

# Compiladores

## Exercícios sobre análise lexical

Licenciatura em Engenharia Informática  
Universidade de Coimbra

Ano Letivo 2024/25

1. Considere as seguintes categorias lexicais:

$$\begin{aligned} \text{ID} &= [\text{a-z}] [\text{a-z0-9}]^* \\ \text{NUM} &= [0-9]^+ \\ \text{INT} &= \text{int} \\ \text{IF} &= \text{if} \\ \text{ELSE} &= \text{else} \\ \text{END} &= \text{end} \\ \text{BLANK} &= (< \text{space} > \mid < \text{newline} >)^+ \end{aligned}$$

- (a) Construa um autómato finito não-determinístico para cada uma dessas expressões regulares.
- (b) Partindo dos autómatos obtidos na alínea anterior, construa um autómato finito determinístico que permita reconhecer palavras de cada uma das categorias dadas, diferenciando entre elas. Justifique as opções que fizer.
- (c) Formalize a classe de comentários do tipo `/* ... */` (sequências iniciadas por `/*` e terminadas pela primeira ocorrência de `*/`) em termos de um autómato finito determinístico, e obtenha a correspondente expressão regular.
2. Numa hipotética linguagem assembly, as categorias lexicais definidas são as seguintes:

$$\begin{aligned} \text{ID} &= [\text{a-z}][0-9\text{a-z}]^* \\ \text{LABEL} &= [0-9\text{a-z}]^+ \text{ ":"} \\ \text{NUM} &= [0-9]^+ \\ \text{BYTE} &= 0\text{x}[0-9\text{a-f}][0-9\text{a-f}] \\ \text{WORD} &= 0\text{x}[0-9\text{a-f}][0-9\text{a-f}][0-9\text{a-f}][0-9\text{a-f}] \\ \text{COMMA} &= \text{" , " } \\ \text{SPC} &= (<\text{space}> \mid <\text{tab}> \mid <\text{newline}>)^+ \end{aligned}$$

- (a) Construa um autômato finito não-determinístico para cada uma dessas expressões regulares.
- (b) Partindo dos autômatos obtidos na alínea anterior, construa um autômato finito determinístico que permita reconhecer palavras de cada uma das categorias dadas, diferenciando entre elas. Justifique as opções que fizer.
- (c) Descreva a operação de um analisador lexical baseado no autômato obtido na alínea anterior sobre as seguintes linhas de entrada, indicando os tokens identificados e as sequências de símbolos não reconhecidas (UNKNOWN):

```
loop: cmp 051:, 0xcd10xba
      :ab 0x123 ret
```

3. À semelhança da tipografia convencional, alguns processadores de texto representam certas sequências de caracteres, nomeadamente ff, fi, fl, ffi e ffl, por símbolos especiais, designados “ligaduras tipográficas” (ff, fi, fl, ffi e ffl, respetivamente). Por outro lado, as reticências (sequências de 3 pontos) também são impressas de modo especial, porque o espaçamento entre os pontos é diferente do normal.

Para além disso, o espaço entre as palavras é controlado pelo processador de texto. Sequências de espaços e/ou mudanças de linha são tratadas como um único espaço se o número de mudanças de linha for no máximo um, e como uma mudança de parágrafo se o número de mudanças de linha for dois ou mais. Finalmente, o espaço deixado a seguir a um ponto final, se existir, é ligeiramente superior ao espaço normal, *exceto* quando o ponto se segue a uma maiúscula.

As transformações descritas acima podem ser implementadas através da análise lexical do texto a processar. Tendo em vista a implementação de um tal analisador:

- (a) Defina as categorias lexicais que considerar necessárias, apresentando as respetivas expressões regulares.
- (b) Construa um autômato finito determinístico que permita reconhecer os tokens de cada uma das categorias que definiu, diferenciando entre elas. Justifique as opções que fizer.
- (c) Assumindo um alfabeto de entrada com 256 caracteres, quantos valores inteiros seriam necessários para representar a tabela de transição de estados e o vetor de ações deste analisador? Justifique.