

Compiladores

Aula 3

Autômatos Finitos

Prof. Dr. Luiz Eduardo G. Martins

UNIFESP



Compiladores: Autômatos Finitos

- Um **Autômato Finito** é uma máquina de estados finitos que permite reconhecer, por meio de um conjunto de estados e transições dirigidas, pela ocorrência de símbolos de um **alfabeto**, se uma determinada cadeia de caracteres (*string*) pertence ou não a uma **linguagem regular**
- Os Autômatos Finitos podem ser usados para descrever o processo de reconhecimento de cadeias de entrada (identificação de *tokens*)
- Portanto, são muito úteis na construção do **Analizador Léxico**

Compiladores: Autômatos Finitos

- ▶ O núcleo do analisador léxico é uma implementação de um **autômato finito**
 - ▶ máquina de estados finitos que aceita símbolos de uma **string**
 - ▶ ao final da **string** indica se ela é válida para a gramática ou não
 - ▶ autômato é definido para cada conjunto de símbolos que deve ser reconhecido
- ▶ Cada *token* pode ser descrito por uma lista ou por uma expressão regular
 - Portanto, existe uma forte relação entre *token*, expressão regular, e autômato finito

Compiladores: Autômatos Finitos

- Definição Formal

Autômato é descrito por uma quintupla

$$M = (K, \Sigma, \delta, s, F)$$

Compiladores: Autômatos Finitos

- Definição Formal

Autômato é descrito por uma quintupla

$$M = (K, \Sigma, \delta, s, F)$$

K conjunto (finito) de estados

Σ alfabeto (finito) de entrada

δ conjunto de transições

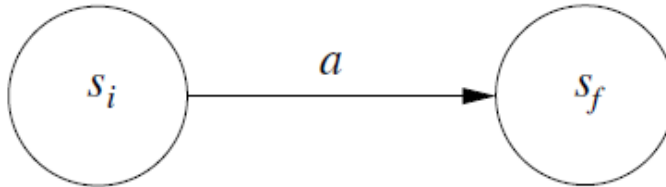
s estado inicial, $s \in K$

F conjunto de estados finais, $F \subseteq K$

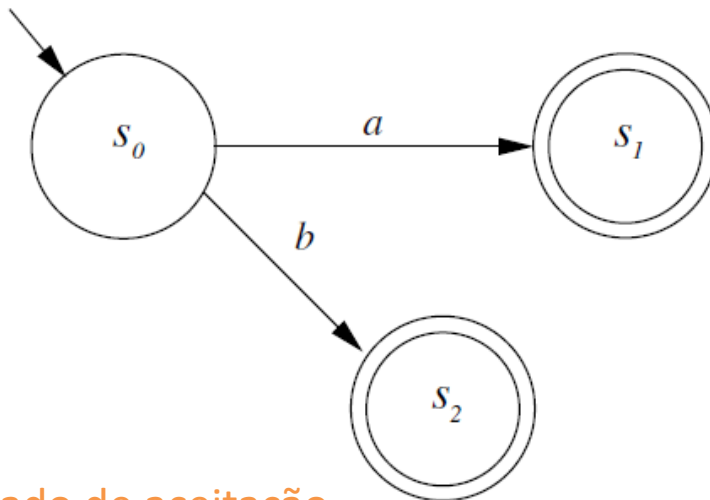
Compiladores: Autômatos Finitos

- Representação Gráfica

Estados e transição



Estados inicial e finais



Representação Formal:

