

(Výpočetní) problémy, rozhodovací (ANO/NE) problémy

(Výpočetní) problémy, rozhodovací (ANO/NE) problémy

Název: Zdvojení slova v abecedě $\{a, b\}$.

Vstup (Instance): $w \in \{a, b\}^*$.

Výstup: ww .

(Výpočetní) problémy, rozhodovací (ANO/NE) problémy

Název: Zdvojení slova v abecedě $\{a, b\}$.

Vstup (Instance): $w \in \{a, b\}^*$.

Výstup: ww .

Název: Příslušnost k jazyku $L = \{a^n b^n c^n \mid n \geq 1\}$.

Vstup: $w \in \{a, b, c\}^*$.

Výstup: ANO, když $w \in L$, NE jinak.

(Výpočetní) problémy, rozhodovací (ANO/NE) problémy

Název: Zdvojení slova v abecedě $\{a, b\}$.

Vstup (Instance): $w \in \{a, b\}^*$.

Výstup: ww .

Název: Příslušnost k jazyku $L = \{a^n b^n c^n \mid n \geq 1\}$.

Vstup: $w \in \{a, b, c\}^*$.

Výstup: ANO, když $w \in L$, NE jinak.

Název: Příslušnost k jazyku $L = \{a^n b^n c^n \mid n \geq 1\}$.

Vstup: $w \in \{a, b, c\}^*$.

Otázka: je $w \in L$?

(Výpočetní) problémy, rozhodovací (ANO/NE) problémy

Název: Zdvojení slova v abecedě $\{a, b\}$.

Vstup (Instance): $w \in \{a, b\}^*$.

Výstup: ww .

Název: Příslušnost k jazyku $L = \{a^n b^n c^n \mid n \geq 1\}$.

Vstup: $w \in \{a, b, c\}^*$.

Výstup: ANO, když $w \in L$, NE jinak.

Název: Příslušnost k jazyku $L = \{a^n b^n c^n \mid n \geq 1\}$.

Vstup: $w \in \{a, b, c\}^*$.

Otázka: je $w \in L$?

Název: Ekvivalence bezkontextových gramatik.

Vstup: dvě BG G_1, G_2 .

Otázka: je $L(G_1) = L(G_2)$?

$$M = (Q, \Sigma, \Gamma, \delta, q_0, F)$$

$\Sigma \subseteq \Gamma$; $\Gamma \setminus \Sigma$ vždy obsahuje speciální (prázdný) symbol $\square$

$$\delta : (Q - F) \times \Gamma \rightarrow Q \times \Gamma \times \{-1, 0, +1\}$$

(základní varianta je tedy *deterministický* Turingův stroj)

Příklad TS (tedy “programu”, de facto množiny instrukcí), rozhodujícího příslušnost k jazyku $\{a^n b^n c^n \mid n \geq 1\}$:

$$M = (Q, \Sigma, \Gamma, \delta, q_0, F)$$

$\Sigma \subseteq \Gamma$; $\Gamma \setminus \Sigma$ vždy obsahuje speciální (prázdný) symbol $\square$

$$\delta : (Q - F) \times \Gamma \rightarrow Q \times \Gamma \times \{-1, 0, +1\}$$

(základní varianta je tedy *deterministický* Turingův stroj)

Příklad TS (tedy “programu”, de facto množiny instrukcí), rozhodujícího příslušnost k jazyku $\{a^n b^n c^n \mid n \geq 1\}$:

$$(q_0, a) \rightarrow (q_1, \bar{a}, +1)$$

$$(q_1, x) \rightarrow (q_1, x, +1) \text{ pro } x \in \{a, \bar{b}\}$$

$$(q_1, b) \rightarrow (q_2, \bar{b}, +1)$$

...

$$M = (Q, \Sigma, \Gamma, \delta, q_0, F)$$

$\Sigma \subseteq \Gamma$; $\Gamma \setminus \Sigma$ vždy obsahuje speciální (prázdný) symbol $\square$

$$\delta : (Q - F) \times \Gamma \rightarrow Q \times \Gamma \times \{-1, 0, +1\}$$

(základní varianta je tedy *deterministický* Turingův stroj)

Příklad TS (tedy “programu”, de facto množiny instrukcí), rozhodujícího příslušnost k jazyku $\{a^n b^n c^n \mid n \geq 1\}$:

$$\begin{aligned}(q_0, a) &\rightarrow (q_1, \bar{a}, +1) \\(q_1, x) &\rightarrow (q_1, x, +1) \text{ pro } x \in \{a, \bar{b}\} \\(q_1, b) &\rightarrow (q_2, \bar{b}, +1) \\&\dots\end{aligned}$$

výpočet = (legální) posloupnost konfigurací ...

$$(q_0 a a a b b b c c c) \vdash (\bar{a} q_1 a a b b b c c c) \vdash (\bar{a} a q_1 a b b b c c c) \vdash (\bar{a} a a q_1 b b b c c c) \dots$$

$M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ definuje částečné zobrazení $f_M : \Sigma^* \rightarrow \Gamma^*$.

