

Licenciatura em Engenharia Informática e de Computadores - LEIC
Licenciatura em Engenharia de Redes de Comunicação e Informação -LERCI

Exercícios de Teoria da Computação

Computabilidade

Secção Ciência da Computação
Departamento de Matemática
Instituto Superior Técnico
2005/2006

1 Programas URM e funções URM-computáveis

1. Considere o programa URM

- 1 J(2,3,5)
- 2 S(1)
- 3 S(3)
- 4 J(1,1,1)

- (a) Faça o fluxograma do programa.
- (b) Emule a execução do programa a partir de uma configuração inicial em que
 - i. o conteúdo de R_1 é 4 e o conteúdo dos outros registros é 0;
 - ii. o conteúdo de R_1 é 5, o conteúdo de R_2 é 2 e o conteúdo dos outros registros é 0;
 - iii. o conteúdo de R_1 é 5, o conteúdo de R_2 é 3, o conteúdo de R_3 é 2 e o conteúdo dos outros registros é 0;
 - iv. o conteúdo de R_1 é 5, o conteúdo de R_2 é 3, o conteúdo de R_3 é 6 e o conteúdo dos outros registros é 0.
- (c) Indique as funções de aridade 1, 2 e 3 que são calculadas por este programa.

2. Considere o programa URM

- 1 J(1,2,6)
- 2 S(2)
- 3 S(3)
- 4 J(1,2,6)
- 5 J(1,1,2)
- 6 T(3,1)

- (a) Faça o fluxograma do programa.
- (b) Emule a execução do programa a partir de uma configuração inicial em que
 - i. o conteúdo de R_1 é 4 e o conteúdo dos outros registros é 0;
 - ii. o conteúdo de R_1 é 5, o conteúdo de R_2 é 2 e o conteúdo dos outros registros é 0;
 - iii. o conteúdo de R_1 é 5, o conteúdo de R_2 é 7 e o conteúdo dos outros registros é 0;
 - iv. o conteúdo de R_1 é 5, o conteúdo de R_2 é 3, o conteúdo de R_3 é 1 e o conteúdo dos outros registros é 0;
 - v. o conteúdo de R_1 é 5, o conteúdo de R_2 é 3, o conteúdo de R_3 é 4 e o conteúdo dos outros registros é 0.

- (c) Indique as funções de aridade 1, 2 e 3 que são calculadas por este programa.

3. Considere o programa URM

- 1 J(1,4,9)
- 2 S(3)
- 3 J(1,3,7)
- 4 S(2)
- 5 S(3)
- 6 J(1,1,3)
- 7 T(2,1)

- (a) Faça o fluxograma do programa.
- (b) Emule a execução do programa a partir de uma configuração inicial em que
- i. o conteúdo de R_1 é 3 e o conteúdo dos outros registros é 0;
 - ii. o conteúdo de R_1 é 4, o conteúdo de R_2 é 2 e o conteúdo dos outros registros é 0;
 - iii. o conteúdo de R_1 é 4, o conteúdo de R_2 é 7 e o conteúdo dos outros registros é 0;
 - iv. o conteúdo de R_1 é 5, o conteúdo de R_2 é 3, o conteúdo de R_3 é 1 e o conteúdo dos outros registros é 0;
 - v. o conteúdo de R_1 é 5, o conteúdo de R_2 é 3, o conteúdo de R_3 é 4 e o conteúdo dos outros registros é 0.
- (c) Indique as funções de aridade 1, 2 e 3 que são calculadas por este programa.

4. Considere o programa URM

- 1 J(1,3,6)
- 2 S(2)
- 3 S(2)
- 4 S(3)
- 5 J(1,1,1)
- 6 T(2,1)

- (a) Faça o fluxograma do programa.
- (b) Emule a execução do programa a partir de uma configuração inicial em que
- i. o conteúdo de R_1 é 3 e o conteúdo dos outros registros é 0;
 - ii. o conteúdo de R_1 é 4, o conteúdo de R_2 é 1 e o conteúdo dos outros registros é 0;

- iii. o conteúdo de R_1 é 4, o conteúdo de R_2 é 7 e o conteúdo dos outros registos é 0;
- iv. o conteúdo de R_1 é 5, o conteúdo de R_2 é 3, o conteúdo de R_3 é 1 e o conteúdo dos outros registos é 0;
- v. o conteúdo de R_1 é 5, o conteúdo de R_2 é 3, o conteúdo de R_3 é 4 e o conteúdo dos outros registos é 0.

