Teoria da Computação

Exame de Época Especial

Universidade da Beira Interior

2007-2008

A consulta dos apontamentos manuscritos e os apontamentos da disciplina (**e só esses**) é tolerada.
Proibido o uso de calculadora e de telemóvel.
Qualquer fraude implica reprovação na disciplina.
Só serão corrigidas as provas **legíveis**.

Relembramos que, na tradição da axiomática de Peano, a notação $\mathbb{N}$ refere-se ao conjunto dos naturais incluindo o 0. Referiremo-nos ao conjunto dos naturais sem o 0 por $\mathbb{N}^*$.

1 Princípios da Teoria da Computação

Uma técnica popular para a demonstração de indecidibilidade de problemas, a *técnica da redução*, consiste em exibir uma transformação do problema estudado para um problema conhecido por ser indecidível.

Se encontrar uma solução para um problema A pode se transformar na (ou pode equivaler numa) procura duma solução para o um problema B, então se B não tem solução algorítmica de certeza que A também não tem. O problema A é assim igualmente indecidível.

Neste contexto, é importante que a referida transformação seja ela própria decidível (tem de existir um algoritmo que a efectue).

Diga porque esta última condição é essencial. Sugestão: digo por exemplo o que aconteceria (em termos de conclusão por tirar sobre o problema A) se a transformação não pudesse ser um algoritmo.

2 Técnicas de Demonstração

Utilize o princípios da gaiola de pombos para demonstrar o seguinte teorema:

> Sejam $a_1 \dots a_n \in \mathbb{N}$, n inteiros naturais positivos distintos. Então existe sempre 2 destes valores cuja a diferença é divisível por $n-1$.

3 OCaml

Considere o tipo das árvores binárias:

```
type 'a bin_tree =
    Empty
  | Node of 'a bin_tree * 'a * 'a bin_tree
```

Defina uma função `igual: 'a bin_tree -> 'a bin_tree -> bool` que devolve $true$ se o seu primeiro argumento contém exactamente os mesmos elementos que o seu segundo argumento, $false$ senão.

4 Autómatos

Considere um autómato $M = \{Q, \Sigma, I, F, R_\delta\}$ não determinista com transições ϵ com $|F| > 1$. É sempre possível transformar um autómato como M num autómato não determinístico com transições ϵ equivalente M' possuindo um só estado final.

1. Proponha um algoritmo que realize tal transformação.
2. Demonstre (ou pelo menos dê um esboço de demonstração) que o autómato resultante M' é equivalente ao autómato M (ou seja que $L(M) = L(M')$). Esta propriedade é a propriedade de correcção do algoritmo proposto.

5 Autómatos de pilha

Defina um autómato com pilha que reconheça a linguagem $\{a^n.b^m.c^{n+m+p} \mid n, m, p \in \mathbb{N}\}$.

6 Máquinas de Turing

1. Diga que configuração atinge a execução da seguinte máquina $M = \{Q, \Gamma, \Sigma, \delta, q_0, \sharp, \emptyset\}$ sobre a palavra $abba$:
   - $Q = \{q_0, q_1, q_2, q_3\}$
   - $\Gamma = \{a, b, A, A', B, B'\}$
   - $\Sigma = \{a, b\}$
   - $\delta = \begin{array}{lllll} q_0 & a & \rightarrow & q_1 & A' & Direita \\ q_0 & b & \rightarrow & q_3 & B' & Direita \\ q_1 & a & \rightarrow & q_1 & a & Direita \\ q_1 & b & \rightarrow & q_1 & b & Direita \\ q_1 & A & \rightarrow & q_1 & A & Direita \\ q_1 & B & \rightarrow & q_1 & B & Direita \\ q_1 & \sharp & \rightarrow & q_2 & A & Esquerda \\ q_2 & a & \rightarrow & q_2 & a & Esquerda \\ q_2 & b & \rightarrow & q_2 & b & Esquerda \\ q_2 & A & \rightarrow & q_2 & A & Esquerda \\ q_2 & A' & \rightarrow & q_0 & A & Direita \\ q_2 & B & \rightarrow & q_2 & B & Esquerda \\ q_2 & B' & \rightarrow & q_0 & B & Direita \\ q_3 & a & \rightarrow & q_3 & a & Direita \\ q_3 & b & \rightarrow & q_3 & b & Direita \\ q_3 & A & \rightarrow & q_3 & A & Direita \\ q_3 & B & \rightarrow & q_3 & B & Direita \\ q_3 & \sharp & \rightarrow & q_2 & B & Esquerda \end{array}$