$$K = \{s_0, s_1, s_2\}$$

$$\Sigma = \{a, b\}$$

$$\delta = \{(s_0, \{a\}, s_1), (s_0, \{b\}, s_2)\}$$

$$s = s_0$$

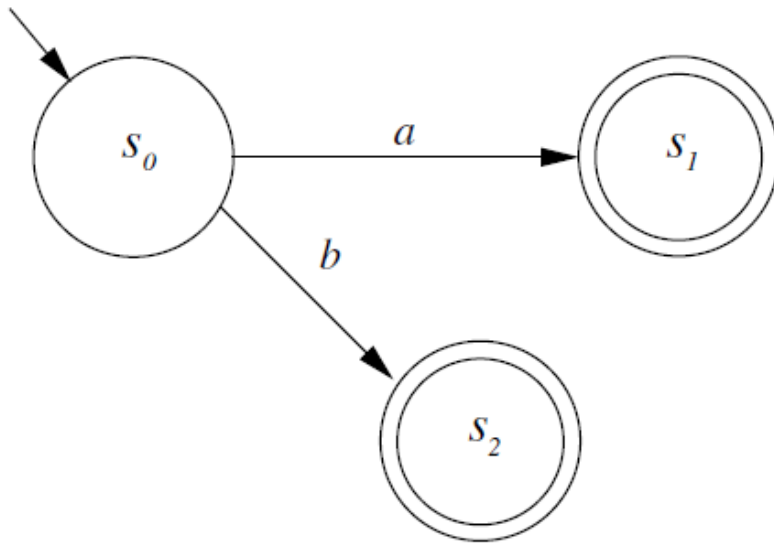
$$F = \{s_1, s_2\}$$

estado final = estado de aceitação
(um ou mais)

Compiladores: Autômatos Finitos

- Representação Tabular

Estados inicial e finais



	s_0	s_1	s_2
a	s_1	—	—
b	s_2	—	—

Estado inicial: s_0

Estados finais: s_1 , s_2

Compiladores: Autômatos Finitos

- Tipos de Autômatos
 1. Não-Determinísticos (AFND ou NFA)
 2. Determinísticos (AFD ou DFA)

Compiladores: Autômatos Finitos

- Não-Determinísticos
 - Pode existir mais de uma transição (aresta) saindo do mesmo estado com o mesmo símbolo
 - Pode existir transição de estado sem a ocorrência de nenhum símbolo de entrada
 - Transição pela *string* vazia (anotada por ϵ)

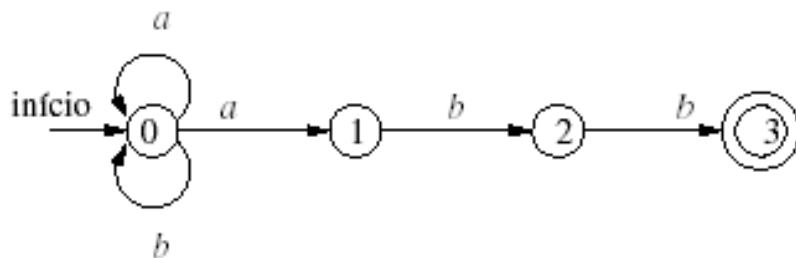


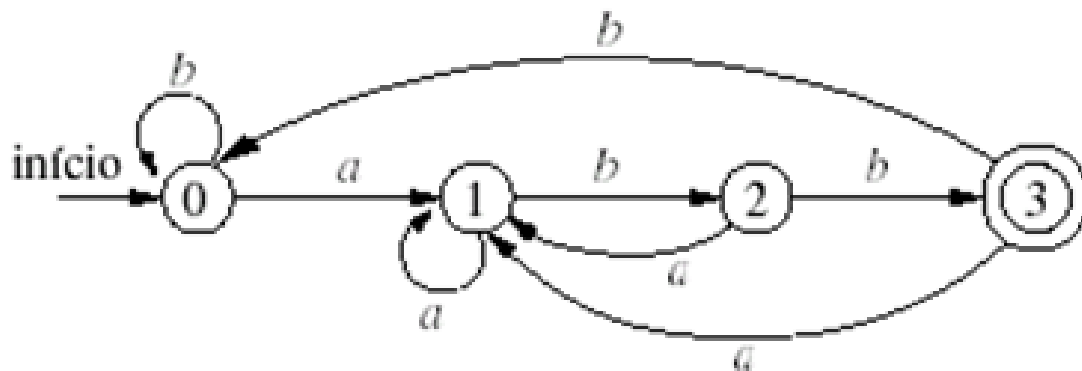
FIGURA 3.24 Um autômato finito não determinista.

ESTADO	<i>a</i>	<i>b</i>	ϵ
0	{0,1}	{0}	\emptyset
1	\emptyset	{2}	\emptyset
2	\emptyset	{3}	\emptyset
3	\emptyset	\emptyset	\emptyset

FIGURA 3.25 Tabela de transição para o NFA da Figura 3.24.

