Segunda Prova 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.

1. (3,0) Sobre Máquinas de Turing Universais (MTUs), responda:

(a) (2,0) codifique a instrução $(q_0, 0, q_1, 1, P)$ de uma MT $M = (Q, \Sigma, I, q)$ tal que $|Q| = 5$ e $\Sigma = \{0, 1, \triangleright, \sqcup, \square\}$.

> Formato da Resposta:
> Você pode inventar a sua codificação ou usar a da apostila. Não é necessário justificar se os passos que você seguiu estiverem claros na resposta. Isto é, se pelo raciocínio descrito na resposta for possível descobrir como é a sua codificação.

(b) (1,0) sendo U uma MTU, explique o que calcula $U(m, x)$.

2. (3,0) Utilizando as MTs M_A, M_B e M_C dadas abaixo, responda:

(a) (1,0) qual a função $f : \mathbb{N} \longrightarrow \mathbb{N}$ calculada por M_A?

> Formato da Resposta:
> A resposta deve ser algo como "$f(x) = x * x - x + 1$" ou "$f(x)$ é o valor tal que ...". Não use características específicas de uma linguagem de programação, por exemplo, que 7/2 é igual a 3 em Java;

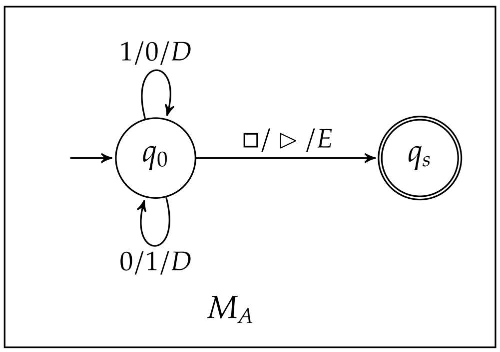
(b) (1,0) descreva a linguagem $L(M_B)$.

Formato da Resposta:
A resposta deve ser algo como

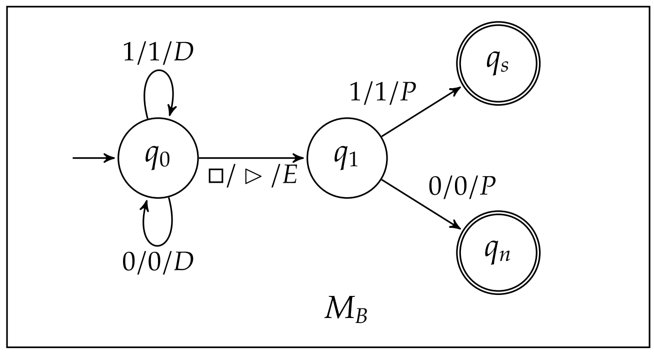
$$L(M_B) = \{\, x \in \mathbb{N} \mid \text{ restrição de } x \,\}$$

Você deve especificar as restrições que x deve obedecer para pertencer à $L(M_B)$. Não é necessário justificar.

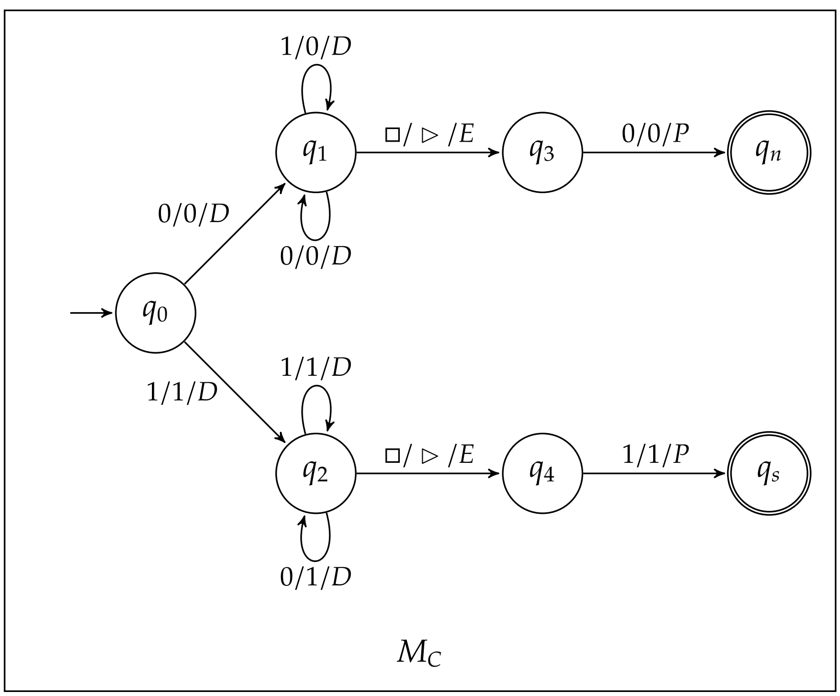
(c) (1,0) descreva a linguagem $L(M_C)$. O formato resposta deve ser como descrita no item anterior.