Problémy řešitelné Turingovými stroji

I/O tabulka Turingova stroje M

(zobrazení f_M)

Tabulka problému P (výstup pro w je nedefinován, označeno $\perp$,
když výpočet M pro w je nekonečný)

Vstup	Příslušný defin. výstup	Vstup	Výstup výpočtu nebo $\perp$
ε	ε	ε	
0	00	0	
1	11	1	
00	0000	00	
01	0101	01	
10	1010	10	
11	1111	11	
000	000000	000	
...	...	...	...

Problémy řešitelné Turingovými stroji

I/O tabulka Turingova stroje M

(zobrazení f_M)

Tabulka problému P (výstup pro w je nedefinován, označeno $\perp$,
když výpočet M pro w je nekonečný)

Vstup	Příslušný defin. výstup	Vstup	Výstup výpočtu nebo $\perp$
ε	ε	ε	
0	00	0	
1	11	1	
00	0000	00	
01	0101	01	
10	1010	10	
11	1111	11	
000	000000	000	
...	...	...	...

Problém P je “turingovsky” řešitelný, jestliže existuje TS M , jehož I/O tabulka se rovná tabulce P (stroj M ke každému vstupu vydá výstup specifikovaný problémem P)

Rozhodovací (ANO/NE) problémy

Rozhodovací (ANO/NE) problémy

ANO/NE problém je rozhodnutelný, jestliže

Rozhodovací (ANO/NE) problémy

ANO/NE problém je rozhodnutelný, jestliže existuje Turingův stroj, který ho rozhoduje (I/O tabulka [I/O zobrazení] stroje se rovná tabulce [funkci] specifikované problémem).

Rozhodovací (ANO/NE) problémy

ANO/NE problém je rozhodnutelný, jestliže existuje Turingův stroj, který ho rozhoduje (I/O tabulka [I/O zobrazení] stroje se rovná tabulce [funkci] specifikované problémem).

Název: Sedmičky v Ludolfově čísle π

Vstup: přirozené číslo k

Otázka: Existuje v desetinném rozvoji čísla π úsek sedmiček délky k ?

Rozhodovací (ANO/NE) problémy

ANO/NE problém je rozhodnutelný, jestliže existuje Turingův stroj, který ho rozhoduje (I/O tabulka [I/O zobrazení] stroje se rovná tabulce [funkci] specifikované problémem).

Název: Sedmičky v Ludolfově čísle π

Vstup: přirozené číslo k

Otázka: Existuje v desetinném rozvoji čísla π úsek sedmiček délky k ?

Vstup	Výstup
0	ANO
1	ANO
2	ANO
3	ANO
4	ANO
5	ANO
...	...

Rozhodovací (ANO/NE) problémy

ANO/NE problém je rozhodnutelný, jestliže existuje Turingův stroj, který ho rozhoduje (I/O tabulka [I/O zobrazení] stroje se rovná tabulce [funkci] specifikované problémem).

Název: Sedmičky v Ludolfově čísle π

Vstup: přirozené číslo k

Otázka: Existuje v desetinném rozvoji čísla π úsek sedmiček délky k ?

Vstup	Výstup	Vstup	Výstup
0	ANO	0	ANO
1	ANO	1	NE
2	ANO	2	NE
3	ANO	3	NE
4	ANO	4	NE
5	ANO	5	NE
...	...	...	...

Rozhodovací (ANO/NE) problémy

ANO/NE problém je rozhodnutelný, jestliže existuje Turingův stroj, který ho rozhoduje (I/O tabulka [I/O zobrazení] stroje se rovná tabulce [funkci] specifikované problémem).

Název: Sedmičky v Ludolfově čísle π

Vstup: přirozené číslo k

Otázka: Existuje v desetinném rozvoji čísla π úsek sedmiček délky k ?

Vstup	Výstup	Vstup	Výstup	Vstup	Výstup
0	ANO	0	ANO	0	ANO
1	ANO	1	NE	1	ANO
2	ANO	2	NE	2	NE
3	ANO	3	NE	3	NE
4	ANO	4	NE	4	NE
5	ANO	5	NE	5	NE
...	...	...	...	...	...

