Substitutiva da Primeira Prova de Teoria da Computação
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.

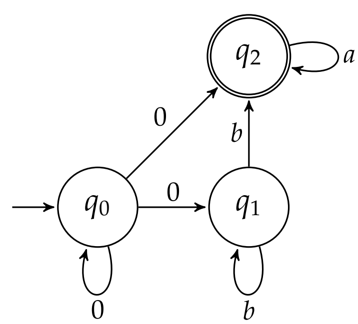
Justifique todas as respostas.

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.

Peça ao professor examplos de árvores de computação, autômatos com pilha e gramáticas.

1. (5,0) Esta questão utiliza o seguinte autômato finito N:



Baseado neste autômato,

(a) (1,0) faça uma expressão regular E tal que $L(E) = L(N)$;

(b) (1,0) faça um autômato finito determinístico M tal que $L(N) = L(M)$. Você deve obrigatoriamente utilizar a técnica de criação de um AFD a partir de um AFND dada no livro do Sipser;

(c) (1,0) defina formalmente este AFND. Isto é, dê os conjuntos e funções de $N = (Q, \Sigma, \delta, q_0, F)$;

(d) (1,0) faça a árvore de computação da cadeia $00ba$. Não é necessário justificar;

(e) (1,0) faça uma gramática G tal que $L(G) = L(N)$.

2. (1,0) Uma MT M com três fitas pode ser simulada por uma MT K com uma fita. Mostre um modo de mapear a fita de M em K. Não é necessário especificar a simulação. Não se esqueça de nenhum detalhe.

3. (2,0) Prove que, se as linguagens L e L^c são computavelmente enumeráveis, então L e L^c são computáveis.

4. (2,0) Em uma linguagem CST (C Sem Tipos), cada função pode ser mapeada em um número inteiro. Suponha que exista, nesta linguagem, uma função `Para(x, y)` que retorna `1` se a função correspondente a `x` pára quando toma a entrada `y`. E `Para(x, y)` retorna `0` se a função correspondente a `x` não pára quando toma a entrada `y`. Usando a função `Q` dada abaixo, encontre uma contradição quando se chama `Q` passando o número de `Q` como parâmetro.

```
void Q( T ) {
   while ( Para(T, T) )
      ;
}
```

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) $(x \cup y)$, (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 definição formal de computação em um AFND é a seguinte: seja $N = (Q, \Sigma, \delta, q_0, F)$ um AFND e w uma cadeia sobre o alfabeto Σ. Então dizemos que N aceita w se podemos escrever w como $y_1 y_2 ... y_m$, onde cada y_i é um membro de $\Sigma \cup \{\epsilon\}$ e existe uma sequência de estados r_0, r_1, ... r_m em Q com três condições: (a) $r_0 = q_0$; (b) $r_{i+1} \in \delta(r_i, y_{i+1})$ para $i = 0, \dots, m-1$ e (c) $r_m \in F$.

Sendo $G = (V, \Sigma, P, S)$ uma gramática, $L(G) = \{w \in \Sigma^\star : S \Longrightarrow^\star w\}$. Sendo E uma expressão regular sobre um alfabeto Σ, $L(E)$ é o conjunto de todas as cadeias reconhecidas por E.

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.