Prova Substitutiva de Teoria da Computação — Prof. José de Oliveira Guimarães
DComp — Campus de Sorocaba da UFSCar, 2014

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. (4,0) Sobre gramáticas e autômatos, responda:

   1. (1,5) faça uma gramática G tal que $L(G) = \{a^n b c^n : n \in \mathbb{N}, n \geqslant 1\}$;

   2. (1,0) explique por que um autômato finito não pode reconhecer $L(G)$. A explicação pode ser informal. $L(G) = \{w : S \Longrightarrow^\star w\}$, sendo S o símbolo inicial da gramática;

   3. (1,5) explique como um autômato com pilha que reconhece $L(G)$ confere se o número de c´s é igual ao número de a´s. Não é necessário fazer o autômato, apenas explique este ponto. E apenas este ponto.

2. (2,0) Sejam M e M' dois autômatos finitos. Mostre como obter um autômato que reconheça $L(M) \cdot \{0\} \cdot L(M')$. Faça apenas o desenho como está no Sipser, com retângulos com três círculos representando autômatos. Lembre-se de que, se A e B são linguagens, $A \cdot B = \{ab : a \in A \text{ e } b \in B\}$. E $\{0\}$ é uma linguagem com um único elemento.

3. (2,0) Uma MT M ignora a sua entrada e imprime um elemento $x_0 \in A$ na fita. Após isto, imprime espaço e um elemento x_1, seguido novamente de espaço, x_2 e assim por diante. Sendo A uma linguagem, qualquer elemento de A irá ser impresso na fita após um tempo indeterminado. E apenas elementos de A serão impressos. Ou seja, M enumera A. Prove que A é computacionalmente enumerável. Isto é, existe uma outra MT P tal que $P(x) = 1$ se $x \in A$ e $P(x) \uparrow$ se $x \notin A$.

4. (4,0) Sobre máquinas de Turing, responda:

   1. (1,5) faça o diagrama de uma MTND que não seja determinística. A MTND pode ter um único estado;

   2. (1,5) faça a codificação da MTND feita no item anterior;

3. (1,0) sejam U_1 e U_2 MT universais diferentes. Qual o resultado da chamada de U_1 passando como parâmetro $\langle U_2 \rangle \sqcup x$, sendo x igual a $\langle M \rangle \sqcup 0$?

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 $\{0, 1, \rhd, \sqcup, \square\} \subset \Sigma$. 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.