

Teoretická informatika TIN - 2024/2025

2. Test 8. 11. 2024

Čas na řešení: 120 minut

(max. zisk 15 bodů – 10 bodů níže odpovídá 1 bodu v hodnocení předmětu)

Jméno/přihlašovací jméno:

Hodnocení:

--	--	--	--	--

Poznámka: Pokud při vypracování zkoušky použijete jinou notaci a konvence, než byly zavedeny na přednáškách, je nutné takovou notaci popsat. Písemnou zkoušku zpracujte čitelně a úhledně.

Příklad 1
50 bodů

- a) Necht' $Z = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je zásobníkový automat (ZA). Formálně definujte konfiguraci ZA (popište význam jednotlivých položek konfigurace) a relaci přechodu mezi konfiguracemi.
- b) Necht' $\Sigma = \{a, b, c\}$. Sestrojte (rozšířený) zásobníkový automat pro jazyk $L = \{w \in \Sigma^* \mid \forall w' \in \Sigma^* : w' \text{ je prefix slova } w \implies \#_a(w') \leq \#_b(w') + \#_c(w')\}$, kde $\#_x(w)$ značí počet znaků $x \in \{a, b, c\}$ v řetězci $w \in \Sigma^*$. Automat popište přechodovým diagramem.
- c) Necht' $G = (N, \Sigma, P, S)$ je bezkontextová gramatika, kde $\Sigma = \{a, b, c\}$. Navrhněte a formálně запиšte algoritmus, který pro gramatiku G spočítá množinu

$$N_\diamond = \{A \in N \mid A \Rightarrow_G^+ w \wedge w \in \Sigma^+ \wedge w \text{ nezačíná písmenem } c\}.$$

V algoritmu můžete využít množiny

- $N_t = \{A \in N \mid A \Rightarrow_G^+ w \wedge w \in \Sigma^*\},$
- $N_\epsilon = \{A \in N \mid A \Rightarrow_G^+ \epsilon\}.$

Ilustrujte použití algoritmu na příkladu gramatiky s pravidly:

$$S \rightarrow VcZ \quad U \rightarrow Vc \mid c \quad V \rightarrow WaUa \quad W \rightarrow cW \mid \epsilon \quad Z \rightarrow aZ$$

TIN — 2. Test 8. 11. 2024

Čas: 120 minut

Prostor pro řešení příkladu 1.

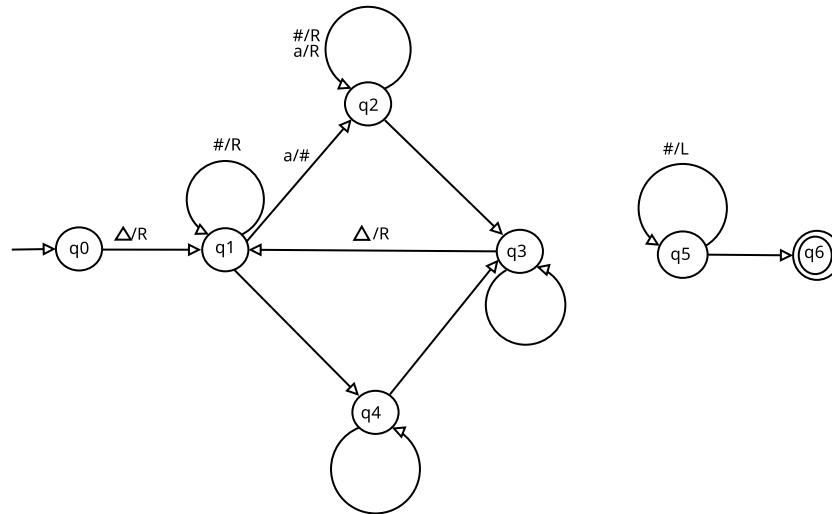
- a) Necht' $\Sigma = \{a, b\}$. Do níže uvedeného diagramu doplňte 1 nový přechod (hranu i její označení) a doplňte označení k neoznačeným přechodům tak, aby vznikl *deterministický* Turingův stroj T , který přijímá jazyk

Příklad 2
40 bodů

$$L_1 = \{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w) + 1\},$$

kde $\#_x(w)$ značí počet symbolů $x \in \{a, b\}$ v řetězci $w \in \{a, b\}^*$.

