



Análise Lexical

Compiladores, Aula Nº 8
João M. P. Cardoso



Da Expressão Regular ao Analizador Lexical

- ✍ Tradução da Expressão Regular para um NFA
- ✍ Do NFA para o DFA
- ✍ Implementação em software do DFA



Interpretador do DFA

Algoritmo do Interpretador

```

State = DFA initial state;
inputChar = getchar();
While(inputChar) {
    State = trans(State, inputChar);
    inputChar = getchar();
}
If(State é um estado de aceitação)
    realizar acção relativa ao estado State (reconheceu String)
Else
    realizar outra acção
    
```

3

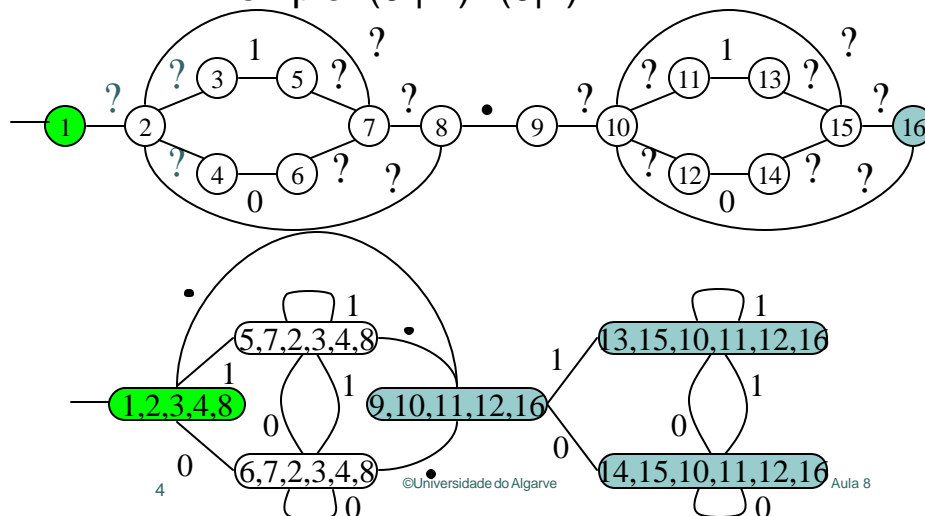
©Universidade do Algarve

Aula 8



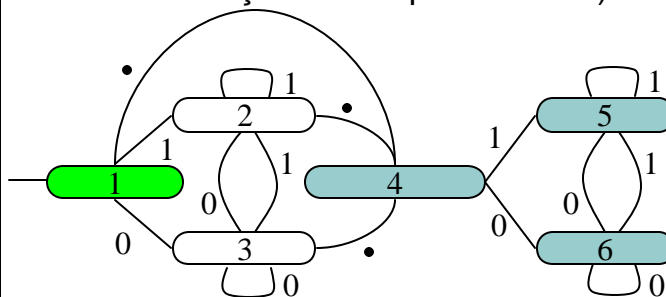
NFA para DFA

Exemplo: $(0 | 1)^*. (0|1)^*$



DFA

✍ Tabela de transição de estados
(adicionamos o estado 0 –
estado morto - para as
transições não apresentadas)



Esta do Actu al	Próximo Estado		
	"0"	"1"	"."
0	0	0	0
1	3	2	4
2	3	2	4
3	3	2	4
4	6	5	0
5	6	5	0
6	6	5	0

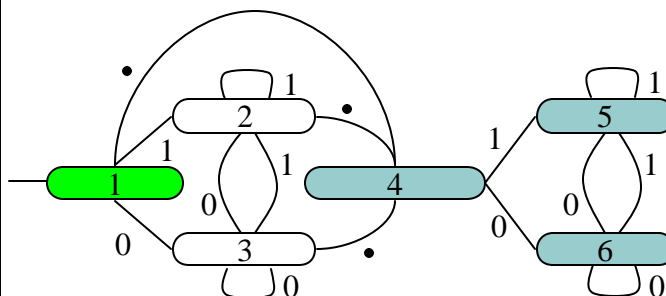
5

©Universidade do Algarve

Aula 8

Implementação do DFA

✍ Implementar a tabela de
transições de estados com
um array bidimensional



Esta do Actu al	Próximo Estado		
	"0"	"1"	"."
0	0	0	0
1	3	2	4
2	3	2	4
3	3	2	4
4	6	5	0
5	6	5	0
6	6	5	0

