TEORIA DA COMPUTAÇÃO

Prova 2 - Prof. Marcus Ramos - 10/06/2011

1) (1 ponto) Provar que a linguagem $\{< M >| M$ é uma Máquina de Turing e $L(M)$ contém todas as cadeias palíndromas sobre o seu alfabeto de entrada, entre outras$\}$ é indecidível. Uma cadeia w é palíndroma se $w = w^R$. Dica: usar o Teorema de Rice.

Solução disponível em http://infolab.stanford.edu/~ullman/ialcsols/sol9.html.

2) (1.5 ponto) Provar que a linguagem $\{< M >| M$ é uma Máquina de Turing que, quando inicia a sua operação com uma fita totalmente em branco, escreve algum símbolo diferente de branco na mesma em algum momento$\}$ é decidível. Dica: considerar as primeiras $|Q|$ transições realizadas por M, onde Q é o seu conjunto de estados.

Solução disponível em http://infolab.stanford.edu/~ullman/ialcsols/sol9.html.

3) (1 ponto) Determinar se existe solução para a instância PCP abaixo. Provar sua resposta.
A = (01,001, 10)
B = (011, 10, 00)

Solução disponível em http://infolab.stanford.edu/~ullman/ialcsols/sol9.html.

4) (1.5 ponto) Considerar instâncias PCP sobre um alfabeto de um único símbolo. Se todas as cadeias da lista A tiverem comprimento maior que as correspondentes cadeias da lista B, então não haverá solução. Tampouco haverá se as cadeias da lista B tiverem comprimento maior que as correspondentes cadeias da lista A. Provar que em todos os outros haverá solução e que portanto esse problema é decidível. Dica: considerar as cadeias ${a_i}^n {a_j}^m$ e ${b_i}^n {b_j}^m$, onde i refere-se ao índice de um par (a_i, b_i) tal que $|b_i| - |a_i| = m$ e j refere-se ao índice de um par (a_j, b_j) tal que $|a_j| - |b_j| = n$.

Solução disponível em http://infolab.stanford.edu/~ullman/ialcsols/sol9.html.

5) (1 ponto) Provar que se $P_1 \in \mathcal{P}$ e $P_2 \in \mathcal{P}$, então $P_1 \cup P_2 \in \mathcal{P}$.

Solução disponível em http://infolab.stanford.edu/~ullman/ialcsols/sol10.html.

6) (1 ponto) A fórmula $x \wedge (y \vee \sim x) \wedge (z \vee \sim y)$ pertence à SAT ($\sim x$ denota a negação de x)? Justificar sua resposta.

Solução disponível em http://infolab.stanford.edu/~ullman/ialcsols/sol10.html.

7) (2 pontos) Conceituar:
- (0.4 ponto) $f(n)$ é $O(n^2)$;
  Existem constantes n_0 e c tais que para $n \geq n_0$, $f(n) \leq c * n^2$.
- (0.4 ponto) Classe $\mathcal{NP}$;

Conjunto das linguagens que podem ser decididas por uma Máquina de Turing não-determinística de tempo polinomial. Ou, que podem ser verificadas em tempo polinomial.

- (0.4 ponto) Redução de tempo polinomial;
  Função f que mapeia sentenças de uma linguagem L_1 em sentenças de uma linguagem L_2 em tempo polinomial no comprimento da sentença de L_1.

- (0.4 ponto) Problema $\mathcal{NP}$-hard;
  Problema para o qual todos os problemas de $\mathcal{NP}$ podem ser reduzidos em tempo polinomial.

- (0.4 ponto) Problema $\mathcal{NP}$-completo.
  Problema $\mathcal{NP}$-hard que pertence à $\mathcal{NP}$.

8) (1 ponto) Se $P_2 \in \mathcal{P}$ e existe uma redução de tempo polinomial de P_1 para P_2, o que se pode inferir sobre P_1? Justificar sua resposta.

Nesse caso pode-se concluir que $P_1 \in \mathcal{P}$, pois o seguinte algoritmo de tempo polinomial permite decidir P_1:

- Se f é a redução de tempo polinomial, calcular $f(w)$;
- Determinar se $f(w) \in P_2$;
- Em caso afirmativo, dar como saída SIM, caso contrário NÃO.

Como o tempo total para decidir P_1 é a combinação do tempo da redução (que é polinomial) com o tempo da decisão em P_2 (que é polinomial sobre uma cadeia de comprimento polinomial), segue que o mesmo resulta polinomial e portanto $P_1 \in \mathcal{P}$.