

Teoretická informatika TIN - 2024/2025
Závěrečná zkouška (1. opravný termín) 16. 1. 2025
Čas na řešení: 220 minut

(max. zisk 60 bodů – 10 bodů níže odpovídá 1 bodu v hodnocení předmětu)

Jméno/přihlašovací jméno:

--	--	--	--

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.

K danému příkladu využijte důsledně jemu vymezený prostor.

Příklad 1
120 bodů
min. 30 bodů

- 1a) Formálně zapište *Pumping lemma pro regulární jazyky*. Dále uveďte hlavní myšlenku jejího důkazu (stačí 2–3 věty bez technických detailů důkazu).

(30 bodů)

- 1b) Rozhodněte a dokažte, zda jsou následující jazyky regulární:

a) $L_1 = \{w \in \{a, b, c\}^* \mid (\#_a(w) = \#_b(w) + \#_c(w)) \wedge \#_a(w) > 2\},$

b) $L_2 = \{w \in \{a, b, c\}^* \mid (\#_a(w) = \#_b(w) + \#_c(w)) \wedge \#_a(w) < 2\},$

kde $\#_x(w)$ značí počet znaků x v řetězci w .

Poznámka: Při dokazování, že je jazyk regulární, stačí uvést odpovídající gramatiku či automat. Při dokazování, že jazyk není regulární, použijte Pumping Lemma nebo Myhill-Nerodovu větu.

(50 bodů)

- 1c) Uvažme $\Sigma = \{a, b\}$. Rozhodněte a dokažte, zda platí následující tvrzení:

$$\forall L_1, L_2 \in \mathcal{L}_{fin} : |L_1| \leq |L_2| \Rightarrow |\sim_{L_1}| \leq |\sim_{L_2}|$$

kde $\mathcal{L}_{fin}$ značí třídu konečných jazyků, $\sim_L$ značí relaci prefixové ekvivalence dané jazykem L a $|\sim|$ značí index relace $|\sim|$ (tj. počet tříd rozkladu $\Sigma^* / \sim$).

(40 bodů)

Prostor pro řešení Příkladu 1.

Příklad 2
120 bodů
min. 30 bodů

- 2a) Necht $A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je *nedeterministický zásobníkový automat* (NZA). Formálně definujte i) *přechodovou funkci* δ , ii) *konfiguraci* NZA (popište význam jednotlivých položek konfigurace) a iii) *relaci přechodu* mezi konfiguracemi.

Dále rozhodněte a dokažte (stačí hlavní myšlenka důkazu v rozsahu 2–3 vět), zda je *problém konečnosti jazyka* daného NZA A (tj. zda $L(A)$ je konečný jazyk) *rozhodnutelný*.

(40 bodů)

- 2b) Navrhněte a formálně запиšte algoritmus, který pro gramatiku G spočítá množinu

$$N_a = \{A \in N \mid A \Rightarrow_G^+ w \wedge w \in \Sigma^* \wedge \#_a(w) > 0\},$$

kde $\#_a(w)$ značí počet symbolů a v řetězci $w \in \Sigma^*$. V algoritmu můžete využít množiny:

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

Ilustrujte použití algoritmu na příkladu gramatiky s pravidly

$$S \rightarrow ABX \mid Sb \mid b \quad X \rightarrow XB \mid Xa \quad A \rightarrow BB \mid b \quad B \rightarrow Aa .$$

(50 bodů)

- 2c) Uvažme abecedu $\Sigma = \{a, b\}$. Rozhodněte a dokažte, zda existují neregulární bezkontextové jazyky L_1 a L_2 takové, že $L_1 \setminus L_2$ je nekonečný regulární jazyk. (30 bodů)

(30 bodů)

Prostor pro řešení Příkladu 2.

Příklad 3
180 bodů
min. 50 bodů

- 3a) Formálně definujte *kontextovou gramatiku* (KG) a *relaci přímé derivace* v dané KG a *jazyk generovaný* danou KG.