(c) Indique as funções de aridade 1, 2 e 3 que são calculadas por este programa.

5. Faça o fluxograma correspondente a cada um dos programa URM seguintes e indique as funções de aridade 1, 2 e 3 que são calculadas por estes programas. Para cada um dos programas diga ainda se é ou não normalizado e indique o respectivo espaço de trabalho (ρ).

(a) 1 JUMP[2,3,5]
 2 SUCC[1]
 3 SUCC[3]
 4 JUMP[1,1,1]

(b) 1 JUMP[1,2,3]
 2 JUMP[1,1,4]
 3 SUCC[3]
 4 COPY[3,1]

(c) 1 JUMP[1,2,6]
 2 SUCC[2]
 3 SUCC[3]
 4 JUMP[1,2,6]
 5 JUMP[1,1,2]
 6 COPY[3,1]

(d) 1 JUMP[1,4,9]
 2 SUCC[3]
 3 JUMP[1,3,7]
 4 SUCC[2]
 5 SUCC[3]
 6 JUMP[1,1,3]
 7 COPY[2,1]

(e) 1 JUMP[1,3,6]
 2 SUCC[2]
 3 SUCC[2]
 4 SUCC[3]

5 JUMP[1,1,1]

6 COPY[2,1]

- (f) 1 JUMP[2,4,10]
2 JUMP[1,4,5]
3 SUCC[4]
4 JUMP[1,1,1]
5 JUMP[2,4,10]
6 JUMP[3,5,11]
7 SUCC[4]
8 SUCC[5]
9 JUMP[1,1,5]
10 SUCC[6]
11 COPY[6,1]

6. Prove que as seguintes funções em $[N_0^n \rightarrow N_0]$, $n \in N_0$, são URM-computáveis. Não use oráculos, a menos que expressamente indicado.

- (a) $h(x) = sg(x)$ (se $x = 0$ então 0 senão 1)
(b) $h(x) = \underline{sg}(x)$ (se $x \neq 0$ então 0 senão 1)
(c) $h(x, y) = x + y$
(d) $h(x, y) = max(x, y)$
(e) $h(x, y) = min(x, y)$
(f) $h(x) = x \dot{-} 1$ (x múnus 1, i.e., se $x \geq 1$ então $x - 1$ senão 0)
(g) $h(x) =$ se x é múltiplo de 3 então $x/3$ senão 0
(h) $h(x) =$ quociente da divisão inteira de x por 2
(i) $h(x, y) = x \dot{-} y$ (x múnus y , i.e., se $x \geq y$ então $x - y$ senão 0)
(j) $h(x, y) = x - y$ se $x \geq y$ e $h(x, y)$ não está definida, caso contrário (note que este caso a execução do programa que calcula h deve terminar se e só $x \geq y$).
(k) $h(x, y) = |x - y|$
(l) $h(x, y) = x \times y$
i. usando apenas o oráculo SUM
ii. sem oráculos
(m) $h(x, y) = x^y$
i. usando apenas o oráculo PROD
ii. usando apenas o oráculo SUM
iii. sem oráculos
(n) $h(x) = \sum_{1 \leq i \leq x} i$
i. usando apenas os oráculos PROD e LESS

- ii. usando apenas o oráculo SUM
 - iii. usando apenas o oráculo PROD
 - iv. sem oráculos
- (o) $h(x, y) = z$ onde z é o quociente da divisão inteira de x por y , se y é diferente de 0, e $h(x, y)$ não está definido se $y = 0$:
- i. usando apenas o oráculo SUB
 - ii. usando apenas o oráculo SUB1
 - iii. sem oráculos.

Note que, neste caso, a execução do programa que calcula h deve terminar se e só $y \neq 0$.

