**Segunda Prova de Teoria da Computação — Prof. José de Oliveira Guimarães**
**DComp — Campus de Sorocaba da UFSCar, 2015**

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 MTs e complexidades de linguagens, responda:

(a) (2,0) seja $K = \{2n : n \in \mathbb{N}\}$. Prove que esta linguagem é decidível fazendo explicitamente a representação gráfica da MT $M_0$ que a decide. Lembre-se de que a entrada é um número em binário;

(b) (1,0) qual a complexidade da MT $M_0$ que você construiu? Justifique suscintamente.

(c) (1,0) $K \in P$ ? Neste item você pode assumir que a MT $M_0$ que você construiu é a mais eficiente possível em relação ao tempo.

2. (2,0) Mostre como as fitas de uma MT $M_2$ de duas fitas podem ser mapeadas na única fita de uma MT $M_1$. Assuma que, em certo momento da computação de $M_2$, a primeira fita de $M_2$ guarda os símbolos $\ldots \square a01b\square \ldots$ e a segunda guarda os símbolos $\ldots \square 10ba\square \ldots$. O símbolo corrente da primeira fita é $a$ e o da segunda fita é 0. Obviamente, antes do primeiro e após o último $\square$ só há símbolos $\square$.

3. (3,0) Seja $H = \{\langle M\rangle : M(\langle M\rangle)\downarrow\}$. Esta linguagem é indecidível. A prova parcial deste fato está a seguir, mas com algumas partes ausentes, sendo substituídas por ?$n$. Na folha de respostas, dê o valor de ?0, ?1, etc. Como exemplo, ?0 = $H$.

*Provaremos por contradição supondo a existência de uma MT P que decide H. Então:*

$$\langle M \rangle \in \ ?0 \ \ sse\ M(\langle M \rangle)\downarrow \ \ sse\ P(\langle M \rangle) = \ ?1$$

*Construiremos uma MT Q que toma uma entrada x e:*

1. *simula a execução de P com x como entrada;*
2. *se a saída de P for 1, Q ?2;*
3. *se a saída de P for 0, Q retorna 0.*

*O que acontece se executarmos Q passando ⟨Q⟩ como entrada? ?3. Como a única premissa é a existência de P, concluimos que P não pode existir.*

Nota: ?2 é uma frase e ?3 são várias frases, toda a conclusão.

4. (2,5) Prove que, se $L$ é computável, $K = \{\,0x1 \,|\, x \in L\}$ é computacionalmente enumerável. Obs: a prova de que $K$ é computável valerá 1,5 pontos.

Resumo:

Uma máquina de Turing determinística (MT ou MTD) é uma quadrupla $(Q,\ \Sigma,\ I,\ q)$ na qual $Q$ e $\Sigma$ 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 $d_0 = d_1$. $Q$, $\Sigma$ 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* $L$.

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