Dále rozhodněte a uveďte hlavní myšlenku důkazu (v rozsahu 2–3 vět), zda je *problém příslušnosti do jazyka kontextových gramatik rozhodnutelný* (tj. problém zda $w \in L(G)$ pro danou KG G). (50 bodů)

(50 bodů)

- 3b) Uvažme jazyk

$$L = \{ \langle M_1 \rangle \# \langle M_2 \rangle \mid M_1 \text{ a } M_2 \text{ jsou Turingovy stroje, pro které existuje } w \in \{a, b\}^* \text{ takové, že } w \in L(M_1) \cap L(M_2) \text{ a } M_1 \text{ akceptuje } w \text{ v menším počtu kroků než } M_2 \}.$$

Rozhodněte a dokažte zda:

- i) L je rekurzivní,
- ii) L je rekurzivně vyčíslitelný.

(80 bodů)

- 3c) Mějme teorii T se signaturou $\langle \{Pat_{/0}, Mat_{/0}\}, \{Clovek_{/1}, Vec_{/1}, rozbil_{/2}, =_{/2}\} \rangle$ (= je standardní rovnost) se speciálními axiomy

$$\begin{aligned} \forall x \forall y (rozbil(x, y) \rightarrow (Clovek(x) \wedge Vec(y))) \\ \forall x (Vec(x) \rightarrow \exists y (rozbil(y, x))) \\ \forall x \neg Clovek(x) \vee \neg Vec(x) \end{aligned}$$

- i) Rozhodněte a dokažte, zda T je *bezesporná* a *úplná*.
- ii) Uvažujme formuli $\phi \equiv \exists x rozbil(Pat, x)$. Sestavte interpretaci I takovou, že $I \models T$ a $I \models \phi$, nebo zdůvodněte proč neexistuje.
- iii) Rozhodněte a zdůvodněte, zda $T \models \phi$.

(50 bodů)

Prostor pro řešení Příkladu 3.

Příklad 4
180 bodů
min. 50 bodů

- 4a) Formálně definujte i) *prostorovou složitost* deterministického Turingova stroje, který přijímá jazyk L a ii) třídu $DSPACE[1]$. Dále definujte *SAT problém* (tj. problém splnitelnosti booleovských formulí) a vysvětlete (v rozsahu 1–2 vět) jeho důležitost v informatice.

Dále rozhodněte a zdůvodněte, zda platí následující tvrzení (předpokládejme, že $\mathbf{P} \neq \mathbf{NP}$):

- i) Existuje jazyk L_1 , který je **NP**-úplný a nepatří do třídy **EXP**.
- ii) Existuje jazyk L_2 , který patří do třídy **EXP**, ale není **NP**-úplný.
- iii) Existuje jazyk $L_3 \in DTIME[1] \setminus DSPACE[1]$.

(70 bodů)

- 4b) Nechtě A, B, C jsou pole o velikosti n (indexované od 1), která obsahují přirozená čísla. Uvažme následující rozhodovací problém:

$$P(A, B, C, n) = \text{true} \Leftrightarrow \exists i \in \{1, \dots, n\} : \forall j, k \in \{1, \dots, n\} : A[j] \geq B[i] \geq C[k]$$

Níže uvedený algoritmus $middle(A, B, C, n)$ řeší tento problém.

- (a) Analyzujte a zdůvodněte asymptotickou časovou složitost algoritmu $middle$ v nejhorším případě.
- (b) Analyzujte a zdůvodněte asymptotickou časovou složitost algoritmu $middle$ v nejlepším případě.
- (c) Navrhněte algoritmus $middle^+(A, B, C, n)$, který řeší tento problém optimálně vzhledem k asymptotické časové složitosti v nejhorším případě. Analyzujte a zdůvodněte tuto složitost.

(60 bodů)

```
1 Function middle( $A, B, C, n$ )
2    $found := false$ 
3    $i := 1$ 
4   while  $i \leq n \wedge found = false$  do
5      $found := true$ 
6      $j := 1$ 
7      $k := 1$ 
8     while  $j \leq n \wedge found = true$  do
9       if  $A[j] < B[i]$  then
10         $found := false$ 
11         $j := j + 1$ 
12     while  $k \leq n \wedge found = true$  do
13       if  $B[i] < C[k]$  then
14         $found := false$ 
15         $k := k + 1$ 
16      $i := i + 1$ 
17   return  $found$ 
```

- 4c) Vysvětlete hlavní princip *amortizované časové složitosti* a její analýzy (v rozsahu 3–4 vět). Dále na vhodném příkladu demonstруйте výhody použití amortizované složitosti v porovnání s analýzou v nejhorším případě. (50 bodů)

Prostor pro řešení Příkladu 4.