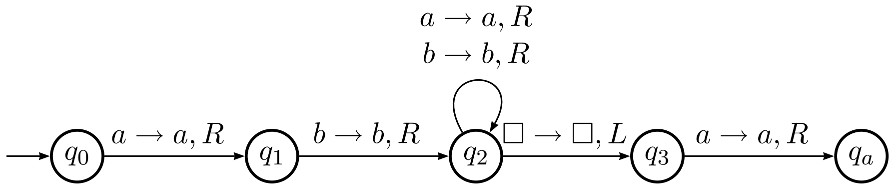
Instituto Superior Técnico
Teoria da Computação - LEIC 2013/2014
Aula prática 1

Nota: Na sequência o símbolo $\square$ representa o símbolo de registo vazio.

1 Máquinas de Turing

1. Considere a máquina de Turing com alfabeto $\Sigma = \{a, b\}$ e alfabeto da fita $\Gamma = \{a, b, \square\}$, cuja representação através de um grafo é a seguinte (usando a convenção de que transições para o estado de rejeição são omitidas):

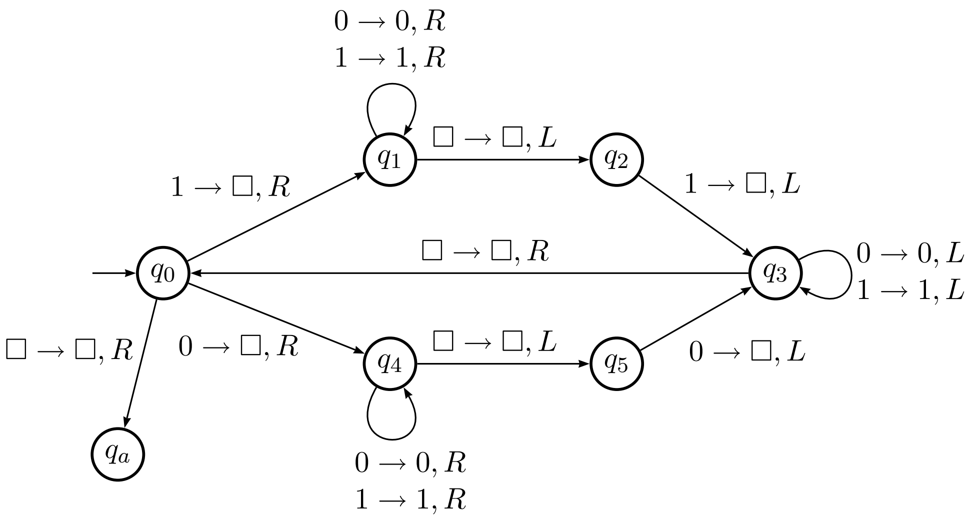


   (a) Descreva a evolução de M a partir da configuração inicial relativa a

      i. aab ii. aaa iii. aba

   (b) Verifique informalmente que a linguagem reconhecida por M, ou linguagem de M, é o conjunto das palavras sobre Σ que começam em ab e terminam em a.

2. Considere a máquina de Turing com alfabeto $\Sigma = \{0, 1\}$ e alfabeto da fita $\Gamma = \{0, 1, \square\}$, cuja representação através de um grafo é a seguinte (usando a convenção de que transições para o estado de rejeição são omitidas):

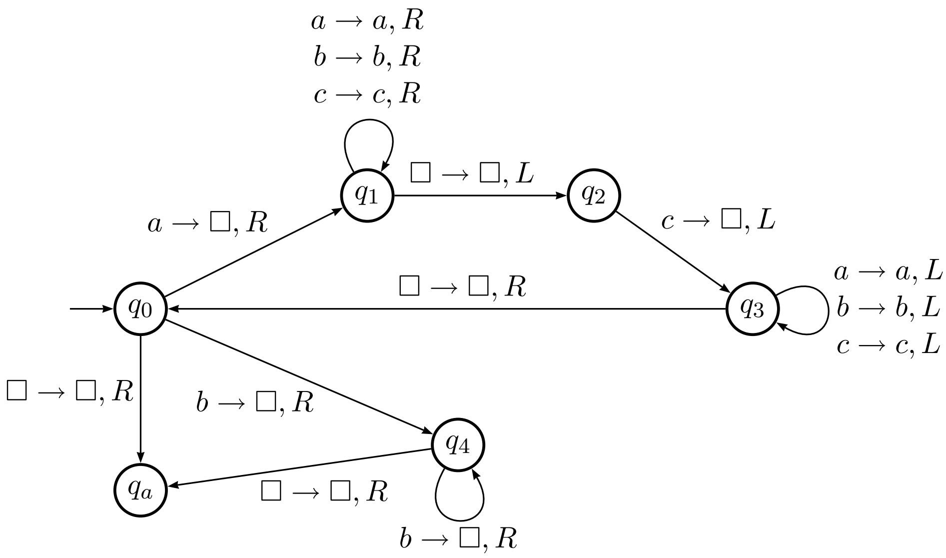


   (a) Descreva a evolução de M a partir da configuração inicial relativa a

      i. 001 ii. 0110 iii. 010 iv. 100001

(b) Verifique informalmente que a linguagem reconhecida por M, ou linguagem de M, é o conjunto das palavras sobre Σ do tipo ww^R (recorde que w^R é a palavra w reflectida).

3. Considere a máquina de Turing com alfabeto $\Sigma = \{a, b\}$ e alfabeto da fita $\Gamma = \{a, b, \Box\}$, cuja representação através de um grafo é a seguinte (usando a convenção de que transições para o estado de rejeição são omitidas):



(a) Descreva a evolução de M a partir da configuração inicial relativa a

i. ac ii. abb iii. $abbc$ iv. $aabcc$

(b) Verifique informalmente que a linguagem reconhecida por M, ou linguagem de M, é o conjunto das palavras sobre Σ do tipo $a^n b^m c^n$ com $m, n \in \mathbb{N}_0$.

2 Especificação de máquinas de Turing

1. Especifique uma máquina de Turing que reconheça as seguintes linguagens sobre o alfabeto $\{0, 1\}$

(a) a linguagem das palavras que têm pelo menos um 0.

(b) a linguagem das palavras que começam e terminam em 1.

(c) a linguagem das palavras cujo comprimento é par.

(d) a linguagem das palavras que têm um número ímpar de 0's.

(e) a linguagem das palavras do tipo $0^n 1^n$ com $n \in \mathbb{N}_0$.

(f) a linguagem das palavras do tipo $0^n 1^{n+1}$ com $n \in \mathbb{N}_0$.

(g) a linguagem das palavras do tipo $0^{2n} 1^n$ com $n \in \mathbb{N}_0$.

(h) a linguagem das palavras do tipo $0^n 1^m$ com $n \in \mathbb{N}_0$ e $n > m$.

(i) a linguagem das palavras do tipo $0^n 1^m$ com $n \in \mathbb{N}_0$ e $n < m$.

2. Especifique uma máquina de Turing que reconheça as seguintes linguagens (linguagens de certificados da adição)

   (a) $L = \{a^n b^m c^{n+m} : \ n, m \in \mathbb{N}_0\}$.
   (b) $L = \{a^m + a^n = a^{m+n} : \ n, m \in \mathbb{N}_0\}$.
   (c) $L = \{(0^n, 0^m, 0^{n+m}) : \ n, m \in \mathbb{N}_0\}$.

