Teoretická informatika TIN - 2023/2024

Závěrečná zkouška (1. opravný termín) 17. 1. 2024

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

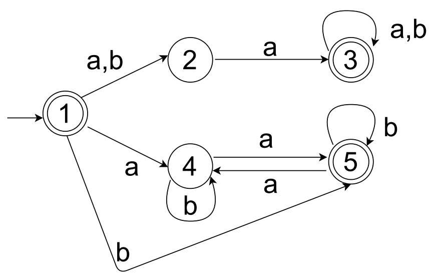
120 bodů

min. 30 bodů

1a) Nechť $A = (Q, \Sigma, \delta, q_0, F)$ je *nedeterministický konečný automat* (NKA). Formálně definujte i) *přechodovou funkci* δ, ii) *konfiguraci* NKA (popište význam jednotlivých položek konfigurace) a iii) *relaci přechodu* mezi konfiguracemi. Dále rozhodněte a dokažte (stačí hlavní myšlenka důkazu), zda je *problém univerzality* daného NKA A (tj. zda $L(A) = \Sigma^*$) *rozhodnutelný*.

(50 bodů)

1b) Uvažme následující nedeterministický konečný automat A.



(a) Převeďte algoritmicky NKA A na ekvivalentní DKA (můžete použít algoritmus, který uvažuje pouze dosažitelné stavy výsledného DKA).

(b) Rozhodněte a dokažte, zda existuje úplný redukovaný deterministický automat A' se 3 stavy takový, že $L(A') = L(A)$.

(c) Formálně zapište jazyk $L(A)$.

(70 bodů)

Prostor pro řešení Příkladu 1.

Prostor pro řešení Příkladu 1.

Příklad 2
120 bodů
min. 30 bodů

2a) Nechť $G = (N, \Sigma, P, S)$ je *bezkontextová gramatika*. Formálně definujte tvar přechodových pravidel P v G. Dále definujte, kdy je G *víceznačná*, a uveďte příklad víceznačné gramatiky, která má 2 neterminály. (30 bodů)

2b) Uvažme jazyk

$$L_1 = \{w \in \{a, b\}^* \mid \#_a(w) < 2 \cdot \#_b(w)\},$$

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

(a) Zkonstruujte bezkontextovou gramatiku G t.ž. $L(G) = L_1$.

(b) Dále pro gramatiku G sestrojte rozšířený zásobníkový automat A, který modeluje *syntaktickou analýzu zdola nahoru.* Automat A popište ve shodě s definicí rozšířeného zásobníkového automatu (tj. nikoliv diagramem).

(60 bodů)

2c) Uvažme abecedu $\Sigma = \{a, b, c\}$. Rozhodněte a dokažte, zda je třída bezkontextových jazyků uzavřena vůči *nekonečnému sjednocení*, tj. pro každou nekonečnou množinu $\{L_0, L_1, L_2 \ldots\}$ bezkontextových jazyků (kde $L_i \subseteq \Sigma^*$) platí, že i jazyk $L = \bigcup_{i=0}^{\infty} L_i$ je bezkontextový. (30 bodů)

Prostor pro řešení Příkladu 2.

Prostor pro řešení Příkladu 2.

Příklad 3
190 bodů
min. 50 bodů

3a) Nechť $M = (Q, \Sigma, \Gamma, \delta, q_0, q_f)$ je *nedeterministický Turingův stroj*. Formálně definujte, kdy je jazyk L i) *rekurzivní* a ii) *rekurzivně vyčíslitelný*. Dále rozhodněte a dokažte, zda existuje Turingův stroj M takový, že $L(M)$ je nekonečný regulární jazyk a množina $\{w \in \Sigma^* \mid M \text{ na } w \text{ cyklí } \}$ je nekonečná.

(40 bodů)

3b) Uvažme $\Sigma = \{a, b\}$. O každém z následujících jazyků rozhodněte a dokažte, zda je či není rekurzivně vyčíslitelný.

i) $L_1 = \{\langle M_1\rangle\#\langle M_2\rangle \mid 0 < |L(M_1)| < |L(M_2)|\}$,

ii) $L_2 = \{\langle M_1\rangle\#\langle M_2\rangle\#\langle n\rangle \mid \ |L(M_1) \cdot L(M_2)| \geq n\}$,

kde $\langle M_1\rangle$ a $\langle M_2\rangle$ označují řetězce, které kódují Turingovy stroje M_1 a M_2, a $\langle n\rangle$ označuje řetězec, který kóduje číslo $n \in \mathbb{N}$.

(70 bodů)

Poznámka: V důkazech použijte redukci či popište fungovaní požadovaného Turingova stroje.