Rozhodovací (ANO/NE) problémy

ANO/NE problém je rozhodnutelný, jestliže existuje Turingův stroj, který ho rozhoduje (I/O tabulka [I/O zobrazení] stroje se rovná tabulce [funkci] specifikované problémem).

Název: Sedmičky v Ludolfově čísle π

Vstup: přirozené číslo k

Otázka: Existuje v desetinném rozvoji čísla π úsek sedmiček délky k ?

Vstup	Výstup	Vstup	Výstup	Vstup	Výstup	Vstup	Výstup
0	ANO	0	ANO	0	ANO	0	ANO
1	ANO	1	NE	1	ANO	1	ANO
2	ANO	2	NE	2	NE	2	ANO
3	ANO	3	NE	3	NE	3	NE
4	ANO	4	NE	4	NE	4	NE
5	ANO	5	NE	5	NE	5	NE
...	...	...	...	...	...	...	...

Rozhodovací (ANO/NE) problémy

ANO/NE problém je rozhodnutelný, jestliže existuje Turingův stroj, který ho rozhoduje (I/O tabulka [I/O zobrazení] stroje se rovná tabulce [funkci] specifikované problémem).

Název: Sedmičky v Ludolfově čísle π

Vstup: přirozené číslo k

Otázka: Existuje v desetinném rozvoji čísla π úsek sedmiček délky k ?

Vstup	Výstup	Vstup	Výstup	Vstup	Výstup	Vstup	Výstup	..
0	ANO	0	ANO	0	ANO	0	ANO	
1	ANO	1	NE	1	ANO	1	ANO	
2	ANO	2	NE	2	NE	2	ANO	
3	ANO	3	NE	3	NE	3	NE	
4	ANO	4	NE	4	NE	4	NE	
5	ANO	5	NE	5	NE	5	NE	
...	...	...	...	...	...	...	...	

Vidíme, že problém je rozhodnutelný, i když neznáme algoritmus ...

Algoritmus = Turingův stroj

Algoritmus = Turingův stroj

algoritmická neřešitelnost

(algoritmická) nerozhodnutelnost

Simulace mezi variantami Turingových strojů

Dvoupáskový Turingův stroj (2P-TS) $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$, kde

$$\delta : (Q - F) \times \Gamma \times \Gamma \rightarrow Q \times \Gamma \times \{-1, 0, +1\} \times \Gamma \times \{-1, 0, +1\},$$

Příklad (zdvojení slova):

Simulace mezi variantami Turingových strojů

Dvoupáskový Turingův stroj (2P-TS) $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$, kde

$$\delta : (Q - F) \times \Gamma \times \Gamma \rightarrow Q \times \Gamma \times \{-1, 0, +1\} \times \Gamma \times \{-1, 0, +1\},$$

Příklad (zdvojení slova):

$$(q_0, x, \square) \rightarrow (q_0, x, +1, x, +1) \text{ pro } x \in \{a, b\}$$

$$(q_0, \square, \square) \rightarrow (q_1, \square, -1, \square, 0)$$

$$(q_1, x, \square) \rightarrow (q_1, x, -1, \square, 0) \text{ pro } x \in \{a, b\}$$

$$(q_1, \square, \square) \rightarrow (q_2, \square, +1, \square, 0)$$

$$(q_2, x, \square) \rightarrow (q_2, x, +1, x, +1) \text{ pro } x \in \{a, b\}$$

$$(q_2, \square, \square) \rightarrow (q_{halt}, \square, 0, \square, 0)$$

Simulace mezi variantami Turingových strojů

Dvoupáskový Turingův stroj (2P-TS) $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$, kde

$$\delta : (Q - F) \times \Gamma \times \Gamma \rightarrow Q \times \Gamma \times \{-1, 0, +1\} \times \Gamma \times \{-1, 0, +1\},$$

Příklad (zdvojení slova):

$$(q_0, x, \square) \rightarrow (q_0, x, +1, x, +1) \text{ pro } x \in \{a, b\}$$

$$(q_0, \square, \square) \rightarrow (q_1, \square, -1, \square, 0)$$

$$(q_1, x, \square) \rightarrow (q_1, x, -1, \square, 0) \text{ pro } x \in \{a, b\}$$

$$(q_1, \square, \square) \rightarrow (q_2, \square, +1, \square, 0)$$

$$(q_2, x, \square) \rightarrow (q_2, x, +1, x, +1) \text{ pro } x \in \{a, b\}$$

$$(q_2, \square, \square) \rightarrow (q_{halt}, \square, 0, \square, 0)$$

Obecný dvoupáskový Turingův stroj (2P-TS) M lze simulovat jednopáskovým dvouhlavým Turingovým strojem (1P-2H-TS) M' . Stroje M , M' mají tedy stejné (*vstupně/výstupní*) *chování*, tj. realizují tutéž (částečnou) funkci $f_M = f_{M'}$ (mají tutéž I/O tabulku).

