, který ji upíná

Nedeterministický algoritmus: pracuje ve dvou fázích

- 1) ráhodná významná řetězec s (tělo bude řešením úlohy)
- 2) navádění se deterministickým alg., který ověří,  $\in O(p(n))$  řetězec s a řešení AND nebo NE.

- řekneme, že nedeterministický algoritmus řeší úlohu  $U$ , jestliže:
  - pro každou AND instance 3S, pro který alg. vrátí AND
  - pro každou NE instance neexistuje S, pro který alg. vrátí AND.
- řekneme, že nedeterministický alg. pracuje v čase  $O(T(n))$ , jestliže když přichází oběma fázemi 1 a 2 pro instance velikosti n poté má  $O(T(n))$  kroků.
- když alg. pracuje v  $O(p(n))$ , pak každá z fází pracuje v  $O(p(n))$  a řetězec má polynomickou délku.

Redukce a polynomická redukce úloh

Jsou dány dve rozhodovací úlohy  $U$  a  $V$ . Řekneme, že úloha  $U$  se redukuje na úlohu  $V$ , jestliže existuje algoritmus, který pro každou instance I úlohy  $U$  vytvoří instance I' úlohy  $V$  tak, že I je AND-instance U právě když I' je AND-instance úlohy V.  
→ Značme  $W = U \triangleleft V$

Pokud je nějaký převáděcí algoritmus polynomický, pak se  $U$  polynomická redukuje na  $V$ .  
Značme to  $U \triangleleft_p V$

- když  $U \triangleleft V$ , tak můžeme říct, že  $U$  není složitější než  $V$ .

Tvrzení: Jsou dány tři rozhodovací úlohy  $U, V, W$ . Pak platí:

pokud  $U \triangleleft_p V$  a  $V \triangleleft_p W$ , tak  $U \triangleleft_p W$ .

NP úplné úlohy: Roshodovaná úloha  $U$  je NP úplná (= NP complete = NPC), jestliž:

- 1)  $U$  patří do kategorie NP
- 2) každá NP úloha se polynomickně redukuje na  $U$

Třída NPC je třída všech NP úplných úloh.

Tvrdzení: Pro dané dve NP úlohy  $U$  a  $V$ , pro které platí  $U \leq_p V$ . Pak:

- 1) jestliž  $V$  je v P, pak taky  $U$  je v P
- 2) jestliž  $U$  je NP úplná, pak i  $V$  je NP úplná.

Věta:  $P \subseteq NP$ . Když  $P = NP$  je stále stejná otázka.

Tvrdzení: Když některá NP úplná úloha patřila do kategorie P, pak  $P = NP$  platí!

DK: Když platil předpoklad, tak  $U \in NPC$ , a zároveň  $U \in P$ . Pak následuje, že  $V \leq_p U$ , ale  $U \in P$ , a proto  $V \in P$ .

Cookova věta: SAT je NP úplná úloha.

Dk: 1) SAT je v NP  $\Rightarrow$  generujeme  $2^M$  všech možných pravidelných ohodnocení a pro každé z nich ověříme, jestli je formulé pravidlá, což jde v polynomickém čase.

2) Ukážeme, že každá NP úloha (NTM) se redukuje na SAT. Pro libovolný NTM M a slovo w konstruujeme formulé  $\varphi_{M,w}$ , aby  $\varphi_{M,w}$  byla splnitelná iff M přijímá w. Budu používat následující logické proměnné:

$$S_i^q = \begin{cases} 1 & : v \text{ čase } i \text{ je } M \text{ ve stavu } q \\ 0 & : \text{---} \end{cases} \quad \leftarrow \text{Počet proměnných je } |Q| \cdot (p(n)+1)$$

$$h_{ij} = \begin{cases} 1 & : v \text{ čase } i \text{ je hlava na pozici } j \text{ pásky} \\ 0 & : \text{---} \end{cases} \quad \leftarrow \text{Počet čas } (p(n)) \cdot (p(n)+1)$$

$$t_{ij}^A = \begin{cases} 1 & : v \text{ čase } i \text{ je na pozici } j \text{ pásky symbol } A \\ 0 & : \text{---} \end{cases} \quad \leftarrow |A| \cdot (p(n)) \cdot (p(n)+1)$$

Obecněji:  $S_i^q$ : jakém je stav,  $h_i$ : kde je hlava,  $t_{ij}^A$ : co je na pásku. Počet proměnných je polynom.