- (p) $h(x, y) = z$ onde z é o quociente da divisão inteira de x por y , se y é diferente de 0, e é 0 se $y=0$:
- i. usando apenas o oráculo SUB
 - ii. usando apenas o oráculo SUB1
 - iii. sem oráculos.

- (q) $h(x, y) = z$ onde z é o resto da divisão inteira de x por y , se y é diferente de 0, e $h(x, y)$ não está definido se $y = 0$:
- i. usando apenas o oráculo SUB
 - ii. usando apenas o oráculo SUB1
 - iii. sem oráculos.

Note que, neste caso, a execução do programa que calcula h deve terminar se e só $y \neq 0$.

- (r) $h(x, y) = z$ onde z é o resto da divisão inteira de x por y , se y é diferente de 0, e é x se $y=0$:
- i. usando apenas o oráculo SUB
 - ii. usando apenas o oráculo SUB1
 - iii. sem oráculos.

- (s) $h(x) = y$ onde y é a raiz quadrada inteira de x . Só pode usar os oráculos LESS e PROD.

- (t) $h(x, y) = z$ onde z é o menor múltiplo comum a x e a y , se $x > 0$ e $y > 0$, e é 0 caso contrário. Só pode usar os oráculos LESS e PROD.

- (u) $h(x, y) = z$ onde z é o expoente do y -ésimo primo na factorização prima de x se x e y são ambos distintos de 0 e é 0 caso contrário. Só pode usar os oráculos PRIMEQ, PROD, MOD e LESS.

- (v) $h(x, y) = z$ onde z é o o logaritmo inteiro de x na base y se x é maior que 0 e y é maior que 1 e é 0 caso contrário:
- i. usando apenas os oráculos PROD e LESS
 - ii. usando apenas os oráculos SUM e LESS.

- (w) $h(x) = y$ onde y é o número de ocorrências em x do maior dígito de x . Só pode usar os oráculos DIGN, DEZ e LESS.

- (x) $h(x) = y$ onde y é o número que se obtém eliminando de x os dígitos que surgem em posições pares, assumindo que o primeiro dígito é o das unidades, o segundo o das dezenas, etc. Por exemplo, $h(635798) = 378$. Só pode usar os oráculos SUM, DEZ, PROD, DIGN.
- (y) $h(x) = 1$ se x é um palíndromo e $h(x) = 0$ caso contrário:
 - i. usando apenas os oráculos SUB1, DEZ, LESS, DIGN
 - ii. usando apenas os oráculos SUM, DEZ, PROD, DIGN.
- (z) $h(x) = y$ onde y é o número que é obtido invertendo a ordem dos dígitos de x . Por exemplo, $h(345) = 543$. Só pode usar os oráculos SUM, DEZ, PROD, DIGN.

7. Prove que os seguintes predicados sobre os naturais são URM-decidíveis. Não use oráculos, a menos que expressamente indicado.

- (a) $P(x) = \text{“}x \text{ é igual a } 3\text{”}$
- (b) $P(x) = \text{“}x \text{ é diferente de } 3\text{”}$
- (c) $P(x) = \text{“}x \text{ é maior que } 3\text{”}$
- (d) $P(x) = \text{“}x \text{ é menor ou igual a } 3\text{”}$
- (e) $P(x, y) = \text{“} x \text{ é igual a } y\text{”}$
- (f) $P(x, y) = \text{“} x \text{ é diferente de } y\text{”}$
- (g) $P(x, y) = \text{“} x \text{ é menor que } y\text{”}$
- (h) $P(x, y) = \text{“} x \text{ é maior que } y\text{”}$
- (i) $P(x, y) = \text{“}x \text{ é maior ou igual que } y\text{”}$
- (j) $P(x) = \text{“}x \text{ é par”}$
- (k) $P(x) = \text{“}x \text{ é ímpar”}$
- (l) $P(x) = \text{“}x \text{ é múltiplo de } 3\text{”}$
- (m) $P(x) = \text{“}x \text{ não é múltiplo de } 3\text{”}$
- (n) $P(x, y) = \text{“}x \text{ é múltiplo de } y\text{”}$:
 - i. usando apenas o oráculo SUM
 - ii. sem oráculos.
- (o) $P(x, y) = \text{“}x \text{ é divisor de } y\text{”}$:
 - i. usando apenas o oráculo SUB
 - ii. usando apenas os oráculos SUM e LESS
 - iii. usando apenas os oráculos PROD e LESS
 - iv. sem oráculos.
- (p) $P(x) = \text{“}x \text{ é quadrado perfeito”}$:
 - i. usando apenas os oráculos PROD e LESS
 - ii. usando apenas os oráculos SUM e LESS.
- (q) $P(x) = \text{“}x \text{ é número primo”}$

