Prova de SAC de Teoria da Computação — Prof. José de Oliveira Guimarães
DComp — Campus de Sorocaba da UFSCar, 2015

Ao final da prova, entregue APENAS a folha de respostas. A folha de questões não será considerada. Utilize quantas folhas de respostas forem necessárias. Não é necessário entregar as questões em ordem. Utilize lápis ou caneta, mas faça letra legível. Na correção, símbolos ou palavras ilegíveis não serão considerados. **Justifique** todas as respostas.

Se você estudou com algum colega para esta prova, não se sente ao lado dele pois é possível que acidentalmente vocês produzam respostas semelhantes para alguma questão. Provas de alunos que fizeram a prova próximos uns dos outros com respostas semelhantes caracterizam cópia de questões. Lembro-os de que todos os envolvidos na utilização de métodos ilegais na realização desta prova receberão zero de nota final da disciplina (e não apenas nesta prova).

Coloque o seu nome na folha de resposta, o mais acima possível na folha, seguido do número da sua coluna de carteiras. A primeira carteira é a mais perto da porta e a última a mais perto das janelas. Não precisa colocar o RA.

1. (3,0) Esta questão se baseia na seguinte gramática na qual as letras em minúscula e dígitos são terminais.

```
S ::= A          S ::= B
A ::= a A b      A ::= 0
B ::= 1 C
C ::= C a        C ::= c
```

Chamaremos esta gramática de G e a linguagem gerada por ela de L(G). Baseado em G, faça:

(a) a descrição de L(G) em termos de conjuntos, algo do tipo $\{0^n 1 a b^k : n > 0 \text{ e } k \geqslant 0\}$;

(b) é possível fazer uma expressão regular que gera a mesma linguagem que esta gramática ? Justifique mesmo que intuitivamente;

(c) o autômato com pilha que decide a linguagem L(G).

2. (1,0) Prove que o conjunto R que contém todas as linguagens regulares é fechado sobre a operação de concatenação. Isto é, dadas L e K linguagens regulares, $L \circ K = \{wt : w \in L \text{ e } t \in K\}$ pertence a R. Faça apenas o desenho como está no Sipser, com retângulos com três círculos representando autômatos.

3. (1,0) Faça um autômato não determinístico que não seja determinístico. Faça a árvore de derivação de uma entrada para este autômato. Esta entrada deve ter pelo menos três elementos.

4. (2,0) Explique como funciona uma MTND. Faça um pequeno exemplo para auxiliar o seu raciocínio.

Faça duas e APENAS duas dentre as questões seguintes. Cada uma delas vale 1,5 pontos.

5. Prove que, se L é uma linguagem decidível, $G = L - \{0\}$, G^c é decidível.

6. Uma MT executa por t passos. Então no máximo quantas células diferentes da fita ela utilizou? Isto é, qual, no máximo, o espaço utilizado pela MT? Diga qual o relacionamento, usando $\subset$, $\in$ e $\notin$, entre as classes TIME(n) e SPACE(n). Mais geralmente, qual o relacionamento entre TIME(f) e SPACE(f)? Justifique.

7. Mostre como simular uma MT M com uma fita infinita nas duas direções por uma MT M' com uma fita infinita em apenas uma direção. Não apenas mostre a correspondência entre as fitas. Mostre como a simulação é feita (como é o algoritmo de M'). No início da execução, assuma que a cabeça de leitura e gravação de

(a) M está sobre o primeiro símbolo da entrada x (que está em binário);

(b) M' está sobre a segunda célula da fita (isto facilitará a simulação).

8. Uma MT M_3 de três fitas pode ser simulada por uma MT M_1 de uma fita. Mostre como é feito o mapeamento entre as três fitas de M_3 para a fita de M_1. Não é necessário explicar a simulação em si.

9. Prove que a linguagem $K = \{<M> \sqcup x : M(x) \downarrow\}$ é recursivamente enumerável.

10. Prove que a linguagem $H = \{<M> \sqcup x : M(x) \downarrow\}$ não é recursiva. Dica: assuma que H é decidida por um programa em C "`int para(CM, x)`" no qual `CM` é o código correspondente a M e `x` é a entrada.

