

# Exame Modelo de Teoria da Computação

December 14, 2012

1. Defina os seguintes conjuntos:

- (a) O conjunto dos números inteiros divisíveis por 2 ou por 3.
- (b) Um certificado de habilitações regista o nome de um aluno, o número do aluno, o curso do aluno, e a lista das cadeiras realizadas com a respectiva nota. Indique qual o conjunto *CERT* com que pode modelar o conjunto de todos os certificados.
- (c) Defina a extensão da função

$$GETC : \wp(CERT) \times NAT \rightarrow (STRING \cup \{-1\})$$

que dado um conjunto de certificados e um número de aluno, devolve o nome de uma cadeira em que o aluno teve a pior nota, ou  $-1$  se o número de aluno não existir no conjunto.

2. Modele o seguinte sistema com uma estrutura. Uma fila de espera mantém informação sobre clientes à espera de um serviço.

- (a) Modele o conjunto de todos os estados possíveis da fila.
- (b) Modele (com uma função) a operação *enq* que adiciona um cliente na cauda da fila.
- (c) Modele (com uma função parcial) a operação *deq* que remove um cliente da frente da fila.
- (d) Modele (com uma função) a operação *empty* que devolve **true** se a fila está vazia e **false** caso contrário.

3. Considere o seguinte jogo para dois jogadores *A* e *B*. O jogador *A* pensa (sem o revelar) num número *S* entre 1 e 4. O jogador *B* tenta adivinhar esse número. O jogo funciona do seguinte modo. Em cada passo do jogo, o jogador *B* propõe um número *k* que julga que *A* pensou. Se adivinhou,

$A$  diz *Alooses*. Se  $S$  é maior que  $k$ , então  $A$  diz *high*, caso contrário  $A$  diz *low* e  $B$  joga de novo, até um máximo de 3 jogadas. Se  $B$  não adivinhar até à terceira jogada,  $A$  diz *Awins*. Cada jogo pode ser representado por uma palavra sobre o alfabeto

$$\Sigma = \{high, low, 1, 2, 3, 4, Alooses, Awins\}$$

Por exemplo, a palavra 3 *high* 2 *Alooses* descreve um jogo válido. Já a palavra 3 *low* 4 *Awins* descreve um jogo inválido.

Desenhe um NFA que representa a linguagem sobre  $\Sigma$  constituída pelas palavras que representam jogos válidos.

4. Considere o alfabeto da alínea anterior e a expressão regular

$$(((1 + 2 + 3)high) + ((2 + 3 + 4)low))^*$$

Defina um NFA e depois um DFA que reconheça a linguagem denotada por esta expressão regular. Justifique aplicando o método de construção de um DFA a partir de um NFA estudado.

5. Considere o alfabeto  $\Sigma = \{x, (, )\}$ . Baseado neste alfabeto definimos a gramática  $G = \langle V, \Sigma, E, R \rangle$  onde  $V = \{E, T\}$  e  $R$  contém as regras:

$$\begin{aligned} E &\rightarrow x \\ E &\rightarrow (E)T \\ T &\rightarrow \epsilon \\ T &\rightarrow xE \end{aligned}$$

- (a) Indique uma derivação para a palavra  $((x)x(x))$ .
- (b) Verifique se a gramática apresentada é LL(1) e em caso positivo indique a tabela de transições do analisador sintático determinista. Em caso negativo indique onde existe um conflito de determinismo.
6. Especifique uma SBTM (Stack Based Turing Machine) que executa a seguinte operação: dado um valor  $v$  qualquer guardado na célula de memória  $M_1$ , termina com o valor  $(v, v)$  em  $M_2$  e  $((v, v), v)$  em  $M_3$ .
7. Dada uma linguagem  $L$ , dizemos que uma SBTM reconhece  $L$  se iniciando uma computação com uma palavra  $m$  em  $M_1$  termina com **true** no topo da pilha se  $m \in L$  e termina com **false** no topo da pilha se  $m \notin L$ .

Discuta a seguinte afirmação, justificando a sua resposta: dado um NFA é sempre possível construir uma SBTM que reconhece a mesma linguagem reconhecida por esse NFA.