PŘÍJMENÍ a JMÉNO:
Login studenta:
DATUM:

Písemná zkouška z předmětu "Teoretická informatika" (UKÁZKA struktury)

Doba trvání: **90 minut** Max. zisk: **62 bodů**

Minimální bodový zisk nutný k uznání: **25 bodů**
je ale nutné také docílit **alespoň minima 11 bodů separátně u každé ze dvou částí písemky**

Obecné pokyny:
Po obdržení zadání ihned do pravého horního rohu napište čitelně své příjmení a jméno, svůj login a datum.
K řešení úloh používejte pomocné papíry. Požadovaný *výsledek* příkladu zapište rozumným způsobem do odpovídajícího místa přímo *u zadání příkladu.* V případě, že se vám stane, že se řešení nevejde do vyhrazeného místa, či jej chcete přepsat apod., pište jej na *zvláštní list* – u příslušného příkladu (např. s číslem [2.3.]) pak napište výrazné ZP (zvláštní papír) a výsledek na zvláštním papíru označte zřetelně číslem příkladu (tedy např [2.3.]). Všechny odevzdávané papíry řádně podepište!
Úlohy řešte v libovolném pořadí, rozvrhněte si dobře čas.
Poznámka. Snažte se vždy o kompletní řešení, za částečné řešení nemusíte dostat žádné body (podle charakteru příkladu).

Poznámka.
Ke zkoušce je možno jít jen po splnění požadavků k zápočtu.
Kromě čistého papíru a psacích potřeb není povoleno používat žádné další pomůcky.
Doneste si s sebou nějaký identifikační doklad (nejlépe občanský průkaz).
==

Písemka je rozdělena do 2 částí.

1. část (0 - 31 bodů; minimum k celkovému uznání písemky 11 bodů)
prověřuje hlavně oblast regulárních a bezkontextových jazyků a jim odpovídajících automatů a gramatik,

2. část (0 - 31 bodů; minimum k celkovému uznání písemky 11 bodů)
prověřuje hlavně oblast teorie algoritmů (vyčíslitelnost a složitost).

V dalším textu je přiblížena struktura písemky, včetně přibližných bodů.

Uvedené **konkrétní příklady reprezentují vždy jen část příslušné oblasti**, skutečné příklady budou samozřejmě vybírány z celé škály odpovídající oblasti.

======================================

1. část

(celkově 0 - 31 bodů, nutné minimum 11 bodů)

Příklad [1.1] (9 bodů)

(konstrukce z oblasti KA, RV, BG, ZA, …)

Definujme regulární gramatiku jako bezkontextovou gramatiku, v níž jsou jen pravidla typu $X \to aY$, $X \to Y$, $X \to \varepsilon$. Navrhněte regulární gramatiku, která generuje níže specifikovaný jazyk L. (Nápověda. Konečný (i nedeterministický) automat je možné snadno převést na regulární gramatiku: stavy automatu lze přirozeně chápat jako neterminály, neterminály odpovídající přijímajícím stavům je možné přepsat na ε.)

$$L = \{w \in \{a,b\}^* \mid w \text{ obsahuje podřetězec } abba \text{ a počet symbolů } a \text{ ve } w \text{ je sudý } \}$$

Váš postup musí být přehledný (okomentovaný), aby bylo snadné ověřit, že vámi uvedená gramatika opravdu generuje jazyk L.

Příklad [1.2] (6 bodů)

(aplikace algoritmického postupu u KA, RV, BG, ZA, …)

Stručně, ale přesně, popište postup redukce gramatiky a přehledně jej aplikujte na zde uvedenou bezkontextovou gramatiku (kde A_1 je počáteční neterminál).

$$
\begin{array}{ll}
A_1 \to aA_1bA_5 \mid A_8A_3 & A_7 \to bbA_7 \mid A_9bA_{12} \\
A_2 \to A_4A_5 \mid A_5bA_2 & A_8 \to A_2bA_{11} \mid ab \\
A_3 \to A_3aA_8b \mid A_8bA_6aA_6 & A_9 \to bb \mid aA_{12}b \\
A_4 \to aA_2A_5 \mid A_5aaA_4 & A_{10} \to A_1abA_4 \mid bA_{10}b \\
A_5 \to A_1aA_4 \mid bA_7A_2 & A_{11} \to A_1bbA_6 \mid A_3bA_8 \\
A_6 \to baA_8a \mid A_{11}abaA_1 & A_{12} \to aA_1bA_{12}a \mid aA_9
\end{array}
$$

Příklad [1.3] (6 bodů)

(porozumění jazykovým operacím a pojmům z oblasti KA, RV, BG, ZA, uzávěrové vlastnosti třídy REG, CFL, ...)

Vysvětlete na rozumném malém příkladu, jak lze k bezkontextové gramatice G sestrojit G' tak, že $L(G') = (L(G))^*$.

Příklad [1.4] (4 body)

(zařazení jazyků do hierarchie)

Vyjmenujte, které z daných jazyků ($|w|$ označuje délku slova w, $|w|_a$ označuje počet výskytů symbolu a ve w, w^R označuje zrcadlový obraz slova w)

$$
\begin{aligned}
L_1 &= \{w \in \{a,b\}^* \mid w = vvv \text{ pro nějaké } v\}, \\
L_2 &= \{0^k 1^\ell 0^m \mid k = \ell \text{ nebo } \ell = m\}, \\
L_3 &= \{0^i 1^j \mid (i \; mod \; 3) = (j \; mod \; 3)\}, \\
L_4 &= \{w \in \{a, +, (,)\}^* \mid w \text{ je generováno gramatikou } S \to S + S \,|\, (S) \,|\, a \;\} \\
L_5 &= \{ww^R \mid w \in \{a,b\}^*\}, \\
L_6 &= \{w \in \{a,b\}^* \mid \text{jestliže } w \text{ neobsahuje podslovo } aaba, \text{ pak má sufix } bba\}
\end{aligned}
$$

jsou regulární:

jsou bezkontextové, ale ne regulární:

nejsou bezkontextové:

Příklad [1.5] (6 bodů)

(důkaz tvrzení, správnosti konstrukce, ...)

