Universidade de Aveiro

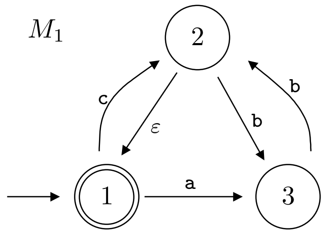
Departamento de Electrónica, Telecomunicações e Informática

Linguagens Formais e Autómatos / Compiladores

Exame intercalar modelo Maio de 2019

NºMec: **Nome:**

1. Sobre o alfabeto $A = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, considere a linguagem L_1, definida pelo autómato finito M_1, a linguagem L_2, definida pela gramática G_2 (cujo símbolo inicial é S_2) e a linguagem L_3.



$$S_2 \to a\,X$$
$$X \to b \mid b\,c\,b\,X \mid b\,S_2$$

$$L_3 = \{ab(c)^m(bb)^n \;:\; m > 0 \wedge n \geq 0\}$$

(a) Seja $L_4 = L_1 \cap L_2$. Das seguintes afirmações assinale as que são verdadeiras. (Uma resposta errada implica uma penalização.)

[x] $ab \in L_4$	[ ] $cabb \in L_4$
[x] $abab \in L_4$	[x] $abcbb \in L_4$

ab em M1: 1 -> a -> 3 -> b -> 2 -> \e -> 1
ab em G2: S2 => a X => a b

...

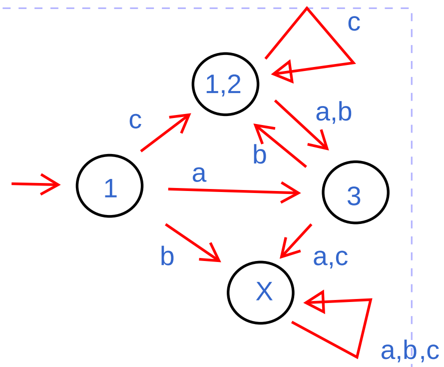
(b) Determine um autómatos finito determinista equivalente a M_1.

Usar-se-á a notação
A -> (x) -> B para
representar a transição do
estado A para o estado B
por ocorrência de um x

inicial : 1
(apenas ele, porque não há
transições-\e a sair dele)

1 -> (a) -> 3
1 -> (b) -> X
1 -> (c) -> 1,2
3 -> (a,c) -> X
3 -> (b) -> 1,2
1,2 -> (a,b) -> 3
1,2 -> (c) -> 1,2
X -> (a,b,c) -> X

aceitação : 1 e 1,2
(todos os que têm o 1)



(c) Obtenha um **autómato finito**, determinista ou não determinista, mas não generalizado, que reconheça a linguagem $L_5 = L_1 \cdot L_2$. Apresente os passos intermédios e/ou o raciocínio adequados para justificar a sua resposta.

Conversão de G2 para autómato
- Os estados S2 e X correspondem aos símbolos não terminais homónimos
-Os estados X1 e X2 foram introduzidos para quebrar a sequência bcb
- O estado A foi introduzido para a aceitação

inicial : S2

S2 -> (a) -> X
X -> (b) -> A
X -> (b) -> X1
X1 -> (c) -> X2
X2 -> (b) -> X
X -> (b) -> S

aceitação : A

Concatenação

-estado 1, de M1, deixa de ser aceitação
- aparece uma transição-\e desse estado para o inicial do seguinte

1 -> (\e) -> S2

(d) Das seguintes expressões regulares apenas uma representa a linguagem L_3. Assinale-a.

