Teoria da Computação

Frequência

Duração: 2 horas

Universidade da Beira Interior

Quinta-Feira 10 de Janeiro de 2008

A consulta dos apontamentos manuscritos e os apontamentos da disciplina (**e só esses**) é tolerada.
É 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 ($\{1, 2, 3 \ldots\}$) por $\mathbb{N}^*$.

Os tempos de resolução presentes em cada alínea são meramente indicativos e tem por objectivo ajudar-vos a planear a resolução.

1 Princípios da Teoria da Computação

(10 minutos) As noções de indecidibilidade e de não computabilidade estão ligadas a noção de conjunto infinito. Explique esta relação.

2 Técnicas de Demonstração e OCaml

Considere o tipo *expr* das expressões aritméticas simples seguintes:

```
(* tipo expr onde:
   I = constante inteira , V = variável , A = +, S = -, M = *, D = /
*)
type expr = I of int | V of string | A of expr*expr
          | S of expr*expr | M of expr*expr | D of expr*expr

(* tipo dos ambientes (associação variável-valor)*)
type env = (string*int) list
```

Considere igualmente a função *eval* seguinte:

```
let rec eval (e:expr) (ambiente: env)=
  match e with
     I i -> i
```

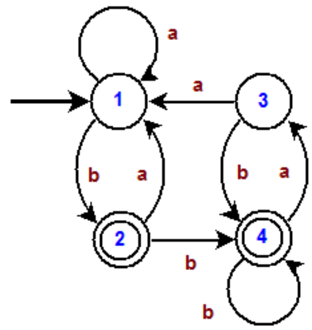

Figura 1: Autómato A_1

```
| V v → assoc v ambiente
| A (e1,e2) → eval e1 ambiente + eval e2 ambiente
| S (e1,e2) → eval e1 ambiente − eval e2 ambiente
| M (e1,e2) → eval e1 ambiente ∗ eval e2 ambiente
| D (e1,e2) → eval e1 ambiente / eval e2 ambiente
```

onde *assoc* é a função que devolve o valor inteiro associado a parâmetro v no ambiente *ambiente*, se este existir.

- (2 minutos) Qual é o princípio de indução associada a definição indutiva de *expr*?
- (15 minutos) Defina uma função `simplify : expr →expr` que execute sobre toda a estrutura do seu parâmetro as transformações seguintes:

  para uma qualquer expressão e, $\begin{array}{ll} e+0=e & e-0=e \\ e*1=e & e/1=e \\ e+e=2*e & e-e=0 \end{array}$

  Por exemplo a expressão $\frac{((x+0)+x)}{1}$ se simplifica em $2*x$ porque, pelas regras definidas, $\frac{((x+0)+x)}{1}$ se transforma em $((x+0)+x)$, $x+0$ se transforma em x e $x+x$ se transforma em $2*x$. Repare que a ordem de aplicação destas simplificações é irrelevante se todas elas são de facto executadas.
- (15 minutos) Demonstre por indução que $\forall e : expr, \forall a : env, eval\ (simplify\ e)\ a = eval\ e\ a$. Assuma para esse efeito e *se necessário* que o ambiente a tem todas as propriedades desejadas. Por exemplo, o ambiente tem todas as variáveis presentes na expressão considerada.

3 Autómatos Finitos e Linguagens regulares

1. (5 minutos) Considere o alfabeto $\Sigma = \{a, b\}$, defina um autómato *determinista* que reconheça a linguagem $\{w_1abw_2 \mid w_1, w_2 \in \{a, b\}^*\}$.
2. Considere o alfabeto $\Sigma = \{a, b\}$ e o autómato A_1 da figura 1:
   (a) (5 minutos) Minimize o autómato A_1.
   (b) (1 minuto) Olhando para o autómato minimal resultante, descreva de forma sucinta e informal (i.e. em português) a linguagem aceite por A_1.
3. (15 minutos) Considere o autómato A_2 da figura 2.

   Utilize o algoritmo de *Mac Naughton-Yamada* para inferir que expressão/linguagem regular aceita este autómato. Pretende-se aqui que apresente *somente* o resultado final (a expressão regular calculada) e o valor de $R(1, 3, 2)$.

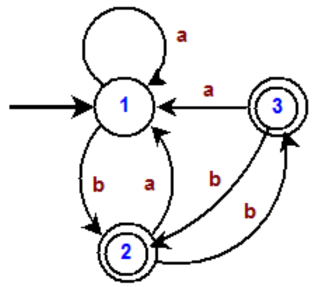

Figura 2: Autómato A_2

4. (10 minutos) Demonstre que a linguagem $\{a^{3^n} \mid n \in \mathbb{N}\}$ não é regular.

4 Gramáticas

1. (10 minutos) Defina uma gramática livre de contexto que gere a linguagem seguinte $\{a^n b^m \mid n, m \in \mathbb{N}, n \neq m\}$.

2. Considere a gramática cujas as produções são as seguintes:

$$\begin{array}{lcl} S & \rightarrow & AB \\ A & \rightarrow & a \\ A & \rightarrow & AS \\ B & \rightarrow & CD \\ C & \rightarrow & b \\ D & \rightarrow & c \end{array}$$

e a palavra $w = aabcbc$.

(a) (10 minutos) Utilize o algoritmo CYK para verificar que a palavra w é reconhecida pela gramática.

(b) (2 minutos) Dê a árvore de derivação de w.

5 Autómatos de pilha

(10 minutos) Defina um autómato com pilha que reconheça a linguagem $\{a^n.b^{3n+m}.c^m \mid n, m \in \mathbb{N}\}$. Sugestão: define um autómato que utilize Z como símbolo inicial de pilha.