Prova Substitutiva de Teoria da Computação — 2016/2
Prof. José de Oliveira Guimarães — DComp — Campus de Sorocaba da UFSCar

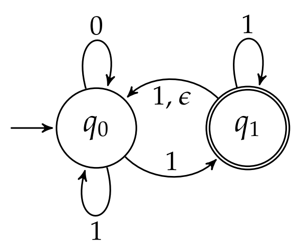
Ao final da prova, entregue APENAS a folha de respostas. A folha de questões não será considerada. Utilize quantas folhas de reposta forem necessárias. Não é necessário entregar as questões em ordem. Utilize lápis ou caneta, mas faça letra legível. Na correção, símbolos ou palavras ilegíveis não serão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

1. (2,0) Esta questão utiliza o seguinte autômato finito:



Baseado nele,

(a) faça a árvore de computação deste autômato com a entrada 110;

(b) defina aceitação de uma cadeia por um AFND.

2. (4,0) Sobre linguagens, gramáticas, expressões regulares e autômatos,

(a) (1,5) faça uma gramática G tal que $L(G) = \{a^n 0^k 1^k b^n \,|\, n \geqslant 0,\ k > 0\}$. Não é necessário justificar;

(b) (1,0) faça uma expressão regular E cuja linguagem seja $\{a^n c | n \geqslant 0\} \cup \{tai^n \,|\, n \geqslant 1\}$. Não é necessário justificar;

(c) (1,5) faça o diagrama de um autômato com pilha que reconheça $L(E)$ ou explique porque isto não é possível. A etiqueta de cada seta do diagrama é do tipo $a, b \longrightarrow c$ sendo a o símbolo corrente, b o topo da pilha, que é desempilhado, c o símbolo da pilha a ser empilhado. Não é necessário justificar.

3. (2,0) Faça os seguintes itens sobre máquinas de Turing (MT).

(a) (1,0) Escreva a tese de Church-Turing.

(b) (1,0) A figura abaixo mostra as duas fitas de uma única MT. Esta MT pode ser simulada por uma outra MT de uma única fita. Para isto, é necessário mapear os símbolos nas duas fitas para uma única fita. Mostre como este mapeamento pode ser feito. Copie o diagrama abaixo na resposta, acrescente a fita da MT de uma única fita e faça a correspondência entre as células. Não é necessário justificar, apenas mostre a correspondência usando setas (devem existir 12 delas na correspondência mais óbvia — há infinitas delas).

$\cdots$		a_2	a_1	a_0	b_0	b_1	b_2		$\cdots$	**fita 1**
$\cdots$		c_2	c_1	c_0	d_0	d_1	d_2		$\cdots$	**fita 2**

4. (4,0) L e K são linguagens computáveis. Existe uma MT M_{L0} que decide L e que executa em tempo polinomial, uma MT M_{L1} que decide L e que executa em tempo exponencial, uma MT M_K que decide K em tempo f, $f(n) = n^2+n-1$. A linguagem I é uma linguagem computacionalmente enumerável e a MT M_I é tal que

$$x \in I \Longrightarrow M_I(x) = 1$$

$$x \notin I \Longrightarrow M_I(x) \uparrow$$

Baseado nestes dados, responda aos itens abaixo.

(a) (1,0) Prove que $L - \{101\}$ é computável. Se precisar descrever uma MT, descreva-a em alto nível, exatamente como descreveria um algoritmo em palavras. Se fosse uma MT que decide o conjunto dos pares, a descrição poderia ser "A MT verifica se o dígito mais à direita da entrada é 0. Se for, produz saída 1. Se este dígito for 1, produz 0 com saída".

(b) (1,5) Se M_K é executada com entrada 10101 é possível que o número de passos que ela toma até parar seja igual a 3? Igual a 25? Igual a 32?

(c) (1,5) Prove que $I \cap L$ é computacionalmente enumerável.

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 $L \circ K = \{vw : v \in L \text{ e } w \in K\}$.

A definição de uma gramática pode ser feita com a seguinte sintaxe:

$A \longrightarrow 0A$
$A \longrightarrow BC$
$B \longrightarrow bB|\epsilon$
$C \longrightarrow c$

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. A saída de uma MT é o conjunto de símbolos da célula corrente até o $\triangleright$ que está à direita (deve existir alguma célula à direita da célula corrente com este símbolo).

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.

Uma linguagem L pertence a $\text{TIME}(f)$ 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 $\text{SPACE}(f)$ 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 $cf(n)$. $\text{NTIME}(f)$ e $\text{NSPACE}(f)$ têm definições análogas, mas que usam MTND´s.

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