Poznámka: V rámci řešení je dovoleno použít „vícenásobné“ přechody, jako již jsou uvedeny např. ze stavu q_2 .



- b) Dokažte, že jazyk L_2 je rekurzivně vyčíslitelný.

$$L_2 = \{ \langle M_1 \rangle \# \langle M_2 \rangle \mid L(M_1) \cdot L(M_2) \neq \emptyset \},$$

kde $\langle M \rangle$ značí kód **Turingova stroje** M .

Poznámka: V důkazu stačí uvést slovní popis fungování Turingova stroje (deterministického či nedeterministického, dle libosti s více páskami), který akceptuje jazyk L_2 .

- c) Necht' $\Sigma = \{a, b\}$. Rozhodněte a dokažte, zda existuje *lineárně ohraničený* automat M takový, že

$$L(M) \neq \emptyset \wedge \text{množina } \{w \in \Sigma^* \mid M \text{ na } w \text{ cyklí}\} \text{ je nekonečná.}$$

TIN — 2. Test 8. 11. 2024

Čas: 120 minut

Prostor pro řešení příkladu 2.

a) Formálně definujte i) *časovou složitost* deterministického Turingova stroje, který přijímá jazyk L a ii) *třídu jazyků* $DTIME[2^n]$. **Příklad 3**
60 bodů

b) Předpokládejte, že $\mathbf{P} \neq \mathbf{NP}$. Uveďte (tj. formálně запиšte) příklady následujících jazyků (příslušnost do požadované třídy jazyků nemusíte dokazovat):

1. Rekurzivní jazyk $L_1 \notin \mathbf{P}$.
2. $L_2 \in DTIME[1]$.
3. $L_3 \in \mathbf{P} \setminus DSPACE[1]$.

Poznámka: U třídy $DSPACE[1]$ nepočítáme buňky pásky, na nichž je zapsán vstup za předpokladu, že obsah těchto buněk není modifikován.

c) Uvažme níže uvedený algoritmus $merge(A, B)$, který má na vstupu dvě seřazená pole A a B , obě o velikosti $n/2$ (n je sudé). $merge(A, B)$ vrací seřazené pole C , které vzniklo spojením polí A a B (tj. standardní operací *merge*).

1. Analyzujte asymptotickou časovou složitost algoritmu $merge(A, B)$ v nejhorším případě.

2. Analyzujte asymptotickou časovou složitost algoritmu $merge(A, B)$ v nejlepším případě.

3. Navrhněte algoritmus, který řeší stejný problém a má lepší asymptotickou časovou složitost v nejhorším případě. Analyzujte tuto složitost.

```

1 Function merge( $A, B$ )
2    $C := \text{new int}[n]$ 
3   for  $\text{int } i := 0; i < n/2; i := i + 1$  do
4      $C[i] := A[i]$ 
5    $\text{size} := n/2$ 
6   for  $\text{int } i := 0; i < n/2; i := i + 1$  do
7      $j := 0$ 
8     while  $j < \text{size} \wedge C[j] < B[i]$  do
9        $j := j + 1$ 
10     $\text{tmp} := B[i]$ 
11    while  $j \leq \text{size}$  do
12      swap( $\text{tmp}, C[j]$ )
13       $j := j + 1$ 
14     $\text{size} := \text{size} + 1$ 
15  return  $C$ 
```

Poznámka: Předpokládejte uniformní cenové kritérium, kde složitost každého řádku programu je 1.

TIN — 2. Test 8. 11. 2024

Čas: 120 minut

Prostor pro řešení příkladu 3.

TIN — 2. Test 8. 11. 2024

Čas: 120 minut

Prostor na řešení.

TIN — 2. Test 8. 11. 2024

Čas: 120 minut

Prostor na řešení.
