Quarto Teste

Universidade Federal de Goiás (UFG) - Regional Jataí
Bacharelado em Ciência da Computação
Teoria da Computação
Esdras Lins Bispo Jr.

21 de agosto de 2017

ORIENTAÇÕES PARA A RESOLUÇÃO

- A avaliação é individual, sem consulta;
- A pontuação máxima desta avaliação é 10,0 (dez) pontos, sendo uma das 06 (seis) componentes que formarão a média final da disciplina: quatro testes, uma prova e exercícios;
- A média final (MF) será calculada assim como se segue

$$\begin{aligned} MF &= MIN(10, S) \\ S &= (\sum_{i=1}^{4} 0,2.T_i) + 0,2.P + 0,1.EA + EB \end{aligned}$$

  em que

  - S é o somatório da pontuação de todas as avaliações,
  - T_i é a pontuação obtida no teste i,
  - P é a pontuação obtida na prova,
  - EA é a pontuação total dos exercícios de aquecimentos, e
  - EB é a pontuação total dos exercícios-bônus.
- O conteúdo exigido desta avaliação compreende o seguinte ponto apresentado no Plano de Ensino da disciplina: (3) Problemas Decidíveis, (4) Problemas Indecidíveis e (5) Complexidade de Tempo.

Nome:

Quarto Teste

1. (5,0 pt) Seja $TODAS_{AFD} = \{\langle A\rangle \mid A$ é um AFD e $L(A) = \Sigma^*\}$. Mostre que $TODAS_{AFD} \in \mathbf{P}$.

   Prova: Como passo auxiliar, o AFD B será construído de forma que $L(B) = \overline{L(A)}$ (Definição 1.16). É possível construir B a partir de A, invertendo os seus estados: se o estado é simples em A, torna-se final em B (e vice-versa).

   Para mostrar que $TODAS_{AFD} \in \mathbf{P}$, é necessário construir uma máquina de Turing simples (MTS) que a decide em tempo polinomial. Construiremos M que decide $TODAS_{AFD}$:

   M = "Sobre a entrada $\langle A\rangle$, em que A é um AFD, faça:

   (a) Construa o AFD B conforme apresentado anteriormente;

   (b) Marque o estado inicial de B.

   (c) Repita até que nenhum estado novo venha a ser marcado:

      i. Marque qualquer estado que tenha uma transição chegando nele a partir de qualquer estado que já está marcado.

   (d) Se nenhum estado de aceitação estiver marcado, *aceite*; caso contrário, *rejeite*.".

   O tempo de execução t de M é igual a soma do tempo de execução dos passos (a), (b), (c) e (d). Considere n como sendo o número de estados do AFD A. Logo, $t = O(n) + O(1) + O(n) \times O(n^2) + O(n) = O(n^3)$.

   3 é um número natural e $TODAS_{AFD} \in \text{TIME}(n^3)$. Logo, podemos afirmar que $TODAS_{AFD} \in \mathbf{P}$ ■

2. (5,0 pt) Mostre que **NP** é fechada sob operação de união.

Prova: Sejam A e B duas linguagens decidíveis em **NP**. Sejam M_A e M_B duas máquinas de Turing não-determinísticas que decidem as linguagens A e B, respectivamente (pois se uma linguagem é decidível, então uma máquina de Turing a decide). Como A e B são decidíveis em tempo polinomial não-determinístico, A e B pertencem a NTIME(n^k) e NTIME(n^l) respectivamente (em que k e l são números naturais). Iremos construir a máquina de Turing não-determinística M_{aux}, a partir de M_A e M_B, que decide $A \cup B$ em tempo polinomial não-determinístico. A descrição de M_{aux} é dada a seguir:

M_{aux} = "Sobre a entrada ω, faça:

(a) Rode M_A sobre ω.

(b) Rode M_B sobre ω.