Podejte důkaz tvrzení, že každý jazyk generovaný bezkontextovou gramatikou je rozpoznáván zásobníkovým automatem.

2. část

(celkově 0 - 31 bodů, nutné minimum 11 bodů)

Příklad [2.1] (8 bodů)

(konstrukce prokazující pochopení modelů TS a RAM)

Sestrojte přehledně Turingův stroj přijímající (rozhodující) jazyk $\{a^n b^n \mid n \geq 0\}$. Pak stručně vysvětlete, jak jej lze simulovat strojem RAM.

Jiný příklad: může být uveden např. kód RAMu, o němž máte zjistit, co dělá pro konkrétní vstup, případně obecně ....

Příklad [2.2] (8 bodů)

(pochopení pojmu převeditelnosti mezi konkrétními problémy, návrh nedeterministického algoritmu, ...)

Řekněme, že bychom měli ukázat, že problém ekvivalence nedeterministických konečných automatů (Eq-NFA) je polynomiálně převeditelný na problém kvantifikovaných booleovských formulí (QBF).

Měli bychom tedy ukázat, že existuje algoritmus *Alg*, který

- očekává na vstupu (doplňte, co *konkrétně* je vstupem):

- a pro takový vstup vydá výstup (doplňte, co *konkrétně* je výstupem):

- přičemž výstup musí splňovat následující podmínku (ve vztahu ke vstupu):

- Musí *Alg* splňovat jestě nějakou podmínku? (Pokud ano, jakou?):

- Dále uveďte příklady konkrétních instancí uvedených problémů (QBF a Eq-NFA), pro které je odpověď Ano, a příklady konkrétních instancí, pro které je odpověď Ne:

V jiné souvislosti se může objevit např.:

Vysvětlete, co to znamená, když se řekne, že problém hamiltonovské kružnice (HK) patří do NPTIME. Pak stručně, ale jasně a srozumitelně prokažte, že HK patří do NPTIME.

Příklad [2.3.] (5 bodů)

(rozhodnutelné a nerozhodnutelné problémy, částečně rozhodnutelné problémy, aplikace Riceovy věty, ...)

Zjistěte (a zdůvodněte), pro které z následujících vlastností Turingových strojů plyne nerozhodnutelnost z Riceovy věty. Vlastnost je dána otázkou, na niž je odpověď buď ANO (zadaný stroj M vlastnost má) nebo NE (M vlastnost nemá).

a/ Zastaví se M na řetězec 001 ?

b/ Má M více než sto stavů ?

c/ Má v nějakém případě výpočet stroje M více kroků než tisícinásobek délky vstupu ?

d/ Platí, že pro libovolné n se M na vstupech délky nejvýše n vícekrát zastaví než nezastaví ?

e/ Zastaví se M na každém vstupu w za méně než $|w|^2$ kroků?

f/ Je pravda, že pro lib. vstupní slovo M realizuje jeho zdvojení ?

g/ Je pravda, že M má nejvýše sto stavů nebo více než sto stavů ?

Příklad [2.4.] (4 body)

(složitost algoritmů, příklady C-úplných problémů, zařazení problémů do tříd složitosti, ...)

Vypište písmena těch následujících vztahů (s velkým O a malým o), které

- platí:
- neplatí:

a) $100n^4 \in O(0.1n^4 - 1000n^2)$

b) $\sqrt{n} \in o(n)$

c) $\log n \in O(\sqrt{n})$

d) $2^n \in O(n^{1024})$

e) $n^{1000} \in o(2^n)$

f) $n! \in O(2^n)$

JINÝ PŘÍKLAD:

Vysvětlete, co je to PSPACE-úplný problém, a uveďte příklad takového problému.

Instance:

Otázka:

JINÝ PŘÍKLAD:

Ke každému z následujících problémů uveďte vždy nejmenší třídu složitosti, o které víte, že daný problém obsahuje (třídy jsou brány vzhledem k RAMu s logaritmickou mírou); zmiňte přitom, co považujete za míru velikosti vstupu, pokud se nejedná o obvyklou velikost přirozeného zápisu instance problému.

- jazyková ekvivalence dvou determ. konečných automatů
- problém pravdivosti kvantifikovaných booleovských formulí
- problém jazykové ekvivalence dvou bezkontextových gramatik
- problém jazykové ekvivalence dvou deterministických zásobníkových automatů
- třídění řádků textového souboru podle prvních slov v řádcích
- problém zjištění, zda zadané dekadické číslo je prvočíslem

Příklad [2.5.] (6 bodů)

(aproximační a/nebo pravděpodobnostní algoritmy ....)

Popište přesně, co to je pravděpodobnostní algoritmus, vysvětlete, kdy má jeho použití význam a popište co možná nejprecizněji nějaký konkrétní příklad takového algoritmu.

Jiný příklad: Vysvětlete, jak lze využitím algoritmu nalézajícího minimální kostru v grafu aproximovat problém Δ-TSP (problém obchodního cestujícího, v jehož instancích je splněna trojúhelníková nerovnost) s aproximačním poměrem 2.