

Exame de Dados e Computação 2020

Universidade do Minho, 11 de Julho de 2020

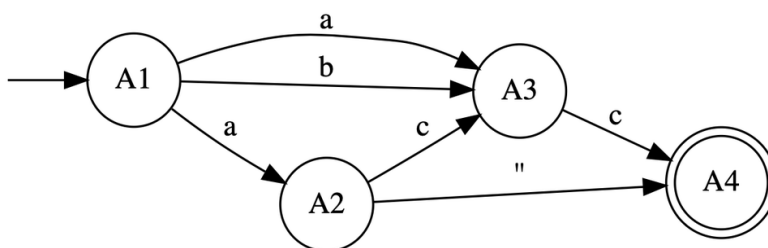
I. Autómatos Finitos

1. Considere o alfabeto $\Sigma = \{a, b\}$. Defina um autómato determinístico (DFA) que reconheça a linguagem das palavras com exatamente duas ocorrências de "aa" e zero ocorrências de "bbb" (i.e. as palavras que contêm como subpalavra "aa" duas vezes, e não contêm "bbb").
2. Considere a seguinte afirmação:

Para qualquer $n \geq 0$, é possível construir um autómato finito que reconhece a linguagem $\{a^n b^n\}$.

Diga se esta afirmação é verdadeira (e nesse caso mostre como se constrói esse autómato para um valor pequeno mas representativo de n à sua escolha) ou falsa (e nesse caso demonstre porquê).

3. Considere o seguinte autómato finito, não-determinístico (NFA). Note que " denota uma transição- ϵ .



- a. Utilizando o processo sistemático descrito nas aulas, obtenha uma expressão regular correspondente à linguagem reconhecida por este NFA.
- b. Defina um autómato determinístico (DFA) que reconheça esta mesma linguagem.

4. Considere a operação de *união de duas linguagens*. Demonstre que esta operação é fechada nas linguagens regulares, ou seja, a união de duas linguagens regulares é ainda uma linguagem regular.

II. Autômatos de Pilha

1. Apresente um autômato de pilha (PDA) para a linguagem $\{a^n b^n \mid n \geq 0\}$.
2. Simule a execução do autômato anterior correspondente à aceitação da palavra *aaabbb*.

III. Máquinas de Turing

Considerando a representação unária dos números naturais, conceba uma máquina de Turing para implementar a operação “-1”, que calcula o antecessor de um número. Considere que a fita contém inicialmente o número cujo antecessor se pretende calcular, representado por uma sequência de “1”. A cabeça da máquina está inicialmente posicionada sobre o primeiro símbolo do primeiro número, e deverá estar assim no final.

Por exemplo se a fita contiver inicialmente (B é o símbolo “blank”)

... | 1 | 1 | 1 | 1 | B | ...

no final deverá conter

... | 1 | 1 | 1 | B | ...

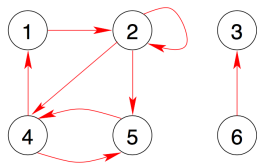
Note que, caso o input seja o número 0, a execução da máquina não deverá parar, executando em ciclo infinito, sinalizando assim que o antecessor de 0 não é um número natural. Apresente a sua MT de forma gráfica.

IV. Programação com Grafos

Tomando por base a representação de grafos orientados estudada, e a função básica de travessia em profundidade:

```
g = {}
```

```
g[1] = [2]
```



Grafo orientado g

```
3 g[2] = [2,4,5]
4 g[3] = []
5 g[4] = [1,5]
6 g[5] = [4]
7 g[6] = [3]
```

1. Escreva uma função `count_reachable` que, quando chamada com um grafo e um vértice, devolve o número de vértices alcançáveis a partir daquele vértice (não contando o próprio). Por exemplo, deve calcular 3 para o vértice 1, 1 para o vértice 6, e 0 para o vértice 3.
2. Escreva uma função `top_reachable` que, quando chamada com um grafo, indica qual o (ou um dos, se não for único) vértice do grafo com maior número de vértices alcançáveis a partir dele. Por exemplo para o grafo acima, devolveria um dos vértices 1, 2, 4, ou 5.

FIM