- i. usando apenas os oráculos PRIMEN e LESS
 - ii. usando apenas o oráculo MOD
 - iii. sem oráculos.
- (r) $P(x, y) =$ “ x e y são primos entre si”
- i. usando apenas o oráculo MOD
 - ii. sem oráculos.
8. Mostre que o comando $T(m, n)$ é redundante, isto é, que pode ser definido como um programa à custa dos outros comandos URM.
9. Considere o seguinte comando de salto, $K_q(m, n, p)$, com $q \in \mathbb{N}$, cuja execução é a seguinte:
- se $r_m = r_n$ e $r_p \leq q$ a execução prossegue na linha r_p , se esta existir no programa, ou termina, em caso contrário;
 - se $r_m = r_n$ e $r_p > q$ a execução termina;
 - se $r_m \neq r_n$ a execução do programa prossegue na linha seguinte.

Mostre que este comando pode ser definido como um programa usando os comandos URM.

10. Considere os autómatos finitos deterministas $D = (Q, I, \delta, q_0, F)$ seguintes e escreva um programa URM que calcule a função característica do predicado $x \in L_D$, usando uma representação numérica apropriada para as palavras $x \in I^*$:
- (a) $D = (Q, I, \delta, q_0, F)$ com
 - $Q = \{q_0, q_1\}$,
 - $I = \{a, b, c\}$,
 - $F = \{q_1\}$,
 - $\delta(q_0, a) = q_1$, $\delta(q_0, b) = q_1$, $\delta(q_1, c) = q_0$.
 - (b) D é um autômato mínimo para a linguagem sobre $I\{a, b\}$ constituída pelas palavras que têm um número par de b 's.
 - (c) D é um autômato mínimo para a linguagem sobre $\{a, b\}$ constituída pelas palavras que começam por a .
 - (d) D é um autômato mínimo para a linguagem sobre $\{a, b\}$ constituída pelas palavras que têm pelo menos um b .
 - (e) D é um autômato mínimo para a linguagem sobre $\{a, b\}$ constituída pelas palavras que começam e terminam em a .
 - (f) D é um autômato mínimo para a linguagem sobre $\{a, b, c\}$ constituída pelas palavras que têm um número ímpar de b 's.
 - (g) D é um autômato mínimo para a linguagem sobre $\{a, b, c\}$ constituída pelas palavras que terminam em b e têm pelo menos um a .
 - (h) D é um autômato mínimo para a linguagem sobre $\{a, b, c\}$ constituída pelas palavras que não têm dois c 's consecutivos.