i) v každém okamžiku je M pouze v jednom stavu

$$\bigvee S_i^q \wedge \bigwedge_{\substack{q \in Q \\ q \neq q'}} (\neg S_i^q \vee \neg S_i^{q'}) \quad \leftarrow \text{celé v konjunkci pět všech časů}$$

$$\text{přidáme } (p(n)+1) \cdot (|Q| + 2 \binom{|Q|}{2}) \text{ formulé}$$

všechny časy  $\leftarrow$  všechny stavů  $\leftarrow$  minimální dvojice stavů

M je aspoň v jednom stavu  $\leftarrow$  M nemí ve dvou stavech současně. Zde to přepsat jako  $\neg(S_i^q \wedge S_i^{q'})$ .

ii) v každém okamžiku je hlava právě jednoho políčka pásky, stejně jako to máme v 3)

$$\bigvee_{\substack{j=0 \dots p(n) \\ j \neq j'}} h_{ij} \wedge \bigwedge_{\substack{j=0 \dots p(n) \\ j \neq j'}} (\neg h_{ij} \vee \neg h_{ij'}) \quad \leftarrow \text{všechna pole} \leftarrow \text{minimální dvojice políček}$$

všechny časy  $\leftarrow$  všechna pole  $\leftarrow$  všechny symboly

iii) v každém okamžiku obsahuje pole j právě jeden symbol

$$\bigvee S_{ij}^A \wedge \bigwedge_{\substack{A \in \Gamma \\ A \neq A'} } (\neg S_{ij}^A \vee \neg S_{ij'}^A) \quad \leftarrow \text{všechny časy} \leftarrow \text{všechna pole} \leftarrow \text{všechny symboly} \leftarrow \text{minimální dvojice symbolů}$$

formulé je:  $(p(n)+1) \cdot (p(n) + 2 \binom{|A|}{2})$

iv) na každém pole je M v počátečním ID. v čase i=0

předpokládáme, že  $w = a_1, a_2, \dots, a_n$ , pak je počáteční stav:

$$S_{0,0}^q \wedge h_{0,1}^A \wedge S_{0,1}^{a_1} \wedge S_{0,2}^{a_2} \wedge \dots \wedge S_{0,n}^{a_n} \wedge S_{0,n+1}^B \wedge \dots \wedge S_{0,p(n)}^B$$

na každém pásky je volné slovo

slyšet pásky je Blank

$\rightarrow$  hlava je první pozici

$\rightarrow$  M je ve stavu q0

v) každý krok  $M$  je ween přechodovou fù  $S$ , tj stav v rámci i+1 je daný  $S$ .  
 předpokládejme přechodovou fù ve tvaru:  $S(q_i, A) = \{(p_1, C_1, D_1), \dots, (p_k, C_k, D_k)\}$ ,  
 kde  $D = 1$  iff posun je doleva (v opaèném pùipadì je  $D = 0$ ).  
 Krok se formuluje jako implikace: když v rámci i jsem nìkde, v rámci i+1 mìru lìjt n —:  

$$(h_{ij} \wedge A_{ij}^A \wedge s_i^r) \Rightarrow \bigvee_{l=1}^k (S_{i+1}^{q_l}, A_{i+1}^{C_l}, h_{i+1, j+D_l})$$

mìru si vybrat z k závisí se na posledním stavu na poslední pozici (j) napìsnu novým  
 nìkdych mìru posledního mìru

$\forall j \in \text{poèet mìru}$   
 $\forall i \in \text{poèet rámci}$

vi) obecn. polí, které M vyma nìcsl., se mìnemí  $|(p(n)+1) \cdot (p(n)) \cdot (|P| \cdot |Q| \cdot 3 + 3k) \cdot |P| \cdot |Q| |$   
 $(\exists h_{ij} \wedge A_{ij}^A) \Rightarrow \bigwedge_{i,j}^A$  ← tohle celé chceme dítat  $\wedge \wedge (\dots)$   
 kde tu znak A je potom  
 formuli bude  $p(n) \cdot p(n) \cdot (1 + 2|P|)$

vii) je-li slovo pøijato, pak v rámci p(n) je M ve stavu qf

$s_{p(n)}^f$ , a mohli bychom pùdat taky:  $s_i^f \Rightarrow s_{p(n)}^f$ , když do konce minìme deg(j) dílu

### 3) závìr dílcezu

(protože ve všech sedmi bodcích jsou pùdky polynomické mnoho formul, sùstava uloha polynomická)

Pozn: Redukce se nelze dleit, protože tich lehkých je mìlo a ty tìká využití nejaky  
 trika je tìká na mìj se akovost pøijít.