Compiladores: Autômatos Finitos

- Determinísticos
 - Não há várias alternativas para a transição a partir de um estado com o mesmo símbolo de entrada
 - Não há transições pela string vazia
 - O AFD é um caso especial de AFND



Compiladores: Autômatos Finitos

- Determinísticos
 - Não há várias alternativas para a transição a partir de um estado com o mesmo símbolo de entrada
 - Não há transições pela string vazia
 - O AFD é um caso especial de AFND

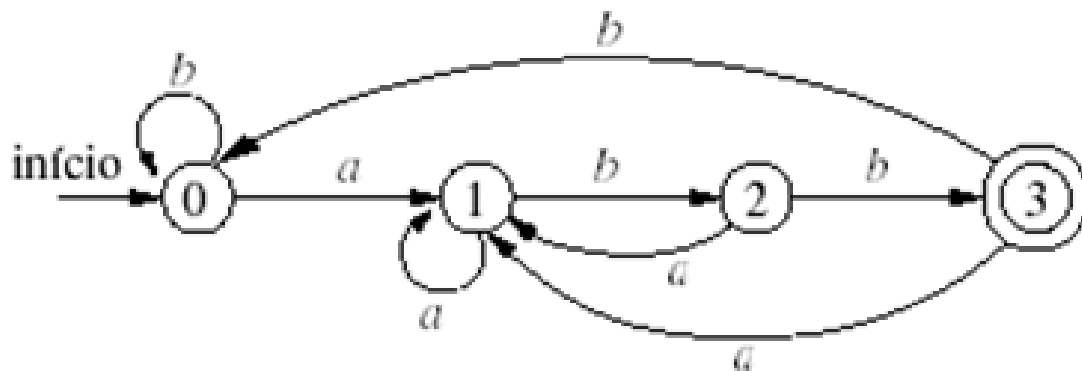


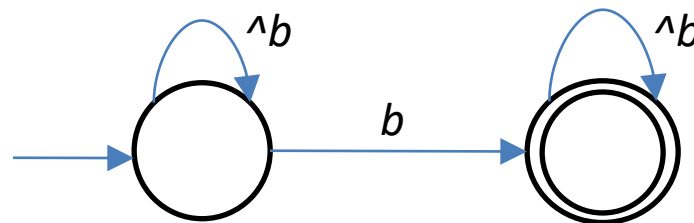
FIGURA 3.28 DFA aceitando $(alb)^+abb$.

Compiladores: Autômatos Finitos

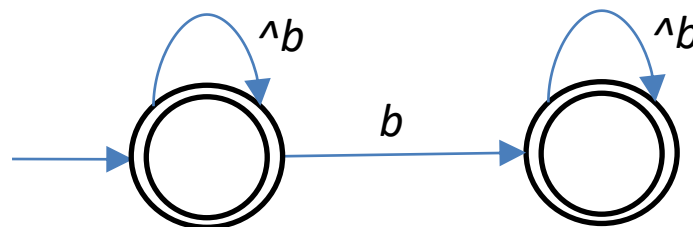
- Aceitação das cadeias de entrada pelos autômatos
 - Um autômato finito *aceita* (consome) a cadeia de entrada se e somente se houver algum caminho no grafo de transições: do estado inicial para um dos estados finais (*estados de aceitação*)

Compiladores: Autômatos Finitos

- Exemplos de AFDs $\Sigma = \{a, b, c, \dots, z\}$
 - Reconhecimento de cadeias de caracteres que contém **exatamente um b**