3. Especifique uma máquina de Turing que reconheça as seguintes linguagens (linguagens de certificados da multiplicação por 2)

   (a) $L = \{a^n b^{2n} : \ n, m \in \mathbb{N}_0\}$.
   (b) $L = \{a^n \times 2 = a^{2n} : \ n \in \mathbb{N}_0\}$.
   (c) $L = \{(0^n, 0^{2n}) : \ n \in \mathbb{N}_0\}$.

4. Especifique uma máquina de Turing que reconheça as seguintes linguagens (linguagens de certificados da multiplicação por 3)

   (a) $L = \{a^n b^{3n} : \ n, m \in \mathbb{N}_0\}$.
   (b) $L = \{a^n \times 3 = a^{3n} : \ n \in \mathbb{N}_0\}$.
   (c) $L = \{(0^n, 0^{3n}) : \ n \in \mathbb{N}_0\}$.

NOTA: Após a aula prática os alunos deverão tentar resolver todos os exercícios que não foram resolvidos na aula. Se tiverem dificuldades ou dúvidas deverão consultar os docentes da disciplina durante os respectivos horários de dúvidas.

Instituto Superior Técnico
Teoria da Computação - LEIC 2013/2014
Aula prática 2

1 Especificação de máquinas de Turing (cont)

1. Especifique um classificador para as seguintes linguagens

   (a) a linguagem sobre $\{0,1\}$ das palavras que são palíndromos (recorde que w^R denota a palavra w invertida, e que uma palavra w é um palíndromo se que $w = w^R$).

   (b) a linguagem das palavras sobre $\{0,1,2\}$ do tipo $0^n 1^n 2^n$ com $n \in \mathbb{N}_0$.

2. Especifique uma máquina de Turing que decida as seguintes linguagens

   (a) $L = \{a^n b^{n \div 2} : \ n \in \mathbb{N}_0\}$.

   (b) $L = \{a^n \div 2 = a^{n \div 2} : \ n \in \mathbb{N}_0\}$.

   (c) $L = \{(0^n, 0^{n \div 2}) : \ n \in \mathbb{N}_0\}$.

   onde $n \div 2$ é o quociente da divisão inteira por 2

3. Mostre que são decidíveis as seguintes linguagens

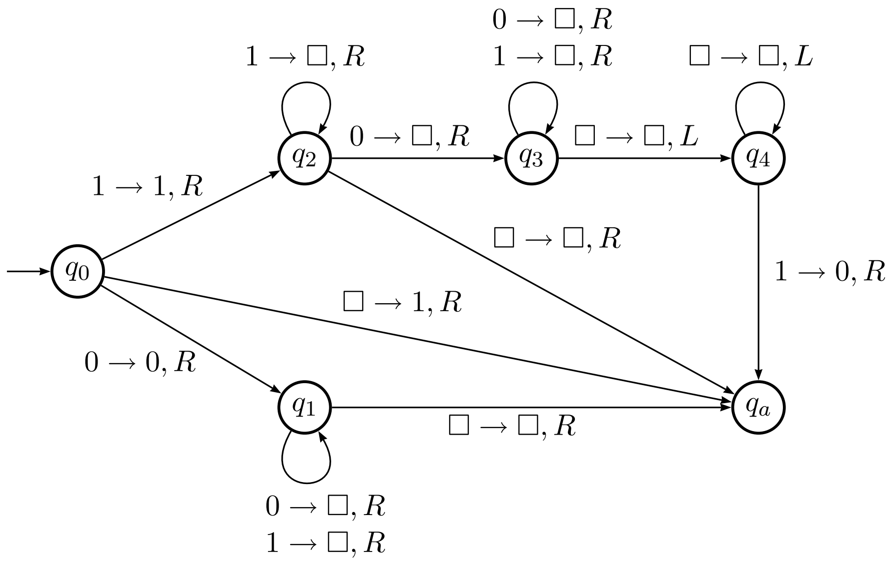
   (a) a linguagem das palavras sobre $\{0,1\}$ com igual número de 0's e de 1's.

   (b) a linguagem das palavras sobre $\{0,1\}$ nas quais o número de 0's é o dobro do números de 1's.

   (c) a linguagem das palavras sobre $\{0,1,2\}$ do tipo $w2w$ com $w \in \{0,1\}^*$.

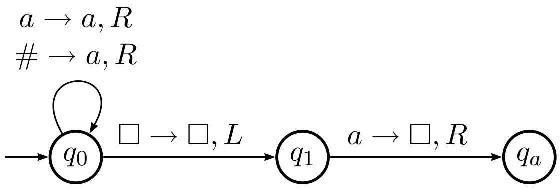
2 Cálculo de funções com máquinas de Turing)

1. Considere a máquina de Turing $M = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$ com $\Sigma = \{0,1\}$, $\Gamma = \{0,1,\Box\}$ e cuja representação através de um grafo é



Verifique informalmente que M calcula a função $\text{AND}: \{0,1\}^* \to \{0,1\}$ dada por $\text{AND}(w) = 0$ se em w existe pelo menos um 0 e $\text{AND}(w) = 1$ caso contrário.

2. Considere a máquina de Turing $M = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$ com $\Sigma = \{a, \#\}$, $\Gamma = \{a, \#, \square\}$ e cuja representação através de um grafo é



Verifique informalmente que M calcula a função $f: \{a\}^* \times \{a\}^* \to \{a\}^*$ dada por $f(a^n, a^m) = a^{m+n}$.

3. Especifique uma máquina de Turing que calcule $f : \{a, b\}^* \to \{0, 1\}$ tal que $f(v) = 1$ se v tem comprimento par e $f(v) = 0$ caso contrário.

4. Considere a função $\text{OR} : \{0,1\}^* \to \{0,1\}$ tal que $\text{OR}(w) = 1$ se em w existe pelo menos um 1 e $\text{OR}(w) = 0$ caso contrário. Especifique uma máquina de Turing que calcule a função OR.

5. Considere a função $\text{XOR} : \{0,1\}^* \to \{0,1\}$ tal que $\text{XOR}(w) = 1$ se em w existe um e um só 1 e $\text{XOR}(w) = 0$ caso contrário. Especifique uma máquina de Turing que calcule a função XOR.

6. Especifique uma máquina de Turing que calcule a função *sucessor*, isto é, a função que a cada número natural faz corresponder o seu sucessor (notação unária: cada natural n é representado pela palavra 1^n).

7. Especifique uma máquina de Turing que calcule a função *dobro*, isto é, a função que a cada número natural faz corresponder o seu dobro (notação unária: cada natural n é representado pela palavra 1^n).

8. Especifique uma máquina de Turing que calcule a função *triplo*, isto é, a função que a cada número natural faz corresponder o seu triplo (notação unária: cada natural n é representado pela palavra 1^n).

9. Especifique uma máquina de Turing que calcule a função que a cada número natural faz corresponder o resto da sua divisão inteira por 2 (notação unária: cada natural n é representado pela palavra 1^n).

10. Especifique uma máquina de Turing que calcule a função que a cada número natural faz corresponder o resto da sua divisão inteira por 3 (notação unária: cada natural n é representado pela palavra 1^n).

11. Usando notação binária para representar os naturais especifique uma máquina de Turing que calcule

    (a) a função *sucessor*

    (b) a função que a cada natural faz corresponder 1 se ele é par e 0 caso contrário.