3c) Mějme teorii T se signaturou $\langle\{Cat_{/0}, Fish_{/0}, Trex_{/0}\}, \{eats_{/2}, =_{/2}\}\rangle$ (= je standardní rovnost) se speciálními axiomy

$$\forall x(x = Fish \lor x = Cat \lor x = Trex)$$
$$\forall x\forall y(eats(x, y) \leftrightarrow \neg eats(y, x))$$
$$\forall x(x = Trex \lor eats(Trex, x))$$

i) Rozhodněte a stručně zdůvodněte, zda T je (a) bezesporná, (b) úplná a (c) rozhodnutelná (tj. množina důsledků T je rozhodnutelná).

ii) Popište hlavní myšlenku, jak lze ve výše zmíněné teorii T eliminovat kvantifikátory (tj., jak převést formuli $\exists x\varphi$, kde φ je bez kvantifikátorů, na ekvivalentní formuli bez kvantifikátorů).

iii) Rozhodněte a stručně zdůvodněte, zda může být Peanova aritmetika T_{PA} redukovatelná na T (tj. zda může existovat vyčíslitelná redukce f, která každou větu φ Peanovy aritmetiky převede na větu $f(\varphi)$ jazyka teorie T takovou, že $T_{PA} \models \varphi \iff T \models f(\varphi)$).

(80 bodů)

Prostor pro řešení Příkladu 3.

Prostor pro řešení Příkladu 3.

Příklad 4
170 bodů
min. 50 bodů

4a) Formálně definujte *asymptotické horní omezení* funkce $f : \mathbb{N} \to \mathbb{N}$ (to jest $\mathcal{O}(f(n))$). Dále rozhodněte a stručně dokažte, zda platí následující tvrzení (předpokládejme, že $\mathbf{P} \neq \mathbf{NP}$).

i) Existuje jazyk L_1, který patří do třídy **NP**, ale není **NP**-úplný.

ii) Existuje jazyk $L_2 \in \mathit{DTIME}[n] \setminus \mathit{DSPACE}[n]$.

iii) Existuje regulární jazyk L_3, který nepatří do třídy **P**.

(60 bodů)

4b) Uvažme orientovaný graf $G = (V, E)$, kde V je množina vrcholů (číslovaných přirozenými čísly od 0) a E je množina hran. Dále uvažme konečnou množinu barev K a zobrazení $c : E \to K$.

Pro libovolné $v_s, v_e \in V$ definujme predikát P: $P(v_s, v_e) = \texttt{true} \iff \exists k \in K :$ existuje cesta

$$v_0, e_0, v_1, \ldots, v_{n-1}, e_{n-1}, v_n : v_s = v_0 \wedge v_e = v_n \wedge \forall\ 0 \leq i < n : e_i = (v_i, v_{i+1}) \wedge e_i \in E \wedge c(e_i) = k.$$

Uvažme algoritmus $cPath(G, v_s, v_e)$, který má na vstupu graf G a jeho dva libovolné vrcholy v_s a v_e. $cPath(G, v_s, v_e)$ vrací $\texttt{true} \iff P(v_s, v_n) = \texttt{true}$

i) Analyzujte asymptotickou časovou složitost algoritmu $cPath(G, v_s, v_e)$ v nejhorším případě.

ii) Analyzujte asymptotickou časovou složitost algoritmu $cPath(G, v_s, v_e)$ v nejlepším případě.

iii) Navrhněte algoritmus $cPath^{+}(G, v_s, v_e)$, který řeší stejný problém a má lepší asymptotickou časovou složitost v nejhorším případě. Analyzujte tuto složitost.

```
Function cPath(G, v_s, v_e)
    for int n := 1; n < |V|; n := n + 1 do
        foreach cestu v_0, e_0, v_1, ..., v_{n-1}, e_{n-1}, v_n v G do
            if v_0 ≠ v_s ∨ v_n ≠ v_e then break
            color := c(e_0)
            res := 1
            for int i := 1; i < n; i := i + 1 do
                if c(e_i) ≠ color then res := 0; break
            if res = 1 then return true
    return false
```

(60 bodů)

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

Poznámka 2: Časovou složitost analyzujte vzhledem k velikosti množiny V a E.

Poznámka 3: V bodě iii) můžete předpokládat, že G je efektivně representován pomocí seznamu následníku Adj, kde $Adj(v, c)$ vrací v konstantním čase množinu hran s barvou c, které vycházejí z vrcholu v.

4c) Uvažme dynamicky alokovanou datovou strukturu *mnozina* implementující konečnou množinu přirozených čísel pomocí uspořádaného jednosměrně vázaného lineárního seznamu (předpokládejte, že uspořádání je rostoucí posloupnost) s ukazatelem *head* na začátek.

- Operace $mnozina.insert(x)$ vloží na správnou pozici do seznamu prvek x, pokud v seznamu není (jinak jde o prázdnou operaci). Tato operace má lineární časovou složitost k velikosti seznamu.
- Operace $mnozina.prune(y)$ je implementována následujícím způsobem:

```
Function prune(y)
    while head ≠ null ∧ head → value < y do
        tmp := head
        head := head → next
        print(tmp → value)
        free(tmp)
```

Uvažme posloupnost n operací $mnozina.insert(x)$ a $mnozina.prune(y)$. Analyzujte a zdůvodněte amortizovanou časovou složitost jedné operace $mnozina.prune(x)$.

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

(50 bodů)

Prostor pro řešení Příkladu 4.