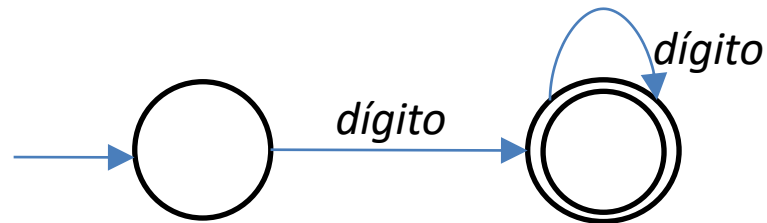


- Reconhecimento de cadeias de caracteres que contém **no máximo um b**



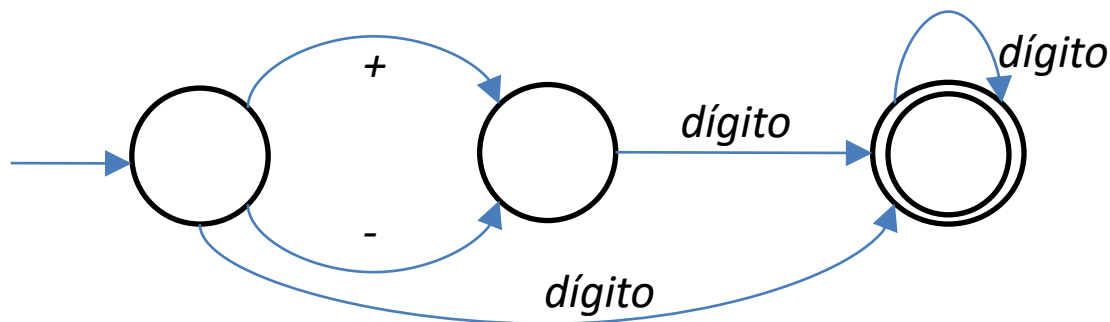
Compiladores: Autômatos Finitos

- Exemplos de AFDs
 - Reconhecimento de um número natural
 - Definições regulares
 - dígito = [0-9]
 - nat = dígito+



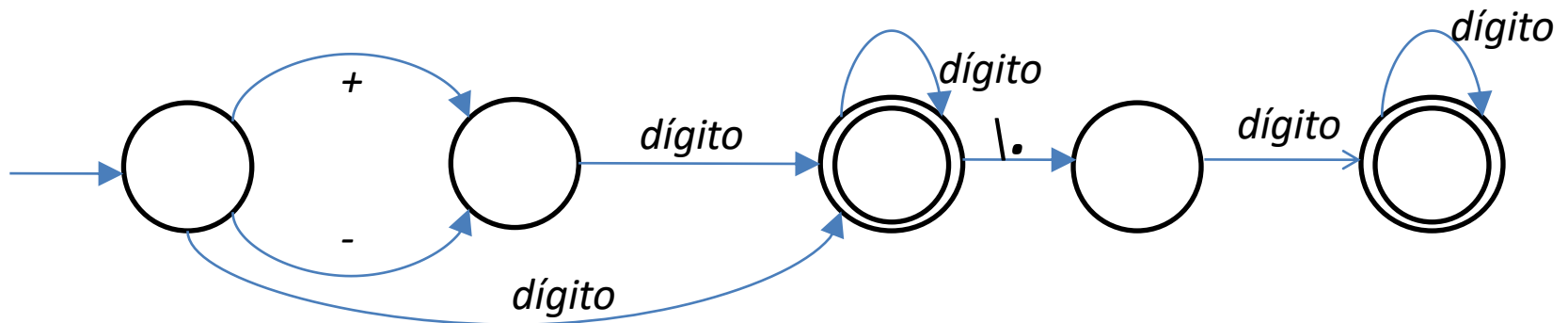
Compiladores: Autômatos Finitos

- Exemplos de AFDs
 - Reconhecimento de um número natural com sinal opcional
 - Definições regulares
 - $\text{dígito} = [0-9]$
 - $\text{nat} = \text{dígito}^+$
 - $\text{signedNat} = (+|-)? \text{nat}$



