

Teste de Dados e Computação

Engenharia Física / Universidade do Minho

29 de Maio 2021

1. Considere a expressão regular $(01)^*0 + 1$
 - a. Seguindo o processo sistemático estudado nas aulas, apresente um autómato finito não-determinístico (NFA) que reconheça a linguagem definida por esta ER
 - b. Converta-o num autómato determinístico (DFA) minimizado, i.e. sem estados redundantes.

2. Considere a linguagem $L = \{0^i1^j \mid i \geq j\}$.
 - a. Mostre que se trata de uma linguagem não-regular
 - b. Apresente uma gramática para L
 - c. Desenhe uma árvore de sintaxe correspondente à derivação da palavra 00011 pela gramática que definiu em b
 - d. Defina um autómato de pilha que reconheça a linguagem L
 - e. Simule a execução do autómato anterior correspondente à aceitação da palavra 00011.

3. Defina uma Máquina de Turing que reconheça a linguagem L do exercício 2.

4. Recorde o projeto prático. Considere o seguinte dicionário D ({ código : set{contactos} }) para representar os indivíduos registado no SNS e os seus contactos (códigos). Escreva uma função em Python3 que dado um dicionário como D (onde a um código de uma pessoa corresponde um conjunto de códigos dos seus contactos) e o código de um cidadão infetado calcula o conjunto de indivíduos potencialmente infetados. Notar que há a possibilidade de ciclos e portanto a computação tem parar evitando estes ciclos. Como exemplo de uso considere a seguinte invocação, que deve retornar o conjunto {12,13,14,15}:

```
infetados({11:{12,13},12:{11,14},13:{11,15},14:{12},15:{13},16:{27}},11)
```

5. Escreva uma função em Python3 que retorna o caracter mais frequente numa *String* dada. Por exemplo, a invocação `mais_freq("um exemplo a ver")` retorna `'e'`.
6. Uma empresa de circuitos eletrónicos regista os seus produtos numa aplicação. Cada placa (circuito) é constituída por um conjunto de componentes. A aplicação usa uma estrutura de dados dicionário para registar esta informação. Assim, temos uma correspondência entre um código da placa e o conjunto de códigos dos componentes que a compõe.
- a. Implemente uma função em Python3 que dado um dicionário deste tipo identifica pelo código o componente mais usado na empresa. Exemplo: para a invocação `mais_usado({11:{1,2,3},12:{1,2},13:{1}})` obtinha-se o valor `1`.
- b. Apresente uma função em Python3 que dado este dicionário retorna um tuplo que representa o par de códigos de placas com mais componentes em comum. Exemplo: `comp_comuns({11:{1,2,3},12:{1,4},13:{1},14:{2,3}})` devolvia o tuplo `(11,14)`.