(c) Se M_A e M_B rejeitam, *rejeite*. Caso contrário, *aceite*".

O tempo de execução t de M_{aux} é igual a soma do tempo de execução dos passos (a), (b) e (c). Logo, $t = O(n^k) + O(n^l) + O(1) = O(n^{max(k,l)})$.

Seja $c = max(k, l)$. Temos assim, $t = O(n^c)$. Como c é um número natural, $A \cup B \in$ NTIME(n^c) e, consequentemente, $A \cup B \in$ **NP**. Logo, podemos afirmar que **NP** é fechada sob a operação de união ■

Teoremas Auxiliares

Definição 1.16: Uma linguagem é chamada de uma linguagem regular se algum autômato finito a reconhece.

Teorema 1.25: A classe de linguagens regulares é fechada sob a operação de união.

Teorema 1.26: A classe de linguagens regulares é fechada sob a operação de concatenação.

Teorema 1.39: Todo autômato finito não-determinístico tem um autômato finito determinístico equivalente.

Teorema 1.49: A classe de linguagens regulares é fechada sob a operação estrela.

Teorema 1.54: Uma linguagem é regular se e somente se alguma expressão regular a descreve.

Definição 3.5: Chame uma linguagem de Turing-reconhecível se alguma máquina de Turing a reconhece.

Definição 3.6: Chame uma linguagem de Turing-decidível ou simplesmente decidível se alguma máquina de Turing a decide.

Teorema 3.13: Toda máquina de Turing multifita tem uma máquina de Turing que lhe é equivalente.

Teorema 3.16: Toda máquina de Turing não-determinística tem uma máquina de Turing determinística que lhe é equivalente.

Teorema 3.21: Uma linguagem é Turing-reconhecível se e somente se algum enumerador a enumera.

Teorema 4.1: A_{AFD} é uma linguagem decidível.

Teorema 4.2: A_{AFN} é uma linguagem decidível.

Teorema 4.3: A_{EXR} é uma linguagem decidível.

Teorema 4.4: V_{AFD} é uma linguagem decidível.

Teorema 4.5: EQ_{AFD} é uma linguagem decidível.

Teorema 4.9: Toda linguagem livre-de-contexto é decidível.

Teorema 4.11: A_{MT} é uma linguagem indecidível.

Definição 4.14: Um conjunto A é contável se é finito ou tem o mesmo tamanho que N.

Teorema 4.15: $\mathbb{Q}$ é contável.

Teorema 4.17: $\mathbb{R}$ é incontável.

Corolário 4.18: Algumas linguagens não são Turing-reconhecíveis.

Teorema 4.22: Uma linguagem é decidível sse ela é Turing-reconhecível e co-Turing-reconhecível.

Corolário 4.23: $\overline{A_{MT}}$ não é Turing-reconhecível.

Teorema 7.8: Seja $t(n)$ uma função, em que $t(n) \geq n$. Então toda máquina de Turing multifita de tempo $t(n)$ tem uma máquina de Turing de um única fita equivalente de tempo $O(t^2(n))$.

Teorema 7.11: Seja $t(n)$ uma função, em que $t(n) \geq n$. Então toda máquina de Turing não-determinística de uma única fita de tempo $t(n)$ tem uma máquina de Turing de um única fita equivalente de tempo $2^{O(t(n))}$.

Definição 7.12: **P** é a classe de linguagens que são decidíveis em tempo polinomial sobre uma máquina de Turing determinística de uma única fita. Em outras palavras, $\mathbf{P} = \bigcup_k \mathbf{TIME}\ (n^k)$.

Definição 7.19: **NP** é a classe das linguagens que têm verificadores de tempo polinomial.

Teorema 7.20: Uma linguagem está em **NP** sse ela é decidida por alguma máquina de Turing não-determinística de tempo polinomial. Em outras palavras, $\mathbf{NP} = \bigcup_k \mathbf{NTIME}\ (n^k)$.