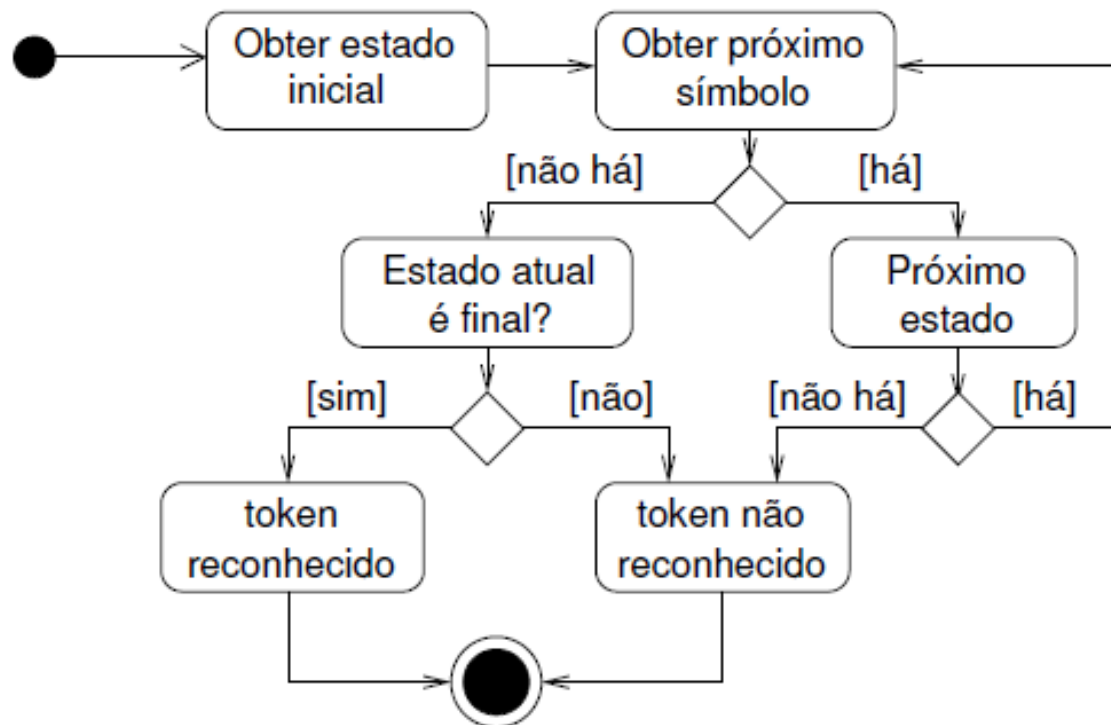
Compiladores: Autômatos Finitos

- Exemplos de AFDs
 - Reconhecimento de um número real
 - Definições regulares
 - $\text{dígito} = [0-9]$
 - $\text{nat} = \text{dígito}^+$
 - $\text{signedNat} = (+|-)? \text{nat}$
 - $\text{real} = \text{signedNat} (\backslash.\text{nat})?$



Compiladores: Autômatos Finitos

- Algoritmo para a identificação de cadeias feita por um AFD (notação de digrama de atividades da UML)



Compiladores: Autômatos Finitos

- Exercícios

1) Construa DFAs para as expressões regulares abaixo:

(a) $a (a \mid b \mid c \mid \dots \mid z)^* a \mid a$

(c) $(1 \mid 2 \mid 3 \mid \dots \mid 9) (0 \mid 1 \mid 2 \mid \dots \mid 9)^* \mid 0 \mid \epsilon$

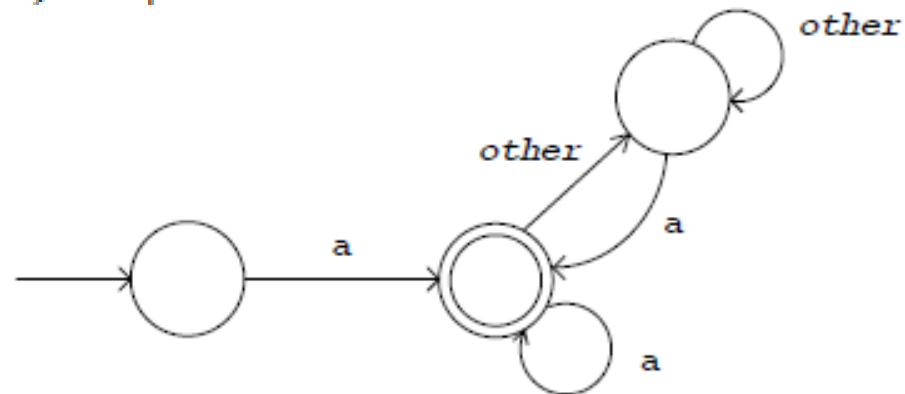
(e) $(0 \mid 1 \mid 2 \mid \dots \mid 8)^* (0 \mid 1 \mid 3 \mid 4 \mid \dots \mid 9)^*$