11. Prove que, se L e L^c são recursivamente enumeráveis, L é recursiva.

12. Explique o que é uma MT Universal U. Qual o correspondente a uma MT Universal na computação "real" ? É possível existir duas MT universais diferentes?

Resumo:

Uma expressão regular sobre um alfabeto Σ é descrita como: a) x é uma e.r. (expressão regular) se $x \in \Sigma$; b) ϵ é uma e.r. c) $\emptyset$ é uma e.r. d) $x?$, $(x \cup y)$, (xy) (concatenação de x e y) e $(x^\star)$ são e.r. se x, y são e.r. Assume-se que a ordem de precedência dos operadores seja, do maior para o menor: $\star$, concatenação e união ($\cup$). Parenteses podem ser removidos se forem redundantes. A linguagem associada a uma expressão regular é definida indutivamente como $L(x) = \{x\}$ se $x \in \Sigma$, $L(\epsilon) = \{\epsilon\}$, $L(\emptyset) = \emptyset$, $L(R_1 \cup R_2) = L(R_1) \cup L(R_2)$, $L(R_1 \circ R_2) = L(R_1) \circ L(R_2)$, $L(R_1^\star) = (L(R_1))^\star$, $L(R?) = L(R) \cup \{\epsilon\}$.

A concatenação de duas linguagens L e K é $L \circ K = \{vw : v \in L \text{ e } w \in K\}$.

Lema do bombeamento para linguagens livres de contexto: Se A é uma linguagem livre de contexto, então existe um número p onde, se s é uma cadeia qualquer em A de comprimento pelo menos p, então s pode ser dividida em cinco partes, $s = uvxyz$ satisfazendo as condições: para cada $i \geq 0$, $uv^ixy^iz \in A$; $\left|vy\right| > 0$; $\left|vxy\right| \leqslant p$.

Uma máquina de Turing determinística (MT ou MTD) é uma quadrupla (Q, Σ, I, q) na qual Q e Σ são conjuntos chamados de conjuntos de estados e de símbolos, I é um conjunto de instruções, $I \subset Q \times \Sigma \times Q \times \Sigma \times D$, $D = \{-1, 0, 1\}$ e $q \in Q$ é chamado de estado inicial. Há dois estados especiais: q_s e q_n, todos elementos de Q e todos diferentes entre si. Neste texto convenciona-se que o estado inicial será q_0 a menos de menção em contrário. Exige-se que $\{0, 1, \triangleright, \sqcup, \square\} \subset \Sigma$. Uma instrução é da forma (q_i, s_j, q_l, s_k, d) na qual $s_k \neq \square$ e $q_i \notin \{q_s, q_n\}$. Se $(q, s, q'_0, s'_0, d_0), (q, s, q'_1, s'_1, d_1) \in I$, então $q'_0 = q'_1$, $s'_0 = s'_1$ e $d_0 = d_1$. Q, Σ e I são conjuntos finitos.

O símbolo $\square$ é o branco utilizado para as células ainda não utilizadas durante a execução e $\sqcup$ é utilizado para separar dados de entrada e saída.

Símbolos: Máquina de Turing Não Determinística (MTND), Máquina de Turing (MT), Máquina de Turing Determinística (MTD). A menos de menção em contrário, uma MT é uma MTD. Todas as linguagens utilizadas são subconjuntos de $\{0, 1\}^\star$.

Uma linguagem L é computável (recursiva) se existe uma MT M de decisão tal que

$$x \in L \text{ sse } M(x) = 1$$

Isto é, M decide L.

Uma linguagem L é ou computacionalmente enumerável, c.e. (recursivamente enumerável, r.e.) se existe uma MT M tal que se $x \in L$ então $M(x) = 1$. Se $x \notin L$, $M(x) \uparrow$. Dizemos que M *aceita* L.