12. Especifique uma máquina de Turing que calcule a função $f : \{a, b, c\}^* \to \{0, 1\}^*$ dada por $f(w) = 0$ se w começa por z ou é a palavra vazia, e $f(w) = 1^{n+m}$ onde n é o número de a's que ocorrem em w e m é o número de b's que ocorrem em w.

NOTA: Após a aula prática os alunos deverão tentar resolver todos os exercícios que não foram resolvidos na aula. Se tiverem dificuldades ou dúvidas deverão consultar os docentes da disciplina durante os respectivos horários de dúvidas.

Instituto Superior Técnico
Teoria da Computação - LEIC 2013/2014
Aula prática 3

1 Cálculo de funções com máquinas de Turing (cont)

1. Especifique uma máquina de Turing que calcule $f : \{a, b\}^* \to \{a, b, \#\}^*$ tal que $f(w) = w\#w$ (isto é, a máquina faz uma cópia do conteúdo da fita na configuração inicial, usando o símbolo # para separar a cópia da palavra original) de modo a que nas configurações de aceitação o dispositivo de leitura se encontre no primeiro registo.

2. Especifique uma máquina de Turing que calcule a função que a cada par de números naturais n e m faz corresponder o natural $n - m$ se $n > m$ e 0 caso contrário (notação unária: cada natural k é representado pela palavra 1^k).

3. Especifique uma máquina de Turing que calcule a função que a cada par de números naturais n e m faz corresponder o natural $n - m$ se $n > m$ e $m - n$ caso contrário (notação unária: cada natural k é representado pela palavra 1^k).

2 Especificação de máquinas de Turing multifita

1. Especifique uma máquina de Turing de 2 fitas que decida a linguagem $\{a^n b^n : n \in \mathbb{N}_0\}$ sobre o alfabeto $\{a, b\}$.

2. Especifique uma máquina de Turing de 2 fitas que decida a linguagem das palavras sobre $\{a, b\}$ que são palíndromos.

3. Especifique uma máquina de Turing de 2 fitas que decida a linguagem das palavras sobre o alfabeto $\{a, b, \#\}$ do tipo $w\#w$ com $w \in \{0, 1\}^*$.

4. Especifique uma máquina de Turing de 2 fitas que decida a linguagem $\{a^n b^n c^n : n \in \mathbb{N}_0\}$ sobre o alfabeto $\{a, b, c\}$. Repita o exercício, mas agora com uma máquina de 3 fitas.

5. Especifique uma máquina de Turing de 2 fitas que decida a linguagem das palavras sobre o alfabeto $\{a, b\}$ nas quais o números de a's é igual ao número de b's. Repita o exercício, mas agora com uma máquina de 3 fitas.

6. Especifique uma máquina de Turing de 3 fitas que decida a linguagem das palavras sobre o alfabeto $\{0, 1, 2\}$ cujo comprimento é maior ou igual que a soma dos dígitos que as constituem.

7. Especifique uma máquina de Turing de 2 fitas que decida a linguagem $\{ww : w \in \{0, 1\}^*\}$ sobre o alfabeto $\{0, 1\}$.

8. Especifique uma máquina de Turing com 2 fitas que calcule a função referida no exercício 1.3. Na configuração de aceitação o resultado pretendido deve estar no início da fita 1, e os restantes registos desta fita devem estar vazios. Repita o exercício, mas agora com uma máquina de 3 fitas.

9. Especifique uma máquina de Turing com 2 fitas que converta um natural em notação unária nesse natural em notação binária, isto é, especifique uma máquina de Turing com 2 fitas que partindo da configuração inicial relativa a 1^n chegue a uma configuração de aceitação na qual a representação binária de n está no início da fita 1 (os restantes registos desta fita devem estar vazios).

10. Especifique uma máquina de Turing com 2 fitas que partindo da configuração inicial relativa a 1^n chegue a uma configuração de aceitação na qual a palavra 1^{2^n} está no início da fita 2, e os restantes registos desta fita devem estar vazios.

NOTA: Após a aula prática os alunos deverão tentar resolver todos os exercícios que não foram resolvidos na aula. Se tiverem dificuldades ou dúvidas deverão consultar os docentes da disciplina durante os respectivos horários de dúvidas.

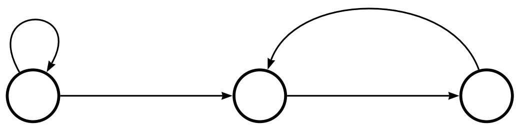
Instituto Superior Técnico
Teoria da Computação - LEIC 2013/2014
Aula prática 4

1 Especificação de máquinas multifita (cont)

Um grafo (orientado) pode ser ser representado por uma palavra sobre o alfabeto $\{0, 1, (,)\}$, por exemplo. Cada nó pode ser representado por uma palavra 1^k, em que podemos assumir que $k \in \mathbb{N}$. O conjunto dos nós pode ser representado por um tuplo constituído pelas representações dos nós, separadas por 0 (em vez de ,). Uma seta com origem no nó (representado por) 1^k e com destino no nó (representado por) 1^m pode ser representada por (1^k01^m). Finalmente, o conjunto das setas pode ser representado por um tuplo constituído pelas representações das setas, separadas por 0. Por exemplo

$$(10110111)((101)0(1011)0(110111)0(111011))$$

representa o seguinte grafo com 3 nós e 4 setas:



Nota: no que se segue, sendo w uma palavra sobre um alfabeto, usa-se $w\Box\Box\ldots$ para afirmar que o conteúdo de uma dada fita de uma máquina de Turing é o seguinte: nos primeiros registos está a palavra w (um símbolo em cada registo) e todos os outros registos estão vazios.

1. Especifique uma máquina de Turing com 2 ou mais fitas que decida a linguagem $L_{\text{nós}}$ sobre $\{0, 1, (,)\}$ das palavras que representam conjuntos de nós de grafos, como acima definido, em que não há nós repetidos.

2. Especifique uma máquina de Turing com 2 ou mais fitas que decida a linguagem L_{setas} sobre $\{0, 1, (,)\}$ das palavras que representam conjuntos de setas de grafos, como acima definido, em que não há setas repetidas.

3. Seja L_{grafos} a linguagem sobre $\{0, 1, (,)\}$ constituída pelas palavras wv, com $w \in L_{\text{nós}}$ e $v \in L_{\text{setas}}$ (ver exercícios 1 e 2), nas quais todos os nós origem e destino das setas em v estão no conjunto de nós w. Especifique uma máquina de Turing com 2 ou mais fitas que, partindo de uma configuração inicial relativa a uma palavra wv com $w \in L_{\text{nós}}$ e $v \in L_{\text{setas}}$, chega a uma configuração de aceitação em que o conteúdo da fita 1 é $1\Box\Box\ldots$ se $wv \in L_{\text{grafos}}$, e, em caso contrário, chega a uma configuração de aceitação em que o conteúdo da fita 1 é $0\Box\Box\ldots$.

4. Especifique uma máquina de Turing com 2 ou mais fitas que, partindo de uma configuração inicial relativa a uma palavra $\alpha \in L_{\text{grafos}}$ (ver exercício 3), chega a uma configuração de aceitação em que no início da fita 1 está o conjunto dos nós de α que são origem de alguma seta de α, e os outros registos estão vazios. Repita o exercício mas agora para o conjunto dos nós de α que são destino de alguma seta.