(g) $b^* a b^* (a b^* a b^*)^* \mid a^* b a^* (b a^* b a^*)^*$

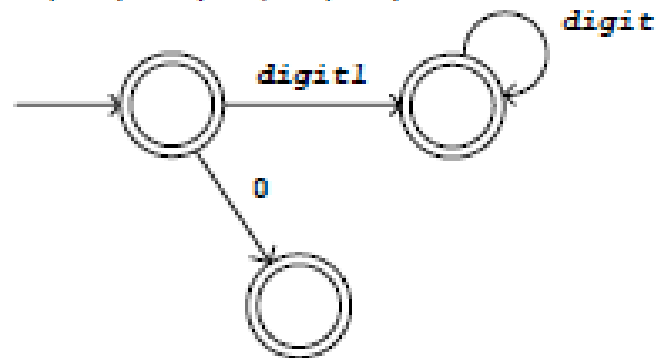
Compiladores: Autômatos Finitos

(a) $a (a | b | c | \dots | z)^* a | a$

other = $b | c | \dots | z$



(c) $(1 | 2 | 3 | \dots | 9) (0 | 1 | 2 | \dots | 9)^* | 0 | \epsilon$

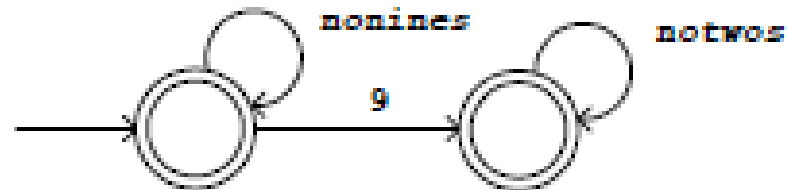


(digit1 = $1|2|3|\dots|9$)

(digit = $0|1|\dots|9$)

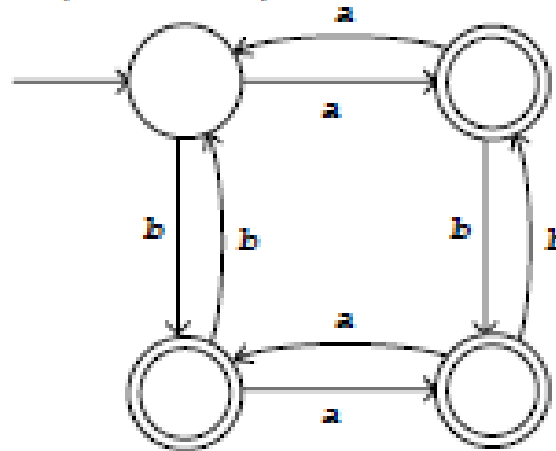
Compiladores: Autômatos Finitos

(e) $(0 | 1 | 2 | \dots | 8)^* (0 | 1 | 3 | 4 | \dots | 9)^*$



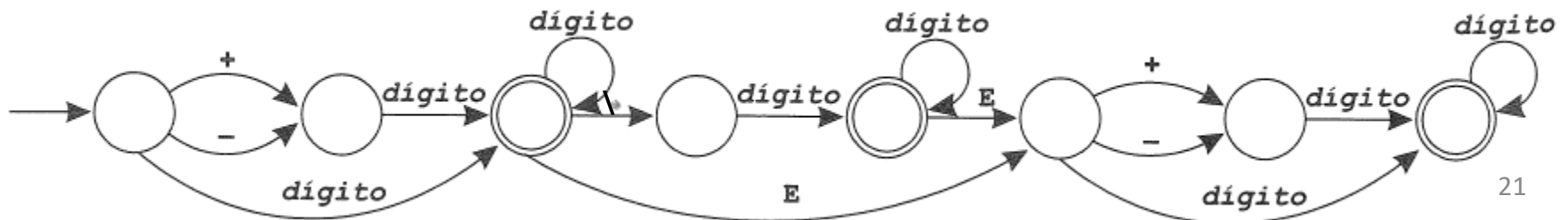
$(nonines = 1|2|3|\dots|8)$
 $(notwos = 0|1|3|4|\dots|9)$

(g) $b^*ab^*(ab^*ab^*)^* | a^*ba^*(ba^*ba^*)^*$



Compiladores: Autômatos Finitos

- Exercício
 - Reconhecimento de um número real com parte exponencial opcional
 - Definições regulares
 - $\text{dígito} = [0-9]$
 - $\text{nat} = \text{dígito}^+$
 - $\text{signedNat} = (+|-)? \text{nat}$
 - $\text{exp} = \text{E signedNat}$
 - $\text{real} = \text{signedNat} (\backslash.\text{nat})? \text{exp?}$



Compiladores: Autômatos Finitos

- Bibliografia consultada

RICARTE, I. **Introdução à Compilação**. Rio de Janeiro: Editora Campus/Elsevier, 2008.

AHO, A. V.; LAM, M. S.; SETHI, R. e ULLMAN, J. D. **Compiladores: princípios, técnicas e ferramentas**. 2ª edição – São Paulo: Pearson Addison-Wesley, 2008

LOUDEN, K. C. **Compiladores: princípios e práticas**. São Paulo: Pioneira Thompson Learning, 2004