Prova de PAC de Teoria da Computação — 2018/2
Prof. José de Oliveira Guimarães — DComp — Campus de Sorocaba da UFSCar

Ao final da prova, entregue APENAS a folha de respostas. A folha de questões não será considerada. Utilize quantas folhas de reposta 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.

A menos de menção em contrário, usaremos o alfabeto $\Sigma = \{0, 1\}$ nesta prova.

Primeira Parte da Matéria

1. (2,0) Dados dois autômatos finitos $M = (Q, \Sigma, \delta, q_0, F)$ e $M' = (Q', \Sigma', \delta', q_1, F')$, mostre como obter o autômato que reconhece as cadeias $L(M)^{\star}L(M')$. Lembre-se de que, se A e B são linguagens, $AB = \{ab : a \in A \text{ e } b \in B\}$. E $A^{\star} = \{x_1x_2 \ldots x_k : k \in \mathbb{N} \text{ e } x_i \in A\}$. Você pode fazer o desenho do diagrama como está no livro do Sipser: um quadrado com três círculos, sendo um deles o estado inicial e os outros dois estados finais.

2. (1,0) Prove que a gramática

$$S \longrightarrow S + S \mid S * S \mid N$$
$$N \longrightarrow 0 \mid 1 \mid \ldots \mid 9$$

é ambígua.

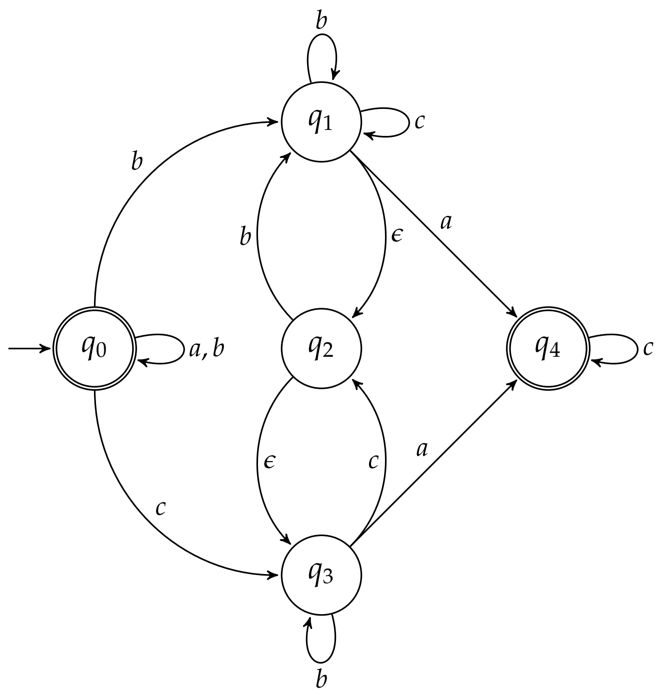
3. (1,0) Faça uma expressão regular E tal que $L(E) = L(N)$, sendo N o AFND com estado inicial q_0, estado final q_2 e cuja tabela de transição é :

δ	a	b	c
q_0	$\{q_0, q_1\}$		$\{q_2\}$
q_1	$\{q_1\}$	$\{q_2\}$	$\{q_2\}$
q_2	$\{q_2\}$		

4. (1,0) O diagrama do autômato finito $M = (Q, \Sigma, \delta, q_0, F)$ está no final desta questão. Utizando-o, responda:

(a) qual o domínio e contra-domínio de δ?

(b) qual o valor de $\delta(q_0, b)$?



Segunda Parte da Matéria

5. (1,5) Escolha e faça UM e APENAS UM dos itens abaixo.

(a) Prove que, se L e L^c são linguagens enumeráveis, L é decidível.

(b) Prove que, se L e K são linguagens decidíveis, $L - K^c$ é enumerável.

(c) Prove que, se L e K são linguagens decidíveis, $L^c \cap K^c$ é decidível.

6. (1,5) Faça uma MTND N com pelo menos dois estados e pelo menos três instruções. N não pode ser uma MTD. Faça uma árvore de computação para uma certa entrada para N (à sua escolha) de tal forma que a árvore tenha pelo menos altura 2 (duas transições).

7. (2,0) Uma instrução de uma MT $M = (Q, \Sigma, I, q)$ é um elemento do conjunto $Q \times \Sigma \times Q \times \Sigma \times D$. Baseado nisto,

(a) (1,0) faça a codificação de uma instrução $(q_0, 0, q_1, 1, E)$ assumindo que $Q = \{q_0, q_1, ..., q_9\}$, $\{0, 1\} \subset \Sigma$ e $|\Sigma| = 5$;

(b) (1,0) explique como funciona uma MT Universal (MTU). Explique que parâmetros ela toma e o que faz.

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 concatenação de duas linguagens L e K é $L \circ K = \{vw : v \in L \text{ e } w \in K\}$. E $L^\star = \{w_1 w_2 \ldots w_n | n \geqslant 0\}$.

O lema do bombeamento para linguagens regulares: se A é uma linguagem regular, então existe um inteiro p dependente de A tal que, se $s \in A$ e $|s| \geq p$, então s pode ser dividida em três partes, $s = xyz$, satisfazendo (a) $xy^iz \in A$ para todo $i \geq 0$; (b) $\left|y\right| > 0$ e (c) $\left|xy\right| \leq p$.

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.

Dizemos que uma MT M enumera os elementos de uma linguagem $L \subset \Sigma^\star$ (ou $A \subset \mathbb{N}$) não vazia se:

(a) M despresa a sua entrada e imprime na fita um elemento de L, seguido de $\sqcup$, seguido de outro elemento de L, seguido de $\sqcup$ e assim por diante;

(b) dado $x \in L$, em algum momento da execução de M x será impresso na fita;

(c) M nunca pára a sua execução.

Note que os elementos impressos na fita por M podem ser repetidos e estar fora de ordem. Em particular, se L (ou A) for finito, haverá repetições.

Uma linguagem L pertence a TIME(f) se existe uma MT M de decisão que toma uma entrada x e que termina a sua execução depois de um número de passos menor ou igual a $cf(n)$ tal que $x \in L$ sse $M(x) = 1$. Ou seja, M executa em tempo $cf(n)$. Uma linguagem L pertence a SPACE(f) se existe uma MT M que toma uma entrada x e que termina a sua execução depois de utilizar um número de células menor ou igual a $cf(n)$ nas fitas de trabalho (todas exceto a de entrada e a de saída). Ou seja, M executa em espaço $cf(n)$. NTIME(f) e NSPACE(f) têm definições análogas, mas que usam MTND´s.

$$\mathrm{P} = \bigcup_{k \in \mathbb{N}} \mathrm{TIME}(n^k)$$

$$\mathrm{NP} = \bigcup_{k \in \mathbb{N}} \mathrm{NTIME}(n^k)$$