5. Diz-se que um nó ν de um grafo orientado é isolado se ν não é origem de uma seta cujo destino é um nó distinto de ν, nem destino de uma seta cuja origem é um nó distinto de ν.

   (a) Especifique uma máquina de Turing com 2 ou mais fitas que, partindo de uma configuração inicial relativa a uma palavra $\alpha\#\nu$ onde $\alpha \in L_{\text{grafos}}$ (ver exercício 3) e ν é um nó de α, chega a uma configuração de aceitação em que o conteúdo da fita 1 é $1\square\square\ldots$ se ν é nó isolado e, em caso contrário, chega a uma configuração de aceitação em que o conteúdo da fita 1 é $0\square\square\ldots$.

   (b) Especifique uma máquina de Turing com 2 ou mais fitas que, partindo de uma configuração inicial relativa a uma palavra $\alpha \in L_{\text{grafos}}$ (ver exercício 3), chega a uma configuração de aceitação em que no início da fita 1 está o conjunto dos nós de α que são nós isolados, e os outros registos estão vazios.

2 Conjuntos numeráveis

Recorde que dados dois conjuntos A e B: (i) A e B *têm o mesmo cardinal*, ou são *equipotentes*, o que se pode representar por $A \approx B$, se existe uma aplicação $f : A \to B$ bijectiva, e (ii) A diz-se *numerável* se for equipotente a $\mathbb{N}_0$.

1. Mostre que se $A \approx B$ e $B \approx C$ então $A \approx C$.
2. Mostre que os conjuntos $\mathbb{N}$, $\mathbb{Z}$ e $\mathbb{Q}$ são numeráveis.
3. Mostre que o conjunto $\{0,1\}^*$ é numerável.
4. Mostre que o conjunto Σ^* é numerável, onde Σ é um qualquer alfabeto.
5. Mostre que $]0,1[\approx]a,b[$, para quaisquer reais a e b tais que $a < b$. (sugestão: mostre que $f :]0,1[\to]a,b[$ tal que $f(x) = (b-a)x + a$ é uma função bijectiva)
6. Mostre que $]0,1[\approx \mathbb{R}$.
7. $\mathbb{N}_0 \times \mathbb{N}_0 \approx \mathbb{N}$. (sugestão: mostre que é bijectiva a função $f : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}$ dada por $f(m,n) = 2^m(2n+1)$; use o facto de que qualquer inteiro positivo x tem uma decomposição única em fatores primos, e observe que se m é a potência de 2 na decomposição de x em fatores primos, então $\frac{x}{2^m}$ é um número ímpar)
8. $\mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0 \approx \mathbb{N}$.
9. Mostre que se $A \subseteq \mathbb{N}_0$ e A é um conjunto infinito então A é numerável.
10. Mostre que se $A \subseteq B$, A é infinito e B é numerável então A é numerável.

NOTA: Após a aula prática os alunos deverão tentar resolver todos os exercícios que não foram resolvidos na aula. Se tiverem dificuldades ou dúvidas deverão consultar os docentes da disciplina durante os respectivos horários de dúvidas.

Instituto Superior Técnico
Teoria da Computação - LEIC 2013/2014
Aula prática 5

1 Conjuntos não numeráveis

1. Use o método da diagonalização para demonstrar que $\mathbb{R}$ não é numerável.
2. Use o método da diagonalização para demonstrar que o conjunto das palavras infinitas sobre $\{0, 1\}$ não é numerável.
3. Use o método da diagonalização para demonstrar que o conjunto de todas as aplicações de $\{0, 1\}^*$ em $\{0, 1\}$ não é numerável.
4. Use o método da diagonalização para demonstrar que o conjunto de todas as aplicações de $\mathbb{N}_0$ em $\mathbb{N}_0$ não é numerável.
5. Seja Σ um alfabeto. Use o exercício 2 e facto de Σ^* ser numerável para demonstrar que o conjunto das linguagens sobre Σ não é numerável.

2 Operações sobre linguagens (conjuntos) decidíveis e semi-decidíveis

1. Sejam L_1, L_2 e L_3 linguagens sobre Σ. Mostre que se L_1, L_2 e L_3 são linguagens decidíveis então são também decidíveis as linguagens
   (a) $L_1 \cap L_2$
   (b) $L_1 \cup L_2$
   (c) $\Sigma^* \backslash L_1$
   (d) $L_1 \backslash L_2$
   (e) $L = \{w \in \Sigma^* : w \in L_1 \text{ ou } w \in L_2, \text{ mas } w \notin L_1 \cap L_2\}$
   (f) $L = \{w \in \Sigma^* : w \in L_1 \cap L_3 \text{ e } w \notin L_2\}$
2. Sejam L_1 e L_2 e L_3 linguagens sobre um mesmo alfabeto Σ. Assumindo que L_1 e L_2 são semi-decidíveis e L_3 é decidível, mostre que
   (a) $L_1 \cap L_2$ é semi-decidível.
   (b) $L_3 \cup L_2$ é semi-decidível.
   (c) $L_1 \cup L_2$ é semi-decidível.
   (d) $L_1 \backslash L_3$ é semi-decidível.
3. Seja L uma linguagem sobre um alfabeto Σ. Mostre que se L e $\Sigma^* \backslash L$ são semi-secidíveis então L é decidível.

3 Existência de linguagens não semi-decidíveis

1. Seja A um conjunto e $\wp(A)$ o conjunto dos subconjuntos de A.

   (a) Mostre A não é equipotente a $\wp(A)$.

   (b) Mostre que o cardinal de A é menor que o cardinal de $\wp(A)$. (este resultado é conhecido por *teorema de Cantor*)

2. Seja Σ um alfabeto. O teorema de Cantor pode ser usado para demonstrar que existem linguagens sobre Σ que não são semi-decidíveis, ou seja, que não são reconhecidas por nenhuma máquina de Turing (sendo assim uma demonstração alternativa à apresentada na aula teórica para este resultado):

   (a) mostre o conjunto das máquinas de Turing é numerável e portanto é equipotente a Σ^*

   (b) usando o teorema de Cantor conclua que o cardinal de Σ^* é menor que o cardinal das linguagens sobre Σ

   (c) usando as alíneas anteriores conclua que existem linguagens sobre Σ que não são reconhecidas por nenhuma máquina de Turing.

NOTA: Após a aula prática os alunos deverão tentar resolver todos os exercícios que não foram resolvidos na aula. Se tiverem dificuldades ou dúvidas deverão consultar os docentes da disciplina durante os respectivos horários de dúvidas.

Instituto Superior Técnico
Teoria da Computação - LEIC 2013/2014
Aula prática 6

No que se segue pode assumir como demonstrado que o conjunto das máquinas de Turing é numerável.

1. Seja $\mathcal{F}^t_{\{1\}^* \to \{1\}^*}$ o conjunto das funções totais de $\{1\}^*$ para $\{1\}^*$.
   (a) Demonstre que $\#\mathbb{N}_0 < \#\mathcal{F}^t_{\{1\}^* \to \{1\}^*}$
   (b) Demonstre que existem funções totais de $\{1\}^*$ para $\{1\}^*$ que não são calculadas por nenhuma máquina de Turing.
   (c) Use o postulado de Church-Turing para concluir que existem funções totais de $\{1\}^*$ para $\{1\}^*$ que não são computáveis.