Simulace mezi variantami Turingových strojů

Dvoupáskový Turingův stroj (2P-TS) $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$, kde

$$\delta : (Q - F) \times \Gamma \times \Gamma \rightarrow Q \times \Gamma \times \{-1, 0, +1\} \times \Gamma \times \{-1, 0, +1\},$$

Příklad (zdvojení slova):

$$(q_0, x, \square) \rightarrow (q_0, x, +1, x, +1) \text{ pro } x \in \{a, b\}$$

$$(q_0, \square, \square) \rightarrow (q_1, \square, -1, \square, 0)$$

$$(q_1, x, \square) \rightarrow (q_1, x, -1, \square, 0) \text{ pro } x \in \{a, b\}$$

$$(q_1, \square, \square) \rightarrow (q_2, \square, +1, \square, 0)$$

$$(q_2, x, \square) \rightarrow (q_2, x, +1, x, +1) \text{ pro } x \in \{a, b\}$$

$$(q_2, \square, \square) \rightarrow (q_{halt}, \square, 0, \square, 0)$$

Obecný dvoupáskový Turingův stroj (2P-TS) M lze simulovat jednopáskovým dvouhlavým Turingovým strojem (1P-2H-TS) M' . Stroje M , M' mají tedy stejné (*vstupně/výstupní*) *chování*, tj. realizují tutéž (částečnou) funkci $f_M = f_{M'}$ (mají tutéž I/O tabulku).

Obecný 1P-2H-TS M' lze simulovat standardním strojem, tedy jednopáskovým jednohlavým Turingovým strojem.

Bez ztráty obecnosti se můžeme omezit na Turingovy stroje s abecedami

$$\Sigma = \{0, 1\},$$

$$\Gamma = \{0, 1, \square\},$$

jejichž I/O zobrazení je typu

$$\{0, 1\}^* \rightarrow \{0, 1\}^* \cup \{\perp\}$$

Název: Halting Problem

Vstup: Turingův stroj M , vstupní slovo w

Otázka: Zastaví se (výpočet stroje) M , když začne pracovat na (vstupním) slově w ?

Název: Halting Problem

Vstup: Turingův stroj M , vstupní slovo w

Otázka: Zastaví se (výpočet stroje) M , když začne pracovat na (vstupním) slově w ?

Název: Diagonal Halting Problem

Vstup: Turingův stroj M

Otázka: Zastaví se (výpočet stroje) M , když začne pracovat na (vstupním) slově, které je kódem M ?

(ANO/NE) problém P_1 je *algoritmicky převeditelný* (stručněji *převeditelný*) na problém P_2 , což označujeme $P_1 \rightsquigarrow P_2$,
jestliže

(ANO/NE) problém P_1 je *algoritmicky převeditelný* (stručněji *převeditelný*) na problém P_2 , což označujeme $P_1 \rightsquigarrow P_2$, jestliže existuje algoritmus, který k instanci I_1 problému P_1 sestrojí instanci I_2 problému P_2 tak, že odpověď na otázku pro I_1 v P_1 je stejná jako odpověď na otázku pro I_2 v P_2 .

(ANO/NE) problém P_1 je *algoritmicky převeditelný* (stručněji *převeditelný*) na problém P_2 , což označujeme $P_1 \rightsquigarrow P_2$, jestliže existuje algoritmus, který k instanci I_1 problému P_1 sestrojí instanci I_2 problému P_2 tak, že odpověď na otázku pro I_1 v P_1 je stejná jako odpověď na otázku pro I_2 v P_2 .

např. DHP je převeditelný na HP ($\text{DHP} \rightsquigarrow \text{HP}$) ...

(ANO/NE) problém P_1 je *algoritmicky převeditelný* (stručněji *převeditelný*) na problém P_2 , což označujeme $P_1 \rightsquigarrow P_2$, jestliže existuje algoritmus, který k instanci I_1 problému P_1 sestrojí instanci I_2 problému P_2 tak, že odpověď na otázku pro I_1 v P_1 je stejná jako odpověď na otázku pro I_2 v P_2 .

např. DHP je převeditelný na HP ($\text{DHP} \rightsquigarrow \text{HP}$) ...

Tvrzení:

Když P_1 je nerozhodnutelný a $P_1 \rightsquigarrow P_2$, tak P_2 je nerozhodnutelný.