M_A



M_B



M_C

3. (3,0) Faça UM e APENAS UM dos itens abaixo.

(a) Os conjuntos A, B e C são tais que $A \cup B \cup C = \mathbb{N}$, $A \cap B = \emptyset$, $A \cap C = \emptyset$ e $B \cap C = \emptyset$. Os conjuntos A e B são computáveis e as MTs que os decidem são M_A e M_B, respectivamente. Prove que C é computável.

> Formato da Resposta:
> C é computável se há uma MT que o decide. Então a resposta deve obrigatoriamente descrever M_C por palavras ou usando uma variação da linguagem C, como feito em aula. Neste último caso, assuma que as funções $m_A(x)$ e $m_B(x)$ decidem A e B.

(b) Este item possui duas partes. Primeiro, prove que o conjunto T de todas as codificações de MTs é enumerável. Assuma que fixamos a codificação. Segundo, prove que o conjunto de todas as *linguagens* decidíveis é enumerável.

(c) Prove que, se o conjunto A é enumerável e B é decidível, $C = A - B$ é enumerável.

> Formato da Resposta:
> A prova pode ser de duas maneiras:
>
> (i) mostrando uma MT M_C tal que $M_C(x) = 1$ se $x \in A - B$ e $M_C(x) \uparrow$ se $x \notin A - B$ ou;
>
> (ii) descrevendo uma MT M_C que enumera os elementos de $A - C$.

4. (3,0) Sobre complexidade, responda UM e APENAS UM dos itens abaixo.

(a) Uma MT M toma uma entrada x cujo tamanho é $n = \log_2 x$. M imprime na fita os números entre 0 e x e para (em velhos tempos, *pára*). Dois números são separados por $\sqcup$. Qual a complexidade em

   (i) tempo de M?

   (ii) espaço de M?

   Justifique.

(b) Pode-se afirmar que $TIME(n^2) \subset SPACE(n^3)$? Responda sim ou não e justifique.

Resumo:

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.

Utilizaremos máquinas de Turing para dois propósitos: a) calcular o valor de uma função computável e b) para problemas de decisão. No primeiro caso, o valor da função com entrada x será denotado por $M(x)$ dado que o nome da TM seja M. No segundo caso, exige-se que, quando a máquina pára, o estado corrente seja q_s ou q_n.

Definição 0.0.1. *Em uma computação $M(x)$ (execução da MT M com a entrada x), se o estado final for q_s, consideraremos que M aceita a entrada x e quando o estado final for q_n, que M rejeita x.*

Para tornar os dois tipos de máquinas (retorna o valor e aceita/rejeita) compatíveis, considere que máquinas do tipo b) também colocam um valor 1 ou 0 na fita ao final da computação, conforme o estado final seja q_s ou q_n. Assim pode-se considerar que os dois tipos de máquinas calculam o valor de uma função. E quando uma MT calcula o valor de uma função, o estado final deverá ser q_s.

Em qualquer caso, quando a máquina pára a cabeça de leitura/gravação estará sobre o símbolo mais à esquerda do resultado. O símbolo $\triangleright$ é colocado na primeira célula à direita da saída para indicar o fim desta. Todas as máquinas de Turing utilizadas neste texto terão todas as características dadas no texto acima.

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.

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

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