2. Repita o exercício 1 para os seguintes conjuntos de funções
   (a) $\mathcal{F}^t_{\{0,1\}^* \to \{0,1\}}$ (funções totais de $\{0,1\}^*$ para $\{0,1\}$).
   (b) $\mathcal{F}^t_{\{0,1\}^* \to \{0,1\}^*}$ (funções totais de $\{0,1\}^*$ para $\{0,1\}^*$).
   (c) $\mathcal{F}^t_{\{a,b,c\}^* \to \{a,b,c\}^*}$ (funções totais de $\{a,b,c\}^*$ para $\{a,b,c\}^*$).
   (d) $\mathcal{F}_{\{1\}^* \to \{1\}^*}$ (funções de $\{1\}^*$ para $\{1^*\}$).
   (e) $\mathcal{F}_{\{0,1\}^* \to \{0,1\}}$ (funções de $\{0,1\}^*$ para $\{0,1\}$).

NOTA: Após a aula prática os alunos deverão tentar resolver todos os exercícios que não foram resolvidos na aula. Se tiverem dificuldades ou dúvidas deverão consultar os docentes da disciplina durante os respectivos horários de dúvidas.

Instituto Superior Técnico
Teoria da Computação - LEIC 2013/2014
Aulas práticas 7 e 8

1. Considere a linguagem

   $$A_{TM} = \{<M, w>\colon\ M \text{ é uma máquina de Turing que aceita } w\}.$$

   (a) Mostre que A_{TM} é semidecidível.

   (b) Mostre que o *problema da aceitação* relativo a máquinas de Turing não é decidível, ou seja, mostre que a linguagem A_{TM} não é decidível.

   (c) Mostre que a linguagem complementar de A_{TM} não é reconhecida por nenhuma máquina de Turing.

2. Mostre que o *problema da paragem* relativo a máquinas de Turing não é decidível, ou seja, mostre não é decidível a linguagem

   $$HALT_{TM} = \{<M, w>\colon\ M \text{ é uma máquina de Turing que aceita ou rejeita } w\}.$$

3. Mostre que o *problema da linguagem vazia* relativo a máquinas de Turing não é decidível, ou seja, mostre não é decidível a linguagem

   $$E_{TM} = \{<M>\colon\ M \text{ é uma máquina de Turing e } L_M = \{\}\}.$$

4. Mostre que o *problema da equivalência* relativo a máquinas de Turing não é decidível, ou seja, mostre não é decidível a linguagem

   $$EQ_{TM} = \{<M_1, M_2>\colon\ M_1,\ M_2 \text{ são máquinas de Turing e } L_{M_1} = L_{M_2}\}.$$

5. Seja Σ um alfabeto e p uma palavra sobre Σ. Mostre que não é decidível a linguagem

   $$DOM^p_{TM} = \{<M>\colon\ M \text{ é uma máquina de Turing que aceita } p\}.$$

   A não decidibilidade desta linguagem é também conhecida como a não decidibilidade do *"input problem"*.

   Sugestão: efectue uma prova por absurdo usando o facto de a linguagem A_{TM} não ser decidível.

6. Seja Σ um alfabeto e p uma palavra sobre Σ. Mostre que não é decidível a linguagem

   $$CDOM^p_{TM} = \{<M>\colon\ M \text{ é uma máquina de Turing que escreve } p\}.$$

   A expressão "M escreve p" significa aqui que existe pelo menos uma configuração inicial a partir da qual M chega a uma configuração de aceitação na qual o conteúdo da fita é p. A não decidibilidade desta linguagem é também conhecida como a não decidibilidade do *"printing problem"*.

   Sugestão: efectue uma prova por absurdo usando o facto de a linguagem A_{TM} não ser decidível.

7. Mostre que não é decidível a linguagem

$$\begin{aligned} FIN_{TM} = \{<M>:\ & M \text{ é uma máquina de Turing} \\ & \text{e } L_M \text{ é uma linguagem finita}\} \end{aligned}$$

Sugestão: efectue uma prova por absurdo usando o facto de a linguagem A_{TM} não ser decidível.

8. Mostre que não é decidível a linguagem

$$\begin{aligned} PAR_{TM} = \{<M>:\ & M \text{ é uma máquina de Turing} \\ & \text{e as palavras aceites por } M \text{ têm comprimento par } \} \end{aligned}$$

Sugestão: efectue uma prova por absurdo usando o facto de a linguagem A_{TM} não ser decidível.

9. Mostre que não é decidível a linguagem

$$\begin{aligned} IMP_{TM} = \{<M>:\ & M \text{ é uma máquina de Turing} \\ & \text{e as palavras aceites por } M \text{ têm comprimento ímpar}\} \end{aligned}$$

Sugestão: efectue uma prova por absurdo usando o facto de a linguagem A_{TM} não ser decidível.

10. Considere o universo das máquinas de Turing com alfabeto $\{0,1\}^*$ e as linguagens sobre $\{0,1\}$.

   (a) Mostre que não é decidível a linguagem

   $$\begin{gathered} L = \{<M>:\ M \text{ é uma máquina de Turing,} \\ \text{e } L_M \text{ é o conjunto das palavras que começam e terminam em } 0\} \end{gathered}$$

   Sugestão: efectue uma prova por absurdo usando o facto de a linguagem A_{TM} não ser decidível.

   (b) Mostre que não é decidível a linguagem

   $$\begin{gathered} L = \{<M>:\ M \text{ é uma máquina de Turing, e } L_M \\ \text{é o conjunto das palavras que têm o mesmo número de 0's e de 1's}\} \end{gathered}$$

   Sugestão: efectue uma prova por absurdo usando o facto de a linguagem A_{TM} não ser decidível.

   (c) Mostre que não é decidível a linguagem

   $$\begin{aligned} L = \{<M>:\ & M \text{ é uma máquina de Turing,} \\ & \text{e } L_M \text{ é o conjunto de todos palíndromos}\} \end{aligned}$$

   Sugestão: efectue uma prova por absurdo usando o facto de a linguagem A_{TM} não ser decidível.

11. Uma *gramática* é um tuplo $G = (V, \Sigma, P, S)$ em que

   - V é um conjunto finito (conjunto dos símbolos auxiliares)

- Σ é um alfabeto
- P é um conjunto finito de regras de substituição (ou produções); uma regra de substituição é uma expressão $\alpha \to \beta$ em que $\alpha, \beta \in (\Sigma \cup V)^*$ e α tem pelo menos um símbolo em V
- $S \in V$ (símbolo inicial).

Uma palavra $w \in \Sigma^*$ diz-se *gerada* por uma gramática $G = (V, \Sigma, P, S)$ se existe uma sequência finita $w_1, \ldots, w_n$ de palavras em $(\Sigma \cup V)^*$ tal que $w_1 = S$, $w_n = w$ e, para cada $1 < i \leq n$, a palavra w_i obtém-se a partir w_{i-1} por aplicação de uma regra de substituição (w_i obtém-se a partir w_{i-1} por aplicação de uma regra de substituição $\alpha \to \beta$ se α é uma subpalavra de w_{i-1}, e w_i se obtém de w_{i-1} substituindo uma ocorrência de α em w_{i-1} por β). A *linguagem gerada* por G é o conjunto das palavras $w \in \Sigma^*$ que são geradas por G.