[ ] $abcc^*bb^*$ [x] $abcc^*(bb)^*$ ab(c)+(bb)*

[ ] $abc^*(bb)^*$ [ ] $abc(c|bb)^*$

(e) Das seguintes gramáticas apenas uma é uma gramática regular que representa a linguagem L_3. Assinale-a.

não é porque não é regular

[ ]
$$\begin{array}{l} S \to a\,b\,C\,B \\ C \to c \mid c\,C \\ B \to \varepsilon \mid b\,b\,B \end{array}$$

[ ]
$$\begin{array}{l} S \to a\,b\,c\,C \\ C \to c\,B \mid c\,C \\ B \to \varepsilon \mid b\,b\,B \end{array}$$

não é porque a seguir ao 'ab' obriga a ter 2 'c'

[x]
$$\begin{array}{l} S \to a\,b\,c\,C \\ C \to B \mid c\,C \\ B \to \varepsilon \mid b\,b\,B \end{array}$$

[ ]
$$\begin{array}{l} S \to a\,b\,C \\ C \to B \mid c\,C \\ B \to \varepsilon \mid b\,b\,B \end{array}$$

não é porque a seguir ao 'ab' pode ter 0 'c'

(f) Obtenha uma **expressão regular** que reconheça a linguagem L_1. Apresente os passos intermédios e/ou o raciocínio adequados para justificar a sua resposta.

inicial 1
final 1

1 -> (a) -> 3
1 -> (c) -> 2
2 -> (\e) -> 1
2 -> (b) -> 3
3 -> (b) -> 2

Garantir que inicial não tem arcos de entrada
inicial A
final 1

A -> (\e) -> 1
1 -> (a) -> 3
1 -> (c) -> 2
2 -> (\e) -> 1
2 -> (b) -> 3
3 -> (b) -> 2

Garantir que final (is) não tem(têm) arcos de saída
inicial A
final B

A -> (\e) -> 1
1 -> (a) -> 3
1 -> (c) -> 2
2 -> (\e) -> 1
2 -> (b) -> 3
3 -> (b) -> 2
1 -> (\e) -> B

Eliminar 3 (porque é o que introduz menos transições)
inicial A
final B

A -> (\e) -> 1
1 -> (c|ab) -> 2
2 -> (\e) -> 1
1 -> (\e) -> B
2 -> (bb) -> 2

Eliminar 2 (porque é agora que introduz menos transições)
inicial A
final B

A -> (\e) -> 1
1 -> (\e) -> B
1-> ((c|ab)(bb)*) -> 1

===========
Eliminar 1

A -> (((c|ab)(bb)*)*) -> B

(g) Mostre que $L_2 \subset L_1$. (Note que se trata do subconjunto em sentido estrito ($\subset$) e não em sentido lato ($\subseteq$).) Apresente os passos intermédios e/ou o raciocínio adequados para justificar a sua resposta.

Um ciclo num autómato corresponde ao fecho de Kleene

A gramática G2

S -> aX
X -> b | (bcb|ba)X

corresponde à expressão regular

a (bcb|ba)* b

Com o (a) pode-se ir de 1 para 3

1 -> (a) -> 3

Com (bcb) ou (ba) pode-se ir de 3 para 3, um ciclo

3 -> (b) -> 2 -> \e -> 1 -> c -> 2 -> b -> 3
3 -> b -> 2 -> \e -> 1 -> a -> 3

Logo com ((bcb|ba)*) pode-se ir de 3 para 3

Finalmente com (b) pode-se ir de 3 para 1

3 -> b -> 2 -> \e -> 1

Donde

1 -> (a) -> 3 -> ((bcb|ba)*) -> 3 -> (b) -> 1

2. Na linguagem Java um literal numérico inteiro pode ser escrito nas bases `2`, `8`, `10` e `16`. Os prefixos `0b`, `0` e `0x` são usados para representar, respetivamente, as bases 2, 8 e 16. A base 10 não tem prefixo. Por exemplo, `0b11`, `0743`, `1299` e `0x12fD` são literais numéricos válidos e `0b2` e `028` são inválidos.

   (.) Apresente uma expressão regular que represente os padrões válidos para os literais numéricos em Java. Pode definir a expressão regular pretendida a partir de outras mais simples.

```
NUM = N2 | N8 | N10 | N16

N2 = '0b' [01]+

N8 = '0' [0-7]*

N10 = [1-9] [0-9]*

N16 = '0x' [0-9a-fA-F]+
```