2 Godelização de programas URM

1. Sendo $\pi : \mathbb{N}_0 \times \mathbb{N}_0 \rightarrow \mathbb{N}_0$, $\xi : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}_0$, $\beta : \mathbb{I} \rightarrow \mathbb{N}_0$ e $\tau : \bigcup_{k \in \mathbb{N}} (\mathbb{N}_0^k) \rightarrow \mathbb{N}_0$ as bijecções estudadas, calcule:
 - (a) $\pi(1, 0)$, $\pi(2, 4)$, $\pi(3, 2)$, $\pi(4, 2)$ e $\pi(3, 5)$
 - (b) $\pi^{-1}(4)$, $\pi^{-1}(31)$, $\pi^{-1}(39)$, $\pi^{-1}(47)$, $\pi^{-1}(72)$, $\pi^{-1}(83)$ $\pi^{-1}(103)$ e $\pi^{-1}(83)$
 - (c) $\xi(1, 1, 1)$, $\xi(1, 2, 1)$, $\xi(3, 1, 3)$ e $\xi(3, 4, 5)$
 - (d) $\xi^{-1}(19)$, $\xi^{-1}(39)$, $\xi^{-1}(103)$, $\xi^{-1}(159)$ e $\xi^{-1}(191)$
 - (e) $\beta(\text{ZERO}[1])$, $\beta(\text{ZERO}[5])$, $\beta(\text{SUCC}[3])$, $\beta(\text{SUCC}[6])$, $\beta(\text{COPY}[2, 11])$, $\beta(\text{COPY}[3, 4])$, $\beta(\text{JUMP}[1, 1, 1])$ e $\beta(\text{JUMP}[2, 1, 2])$
 - (f) $\beta^{-1}(8)$, $\beta^{-1}(12)$, $\beta^{-1}(13)$, $\beta^{-1}(25)$, $\beta^{-1}(34)$, $\beta^{-1}(39)$, $\beta^{-1}(58)$ e $\beta^{-1}(79)$
 - (g) $\tau(1, 1, 1)$, $\tau(2, 0, 1)$, $\tau(3, 2, 1)$, $\tau(1, 2, 3)$, $\tau(0, 2, 1)$ e $\tau(1, 3, 0)$
 - (h) $\tau^{-1}(8)$, $\tau^{-1}(13)$, $\tau^{-1}(24)$, $\tau^{-1}(41)$, $\tau^{-1}(71)$, $\tau^{-1}(129)$ e $\tau^{-1}(241)$

2. Indique o código do programa URM:

- (a) $\langle \text{ZERO}[1] \rangle$
- (b) $\langle \text{SUCC}[1] \rangle$
- (c) $\langle \text{COPY}[2,1] \rangle$
- (d) $\langle \text{ZERO}[1], \text{SUCC}[1], \text{SUCC}[1] \rangle$
- (e) $\langle \text{COPY}[2,1], \text{SUCC}[1], \text{SUCC}[1] \rangle$
- (f) $\langle \text{SUCC}[2], \text{SUCC}[2], \text{COPY}[2,1] \rangle$
- (g) $\langle \text{JUMP}[1,2,2] \rangle$
- (h) $\langle \text{JUMP}[2,1,1], \text{SUCC}[1] \rangle$
- (i) $\langle \text{JUMP}[2,3,5], \text{SUCC}[1], \text{SUCC}[3], \text{JUMP}[1,1,1] \rangle$
- (j) $\langle \text{JUMP}[1,2,3], \text{JUMP}[1,1,4], \text{SUCC}[3], \text{COPY}[3,1] \rangle$

3. Indique os programas URM cujos códigos são:

- (a) 0
- (b) 1
- (c) 2
- (d) 3
- (e) 4
- (f) 5
- (g) 6
- (h) 7
- (i) 8

- (j) 9
- (k) 10
- (l) 65
- (m) 503
- (n) 100
- (o) 8234
- (p) 169
- (q) 123
- (r) 321
- (s) 231
- (t) 127
- (u) 255
- (v) 4159
- (w) 513
- (x) 700
- (y) 4159
- (z) 8197

4. Calcule ϕ_7 , W_7 e E_7

5. Calcule ϕ_{20} , W_{20} e E_{20}

6. Calcule ϕ_{68} , W_{68} e E_{68}

7. Calcule

(a) ϕ_{159} , W_{159} e E_{159}

(b) ϕ_{159}^2 , W_{159}^2 e E_{159}^2

8. Calcule

(a) ϕ_{639} , W_{639} e E_{639}

(b) ϕ_{639}^2 , W_{639}^2 e E_{639}^2

9. Calcule

(a) ϕ_{1343} , W_{1343} e E_{1343}

(b) ϕ_{1343}^2 , W_{1343}^2 e E_{1343}^2

10. Calcule

(a) ϕ_{4127} , W_{4127} e E_{4127}

(b) ϕ_{4127}^2 , W_{4127}^2 e E_{4127}^2

11. Calcule

(a) ϕ_{8224} , W_{8224} e E_{8224}

(b) ϕ_{8224}^2 , W_{8224}^2 e E_{8224}^2