(a) Uma linguagem diz-se *regular* se é gerada por uma gramática na qual todas as regras de substituição são do tipo $X \to \beta$, onde $X \in V$ e $\beta \in \Sigma \cup \{\varepsilon\}$ ou $\beta = sY$ com $s \in \Sigma$ e $Y \in V$. Mostre que não é decidível a linguagem

$$REG_{TM} = \{<M>\colon \ M \text{ é uma máquina de Turing} \\ \text{e } L_M \text{ é uma linguagem regular}\}$$

Sugestão: efectue uma prova por absurdo usando o facto de a linguagem A_{TM} não ser decidível, e pode assumir como provado que a linguagem $\{0^n 1^n\}$ não é regular.

(b) Uma linguagem diz-se *independente do contexto* se é gerada por uma gramática na qual todas as regras de substituição são do tipo $X \to \beta$ com $X \in V$. Mostre que não é decidível a linguagem

$$IC_{TM} = \{<M>\colon \ M \text{ é uma máquina de Turing} \\ \text{e } L_M \text{ é uma linguagem e indepedente do contexto}\}$$

Sugestão: efectue uma prova por absurdo usando o facto de a linguagem A_{TM} não ser decidível, e pode assmir como provado que a linguagem $\{0^n 1^n 2^n\}$ não é independente do contexto.

NOTA: Após a aula prática os alunos deverão tentar resolver todos os exercícios que não foram resolvidos na aula. Se tiverem dificuldades ou dúvidas deverão consultar os docentes da disciplina durante os respectivos horários de dúvidas.

Instituto Superior Técnico
Teoria da Computação - LEIC 2013/2014
Aula prática 9

1 Redução

Recorde que dadas duas linguagens $L_1, L_2 \subseteq \Sigma^*$ se diz que L_1 é redutível a L_2, o que se denota por $L_1 \leq L_2$, se existe uma função $f : \Sigma^* \to \Sigma^*$ total, computável, tal que $w \in L_1$ se e só se $f(w) \in L_2$.

1. Sejam $L_1, L_2 \subseteq \Sigma^*$. Mostre que $L_1 \leq L_2$ se e só se $(\Sigma^* \backslash L_1) \leq (\Sigma^* \backslash L_2)$.

2. Sejam $L_1, L_2, L_3 \subseteq \Sigma^*$. Mostre que se $L_1 \leq L_2$ e $L_2 \leq L_3$ então $L_1 \leq L_3$.

3. Sejam $L_1, L_2 \subseteq \Sigma^*$ tais que $L_1 \leq L_2$. Mostre que
   (a) se L_2 é decidível então L_1 é decidível.
   (b) se L_1 não é decidível então L_2 não é decidível.

4. Sejam $L_1, L_2 \subseteq \Sigma^*$ tais que $L_1 \leq L_2$. Mostre que
   (a) se L_2 é semidecidível então L_1 é semidecidível.
   (b) se L_1 não é semidecidível então L_2 não é semidecidível.

5. Mostre que $L \subseteq \{0,1\}^*$ é decidível se e só se $L \leq \{0^n 1^n : \text{ com } n \in \mathbb{N}_0\}$.

6. Recorde as linguagens

   $A_{TM} = \{<M, w>: \ M \text{ é uma máquina de Turing que aceita } w\}$
   e
   $HALT_{TM} = \{<M, w>: \ M \text{ é uma máquina de Turing que aceita ou rejeita } w\}$

   (a) Mostre que $A_{TM} \leq HALT_{TM}$.
   Sugestão: Considere $f : \{0,1\}^* \to \{0,1\}^*$ dada por

   $$f(\alpha) = \begin{cases} \varepsilon & \text{se } \alpha \text{ não é uma codificação } < M, w > \\ < M', w > & \text{se } \alpha =< M, w > \end{cases}$$

   onde M' é uma máquina que, dado o *input* x, simula o comportamento de M com *input* x, e chega a uma configuração de aceitação se M chega a uma configuração de aceitação, e tem uma evolução infinita caso contrário.

   (b) Use a alínea anterior para demonstrar que $HALT_{TM}$ não é decidível.

7. Mostre que $L \subseteq \{0,1\}^*$ é semidecidível se e só se $L \leq A_{TM}$.

8. Mostre que $A_{TM} \not\leq \overline{A_{TM}}$, onde $\overline{A_{TM}}$ é a linguagem complementar de A_{TM}.

9. Mostre que se $L \subseteq \Sigma^*$ é semidecidível e $L \leq (\Sigma^* \backslash L)$ então L é decidível.

10. Recorde a linguagem

$$EQ_{TM} = \{< M_1, M_2 >: \ L_{M_1} = L_{M_2} \ \}$$

e seja $\overline{EQ_{TM}}$ a linguagem complementar de EQ_{TM}.

   (a) Mostre que $A_{TM} \leq \overline{EQ_{TM}}$.
   Sugestão: Considere uma máquina V que rejeita todas as palavras e $f : \{0,1\}^* \to \{0,1\}^*$ dada por

$$f(\alpha) = \begin{cases} < V, V > & \text{se } \alpha \text{ não é uma codificação } < M, w > \\ < V, S_{M,w} > & \text{se } \alpha =< M, w > \end{cases}$$

   onde $S_{M,w}$ é uma máquina que, dado o *input* x, simula o comportamento de M com *input* w e tem comportamento igual ao de M.

   (b) Use a alínea anterior para demonstrar que EQ_{TM} não é reconhecida por nenhuma máquina de Turing.

2 Teorema de Rice

1. Use o teorema de Rice para demonstrar que as seguintes linguagens não são decidíveis

   (a) $L = \{< M >: \ M$ é uma máquina de Turing e $\varepsilon \notin L_M\}$
   (b) $L = \{< M >: \ M$ é uma máquina de Turing e L_M tem pelo menos duas palavras$\}$
   (c) $L = \{< M >: \ M$ é uma máquina de Turing e L_M é decidível$\}$
   (d) as linguagens referidas nos exercícios 3, 5, 7, 8, 9 e 10 da lista de exercícios para as aulas práticas 7 e 8.

2. Pode usar o teorema de Rice para demonstrar que as seguintes linguagens não são decidíveis? Em caso afirmativo, use-o para demonstrar que a linguagem em causa não é decidível.

   (a) $L = \{< M >: \ M$ é uma máquina de Turing que escreve 111 $\}$
   (a expressão "M escreve 111" significa aqui que existe pelo menos uma configuração inicial a partir da qual M chega a uma configuração de aceitação na qual o conteúdo da fita é 111).
   (b) $L = \{< M >: \ M$ é uma máquina de Turing e M aceita 111$\}$
   (c) $L = \{< M >: \ M$ é uma máquina de Turing e M tem mais que 10 estados$\}$

(d) $L = \{<M>\colon\ M$ é uma máquina de Turing e L_M é semidecidível$\}$

(e) $L = \{<M>\colon\ M$ é uma máquina de Turing e M nunca atinge uma configuração de rejeição$\}$

(f) $L = \{<M>\colon\ M$ é um classificador$\}$

(g) $L = \{<M>\colon\ M$ é uma máquina de Turing e quer L_M quer a linguagem complementar de L_M são linguagens infinitas$\}$

NOTA: Após a aula prática os alunos deverão tentar resolver todos os exercícios que não foram resolvidos na aula. Se tiverem dificuldades ou dúvidas deverão consultar os docentes da disciplina durante os respectivos horários de dúvidas.