Pří: Pøevléte úlohu SAT na 3-CNF SAT, (lo snamená, že každá klauzule má nejmì 3 liberaøky)

Na vstupe je formula  $\Phi = C_1 \wedge \dots \wedge C_k$ , chceme možné formulí  $\Psi = P_1 \wedge \dots \wedge P_l$  tak, aby  $\Phi$  byla splnitelná iff  $\Psi$  je splnitelná, a tìž platilo, že  $P_i$  je disjunkce nejmì 3 liberaøky. Pro jednoduchosť předpokládejme, že klauzule ve  $\Phi$  jsou seřazeny podle poètu literálù v rozsahu

Postupnì procházejme klauzule ve  $\Phi$ :

- pokud  $C_i$  má 3 nebo mìøe liberaøky, pak  $P_i = C_i$ .

- pokud  $C_i = l_1 \vee l_2 \vee \dots \vee l_r$ ,  $r > 3$ , pak do  $\Psi$  mìmoøe pùdat více klauzule:

$$(l_1 \vee l_2 \vee z_1) \wedge (z_2 \vee l_3 \vee z_2) \wedge (z_2 \vee l_4 \vee z_3) \wedge \dots \wedge (z_{r-3} \vee l_{r-1} \vee l_r)$$

Když bychom udìlali resolventu poèelle z, dostaneme sase  $C_i$ . Oproti  $C_i$  jsme pùdali r-3 mìøejších liberaøek.

Cvièení Nauònite TM  $M_1$ , který pøijímá jazyk  $L_1 = \{aibick^k, ijk \geq 1\}$

3.4.2019

$$Q = \{0, 1, 2, 3, F\}$$

$$\Sigma = \{a, b, c\}$$

$$\Gamma = \{a, b, c, B\}$$



Ukáæte práci nad slovem  $w = aabcc$

$aabcc \rightarrow a1bcc \rightarrow aab2cc \rightarrow aabc3c \rightarrow aabcc3B \rightarrow aabcFc$  správné deðátko

Spùtejte  $T(n)$  a  $S(n)$

$T(n)$ : udelá pøesnì  $n+1$  krokù, tj  $T(n) \in \Theta(n)$

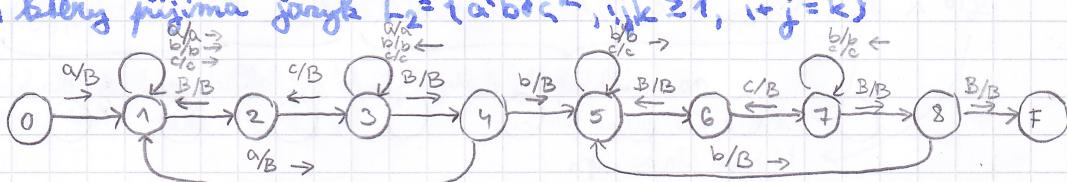
$S(n)$ : mìnslivù  $n+2$  polí, tj  $S(n) \in \Theta(n)$

Nauònite TM  $M_2$ , který pøijímá jazyk  $L_2 = \{aibick^k, ijk \geq 1, i+j=k\}$

$$Q = \{0, \dots, 8, F\}$$

$$\Sigma = \{a, b, c\}$$

$$\Gamma = \{a, b, c, B\}$$



Ukáæte práci nad slovem  $w = aabccc$

$aabccc \rightarrow 1abccc \rightarrow abcc2c \rightarrow abc3c \rightarrow 4abcc \rightarrow 1bcc \rightarrow * bc2c \rightarrow b3c \rightarrow$

$$1 \leftarrow 3bc \leftarrow 3Bbc \leftarrow 4bc \leftarrow 5c \leftarrow c5B \leftarrow 6c \leftarrow 7B \leftarrow 8B \leftarrow FB \text{ následně se rozstaví}$$

### Princip fungování M<sub>2</sub>:

Uděleme slovo, na kterém máme ráz významy jehož a, pak přejedeme na konec a smažeme jeho c. Jakmile jsou všechna a smazána, pak máme se ráz významu b a s konec c.

$0 \rightarrow 1$  (připravíme  $4 \rightarrow 1$ ): smažeme a na ráz významu

1 → 2 vrážíme hlavní na poslední znak

2 → 3 smažeme c na konec slova

3 → 4 vrážíme hlavní se na ráz významu

4 → 5 vrážíme hlavní na první znak

Pokud jsou v této chvíli na ráz významu slova nějaká a, uděleme po hraně  $4 \rightarrow 1$  a opakuje se proces.

Pokud však ještě jsou jen b, přesuneme se do cyklu 5678, kdežto pravý slouží jeho 1234, ale se znaky b. Na konci musí být páška prázdná, tj. odmasazali jsme zbytečné znaky atd. jako znaky c.