6

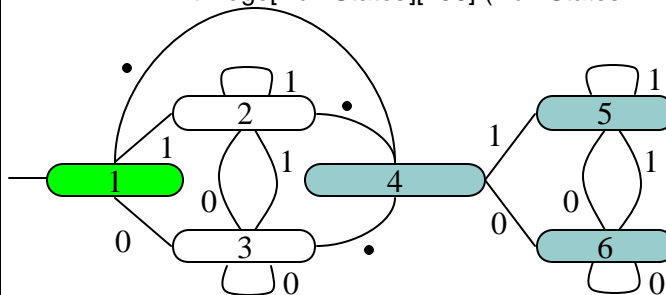
©Universidade do Algarve

Aula 8



Implementação do DFA

- ✍ O array fornecerá uma indexação por número de estado e símbolo
- ✍ Supondo a possibilidade de 256 símbolos:
 - ✍ `int Edge[NumStates][256] (NumStates = 7)`



Esta do Actu al	Próximo Estado		
	"0"	"1"	"."
0	0	0	0
1	3	2	4
2	3	2	4
3	3	2	4
4	6	5	0
5	6	5	0
6	6	5	0

7

©Universidade do Algarve

Aula 8



Implementação do DFA

```
int Edge[NumStates][256] =
{
    /* ... 0 1 2 ... 9 ... "." ... */
    /* estado 0 */ { ..., 0, 0, 0, ..., 0, ..., 0, ...},
    /* estado 1 */ { ..., 3, 2, 0, ..., 0, ..., 4, ...},
    /* estado 2 */ { ..., 3, 2, 0, ..., 0, ..., 4, ...},
    /* estado 3 */ { ..., 3, 2, 0, ..., 0, ..., 4, ...},
    /* estado 4 */ { ..., 6, 5, 0, ..., 0, ..., 0, ...},
    /* estado 5 */ { ..., 6, 5, 0, ..., 0, ..., 0, ...},
    /* estado 6 */ { ..., 6, 5, 0, ..., 0, ..., 0, ...}
}
```

Exemplo:
Edge[3][(int) '.'] ? 4

Esta do Actu al	Próximo Estado		
	"0"	"1"	"."
0	0	0	0
1	3	2	4
2	3	2	4
3	3	2	4
4	6	5	0
5	6	5	0
6	6	5	0

8

©Universidade do Algarve

Aula 8



Implementação do DFA

- Um array unidimensional traduz o número de estado no número da acção a realizar:

```
// estado      0 1 2 3 4 5 6
int final[NumStates] = {0, 0, 0, 0, 1, 1, 1}
```

Exemplo:

final[3] ? 3 (logo deve ser realizada a acção 3)

Esta do Actu al	Próximo Estado		
	"0"	"1"	"."
0	0	0	0
1	3	2	4
2	3	2	4
3	3	2	4
4	6	5	0
5	6	5	0
6	6	5	0



Implementação do DFA

Programação do Interpretador

```
State = 1; // estado inicial do DFA
inputChar = input.read();
While(inputChar) {
    State = edge[State][inputChar];
    inputChar = input.read();
}
If(final[State] == 1)
    realizar acção (reconheceu String)
Else
    realizar outra acção
```

Optimizações na implementação do DFA

- ✍ Representação da tabela de transições por um array bidimensional pode não ser eficiente em termos de espaço de memória

- ✍ Mapeamento indirecto:

```
int Edge[NumStates][4] = {
/* estado 0 */ {0, 0, 0, 0},
/* estado 1 */ {3, 2, 4, 0},
/* estado 2 */ {3, 2, 4, 0},
/* estado 3 */ {3, 2, 4, 0},
/* estado 4 */ {6, 5, 0, 0},
/* estado 5 */ {6, 5, 0, 0},
/* estado 6 */ {6, 5, 0, 0}
}

/* ... "0" "1" "2" "3" ... 9 ... "." ... */
Int map[256] = { ..., 0, 1, 3, 3, ..., 3, ..., 2, ... }
```

Exemplo:

Edge[3][(int) '.'] é feito assim:
Edge[3][map[(int) '.']]

11

©Universidade do Algarve

Aula 8

Optimizações na implementação do DFA

- ✍ Para o exemplo apresentado:
 - ✍ Permitiu passar de um array com 256 ? 7 (1792) elementos para um array com 256 e outro array com 7 ? 4 (28) elementos
 - ✍ De 1792 para 284 elementos
- ✍ Existem outras formas de optimização...

12

©Universidade do Algarve

Aula 8