Instituto Superior Técnico
Teoria da Computação - LEIC 2013/2014
Aula prática 10

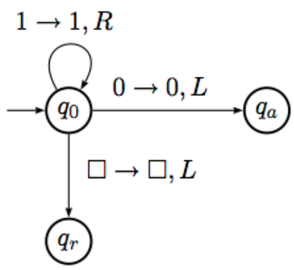
1 Comportamento assimptótico de funções

1. Mostre que
   (a) $n + 5 \in O(n)$ e $n + 5 \in O(n^2)$.
   (b) $n + 5 \in \Theta(n)$ mas $n + 5 \notin \Theta(n^2)$.
2. Seja $p : \mathbb{N}_0 \to \mathbb{R}_0^+$ tal que $p(n)$ é um polinómio de grau k. Mostre que $p(n) \in O(n^{k'})$ qualquer que seja $k' \geq k$.
3. Sejam $f, g : \mathbb{N}_0 \to \mathbb{R}_0^+$.
   (a) Mostre que se $\lim \dfrac{f(n)}{g(n)} = r \in \mathbb{R}_0^+$ então $f \in O(g)$.
   (b) Mostre que se $\lim \dfrac{f(n)}{g(n)} = r \in \mathbb{R}^+$ então $f \in \Theta(g)$.
4. Mostre que
   (a) $2^{n+3} \in O(2^n)$.
   (b) $2^{n+3} \in \Theta(2^n)$.
   (c) $n! \in O(n^n)$.
   (d) $\log(n^2 + 1) \in O(\log n)$.
   (e) $\log(n^2 + 1) \in \Theta(\log n)$.
   (f) $\log(n!) \in O(n \log(n))$.
5. Sejam $f, g : \mathbb{N}_0 \to \mathbb{R}^+$ e $h : \mathbb{N}_0 \to \mathbb{N}_0$. Mostre que se $f \in O(n^{k_1})$, $g \in O(n^{k_2})$ e $h \in O(n^{k_3})$ então
   (a) $f + g \in O(n^{\max\{k_1,k_2\}})$.
   (b) $f \times g \in O(n^{k_1+k_2})$.
   (c) $f \circ h \in O(n^{k_1 \times k_2})$.
6. Sejam $f, g, h, f_1, f_2, g_1, g_2 : \mathbb{N}_0 \to \mathbb{R}_0^+$, e $max(\{g_1, g_2\})$ a função que a cada $n \in \mathbb{N}_0$ faz corresponder $max(\{g_1(n), g_2(n)\})$. Mostre que
   (a) se $f \in O(g)$ e $g \in O(h)$ então $f \in O(h)$.
   (b) se $f_1 \in O(g_1)$ e $f_2 \in O(g_2)$ então $f_1 + f_2 \in O(max(\{g_1, g_2\}))$.
   (c) se $f_1 \in O(g)$ e $f_2 \in O(g)$ então $f_1 + f_2 \in O(g)$.
   (d) se $f_1 \in O(g_1)$ e $f_2 \in O(g_2)$ então $f_1 \times f_2 \in O(g_1 \times g_2)$.

7. Sejam $f, g, h, : \mathbb{N}_0 \to \mathbb{R}_0^+$. Mostre que
   (a) $f \in \Theta(g)$ sse $g \in \Theta(f)$.
   (b) se $f \in \Theta(g)$ e $g \in \Theta(h)$ então $f \in \Theta(h)$.

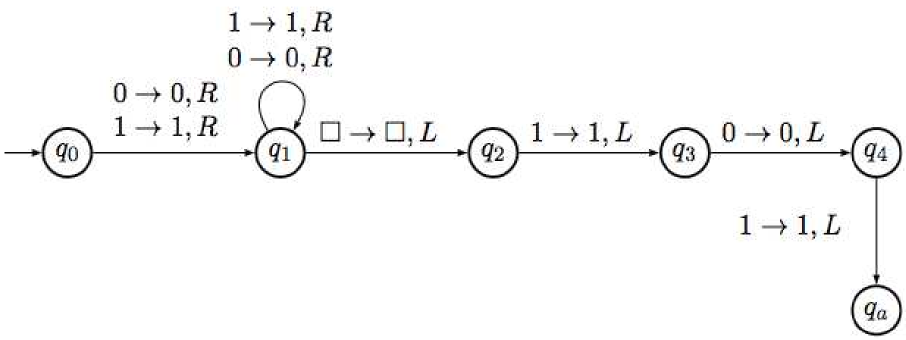
2 Tempo e espaço de máquina de Turing

1. Considere a seguinte máquina de Turing M com alfabeto $\{0, 1\}$ que decide a linguagem $L = \{w \in \{0, 1\}^* : w \text{ tem pelo menos um } 0\}$:



   (a) Calcule o tempo (ou complexidade temporal) t_M de M e caracterize o comportamento assimptótico de t_M indicando uma função g tal que $t_M \in O(g)$.
   (b) Calcule o espaço (ou complexidade espacial) s_M de M e caracterize o comportamento assimptótico de s_M indicando uma função g tal que $s_M \in O(g)$.
   (c) Tendo em conta as alíneas anteriores indique classes de linguagens $TIME(t(n))$ e $SPACE(s(n))$ apropriadas tais que $L \in TIME(t(n))$ e $L \in SPACE(s(n))$. Verifica-se $L \in P$? E $L \in PSPACE$?

2. Considere a seguinte máquina de Turing M com alfabeto $\{0, 1\}$ que decide a linguagem $L = \{w \in \{0, 1\}^* : w \text{ termina em } 101\}$:

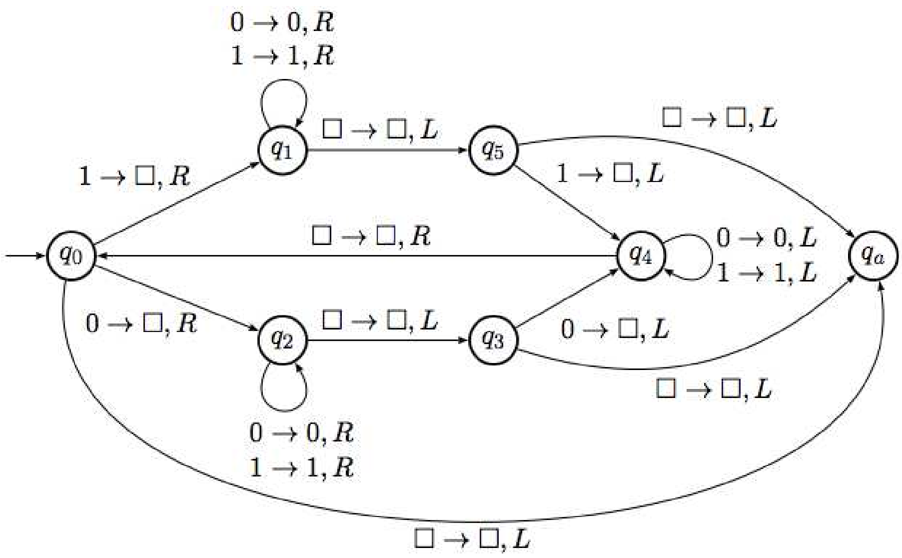


   (a) Calcule o tempo (ou complexidade temporal) t_M de M e caracterize o comportamento assimptótico de t_M indicando uma função g tal que $t_M \in O(g)$.