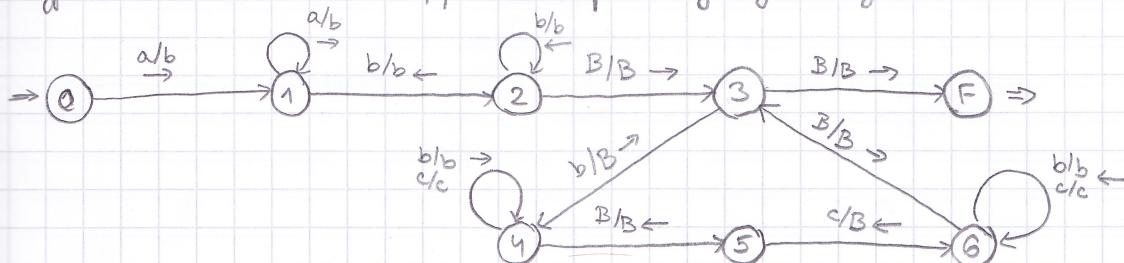
### Vypočítejte T(n) a S(n)

T(n): dvoj projížďek řetězem, a před každým projížděním smaže jeden znak. Celková délka  $n + (n-1) + (n-2) \dots + 2 + 1 + k$  kroků, kde k je konstantní počet kroků na dojedou do cíle. Celkem  $T(n) \in O(n^2)$

S(n): nic se nikan nekopíruje, takže  $S(n) \in O(n)$ .

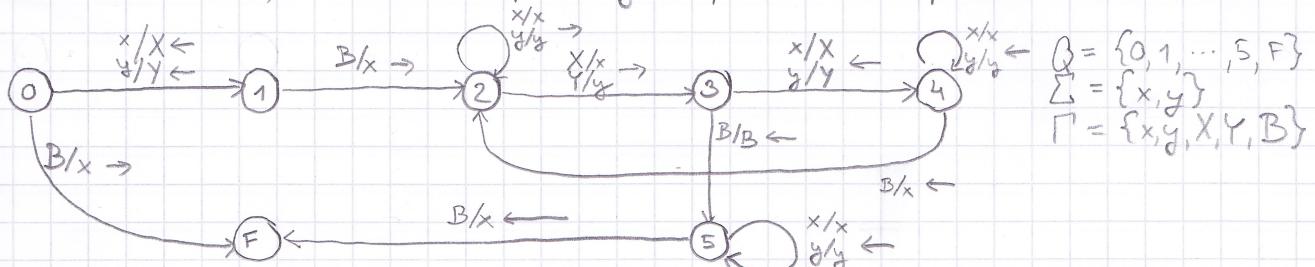
### Alternativní návrh: (na cíli se nekonverguje)

Nejdříve nahradí znak a za b, pak bude potřeba jen jeden cyklus na celou část  $\Rightarrow$  menší návrat.

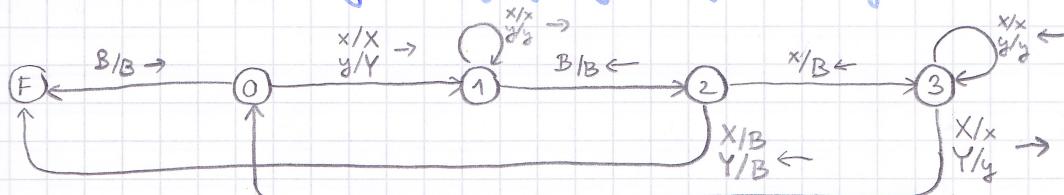


Naučme se TM M<sub>3</sub>, který realizuje f:  $\{x,y\}^* \rightarrow \{x,y\}^*$ :  $w \mapsto x^{k+1}w$ , kde  $k = |w|$ .

Nejdříve nahradíme první znak za cílovou významou, pak na ráz významu nahradíme znak. Pak budeme projíždět dál a dál do slova, nalevo budeme psát myly a napravo budeme posouvat známkou a na konec.



Naučme se TM M<sub>4</sub>, který realizuje f:  $\{x,y\}^* \rightarrow \{x,y\}^*$ :  $a_1a_2 \dots a_n \mapsto a_1a_2 \dots a_{\lfloor \frac{n}{2} \rfloor}$



Díl pro zadání: naučme se TM M<sub>5</sub>, který generuje  $L_5 = \{a_i b a_k, i, k \geq 1, i, j = k\}$

a a a a b b c c c c c c c c

a a a a c c c c c c c c b b

a a a c c c c c c b b

a a c c c c b b

a c c b b

b b

Princip: podslово b je písmen na konec slova. Pokud budu z jedné strany ulíhat po jednom (aleva) a na druhé straně budu přesouvat krajní znaky. Na konci mi mohu uvolnit podslovo bt.