(b) Calcule o espaço (ou complexidade espacial) s_M de M e caracterize o comportamento assimptótico de s_M indicando uma função g tal que $s_M \in O(g)$.

(c) Tendo em conta as alíneas anteriores indique classes de linguagens $TIME(t(n))$ e $SPACE(s(n))$ apropriadas tais que $L \in TIME(t(n))$ e $L \in SPACE(s(n))$. Verifica-se $L \in P$? E $L \in PSPACE$?

3. Considere a seguinte máquina de Turing M com alfabeto $\{0, 1\}$ que decide a linguagem L dos palíndromos em $\{0, 1\}^*$.

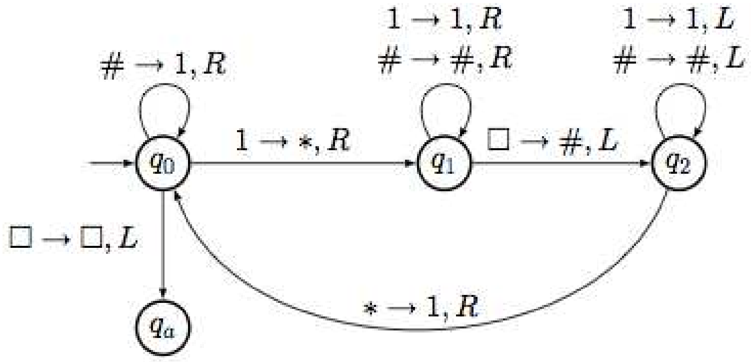


(a) Caracterize o comportamento assimptótico do tempo (ou complexidade temporal) t_M de M indicando uma função g tal que $t_M \in O(g)$.

(b) Caracterize o comportamento assimptótico do espaço (ou complexidade espacila) s_M de M indicando uma função g tal que $s_M \in O(g)$.

(c) Tendo em conta as alíneas anteriores indique classes de linguagens $TIME(t(n))$ e $SPACE(s(n))$ apropriadas tais que $L \in TIME(t(n))$ e $L \in SPACE(s(n))$. Verifica-se $L \in P$? E $L \in PSPACE$?

4. Considere a seguinte máquina de Turing M que calcula a função *dobro*, isto é, a função que a cada número natural em notação unária (n é representado por 1^n) faz corresponder o seu dobro:



Mostre que a função *dobro* pertence a PF e a $PSPACEF$.

NOTA: Após a aula prática os alunos deverão tentar resolver todos os exercícios que não foram resolvidos na aula. Se tiverem dificuldades ou dúvidas deverão consultar os docentes da disciplina durante os respectivos horários de dúvidas.

Instituto Superior Técnico
Teoria da Computação - LEIC 2013/2014
Aulas práticas 11 e 12

1 Classes de complexidade

1. Mostre que
   (a) $P \subseteq PSPACE$.
   (b) $PSPACE \subseteq EXPTIME$.
   (c) $P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIME$
   (recorde que o teorema de Savitch estabelece que se $f : \mathbb{N}_0 \to \mathbb{R}^+$ e $f(n) \geq n$ então $NSPACE(f(n)) \subseteq SPACE(f(n)^2)$).

2. Seja $f : \mathbb{N}_0 \to [1, +\infty[$. Mostre que $TIME(f) \subseteq SPACE(f)$.

3. Mostre que a classe P é fechada para as seguintes operações sobre linguagens:

   (a) intersecção (b) reunião (c) complementação (d) diferença

   Uma classe $\mathcal{C}$ de linguagens diz-se *fechada para a intersecção* se dadas duas linguagens $L_1, L_2 \in \mathcal{C}$ se tem que $L_1 \cap L_2 \in \mathcal{C}$. A noção de classe fechada para a reunião, diferença e para a complementação é naturalmente semelhante.

4. Mostre que a classe $PSPACE$ é fechada para as seguintes operações sobre linguagens:

   (a) intersecção (b) reunião (c) complementação (d) diferença

5. Seja $f : \Sigma^* \to \Sigma^*$ uma função em PF total. Seja x um elemento do contradomíno de f. Demonstre que a linguagem $A = \{y \in \Sigma^* : f(y) = x\}$ pertence à classe P.

6. Mostre que a linguagem $L = \{<N, w>: N$ é uma máquina de Turing que aceita w em, no máximo, $2^{|w|}$ transições$\}$ não está em P.

   Sugestão: Realize uma demonstração por contradição. Comece por concluir que se $L \in P$ então $L' \in P$ onde $L' = \{<N>: N$ é uma máquina de Turing que não aceita $<N>$ em, no máximo, $2^{|<N>|}$ transições$\}$. Considere depois um classificador C' para L' com tempo polinomial, e conclua que se obtém uma contradição quando se analisa a evolução de C' com *input* $<C'>$.

2 Redução

1. Sejam L_1 e L_2 linguagens sobre um alfabeto Σ, e seja $\overline{L_1} = \Sigma^* \backslash L_1$ e $\overline{L_2} = \Sigma^* \backslash L_2$. Mostre que se $L_1 \leq_p L_2$ então $\overline{L_1} \leq_p \overline{L_2}$.

2. Sejam L_1, L_2 e L_3 linguagens sobre um alfabeto Σ. Mostre que se $L_1 \leq_p L_2$ e $L_2 \leq_p L_3$ então $L_1 \leq_p L_3$.

3. Sejam A e B linguagens sobre um alfabeto Σ.

   (a) Mostre que se $A \leq_p B$ e $B \in P$ então $A \in P$.

   (b) Mostre que se $A \leq_p B$ e $A \notin P$ então $B \notin P$

   (c) Repita as alíneas 3a e 3b mas agora para a classe $PSPACE$. .

   (d) Repita as alíneas 3a e 3b mas agora para a classe NP.

4. Sendo $\mathcal{C}$ uma classe de linguagens e L uma linguagem. Mostre que

   (a) se L é uma linguagem $\mathcal{C}$-*difícil* e $L \in P$ então todas as linguagens da classe $\mathcal{C}$ pertencem a P.

   (b) se L é uma linguagem $\mathcal{C}$-*difícil* e $L \in PSPACE$ então todas as linguagens da classe $\mathcal{C}$ pertencem a $PSPACE$.

5. Sendo $\mathcal{C}$ uma classe de linguagens e L uma linguagem. Mostre que

   (a) se L é uma linguagem $\mathcal{C}$-*difícil* e $L \leq_p A$ para alguma linguagem A então A é também $\mathcal{C}$-difícil.

   (b) se L é uma linguagem $\mathcal{C}$-*completa*, $A \in \mathcal{C}$ e $L \leq_p A$ então A é também $\mathcal{C}$-completa.

6. Seja $\mathcal{C}$ uma classe de linguagens fechada para a complementação, isto é, se $A \in \mathcal{C}$ então $\Sigma^* \backslash A \in \mathcal{C}$. Mostre que se L é $\mathcal{C}$-difícil então $\Sigma^* \backslash L$ é $\mathcal{C}$-difícil.

NOTA: Após a aula prática os alunos deverão tentar resolver todos os exercícios que não foram resolvidos na aula. Se tiverem dificuldades ou dúvidas deverão consultar os docentes da disciplina durante os respectivos horários de dúvidas.