

Compiladores

Análise Léxica

Prof. Tales Bitelo Viegas

<https://fb.com/ProfessorTalesViegas>



Análise Léxica

- ▶ Primeira fase de um compilador
- ▶ Lê os caracteres de entrada e produz uma sequência de símbolos léxicos válidos (tokens)
- ▶ Executa tarefas secundárias como:
 - Remover comentários
 - Remover espaços em branco
 - Controlar posição dos elementos, visando mensagens de erro ao programador



Tokens

- ▶ Elementos básicos de qualquer linguagem de programação
 - Numéricos: inteiros, reais, complexos, etc
 - Lógicos: true/false
 - Caracteres: “c”, “a”, “case”
 - Ponteiros
 - Identificadores/nomes: variáveis/nomes de procedimentos identificadas a um dado nome



Tokens

- ▶ Cada token é representado internamente por três informações:
 - **Classe:** representa o tipo do token reconhecido. Ex: identificadores, constantes numéricas, cadeias de caracteres, palavras reservadas, operadores e separadores
 - **Valor:** valor do token, dependendo da classe
 - **Posição:** linha e coluna do texto onde o token foi encontrado



Token – Exemplo

- ▶ while i<100 do i=j+i;
 - [while, , 1x1]
 - [id, i, 1x7]
 - [<, , 1x8]
 - [cte, 100, 1x9]
 - [do, , 1x13]
 - [id, i, 1x16]
 - [=, , 1x17]
 - [id, j, 1x18]
 - [+, , 1x19]
 - [id, i, 1x20]
 - [;, , 1x21]



Reconhecimento de Tokens

- ▶ Diagramas de Transição (Autômatos)
 - Apresentam ações executadas pelo Analisador Léxico
 - Controlam as informações a respeito de caracteres que são examinados, com a leitura do código-fonte



Reconhecimento de Tokens

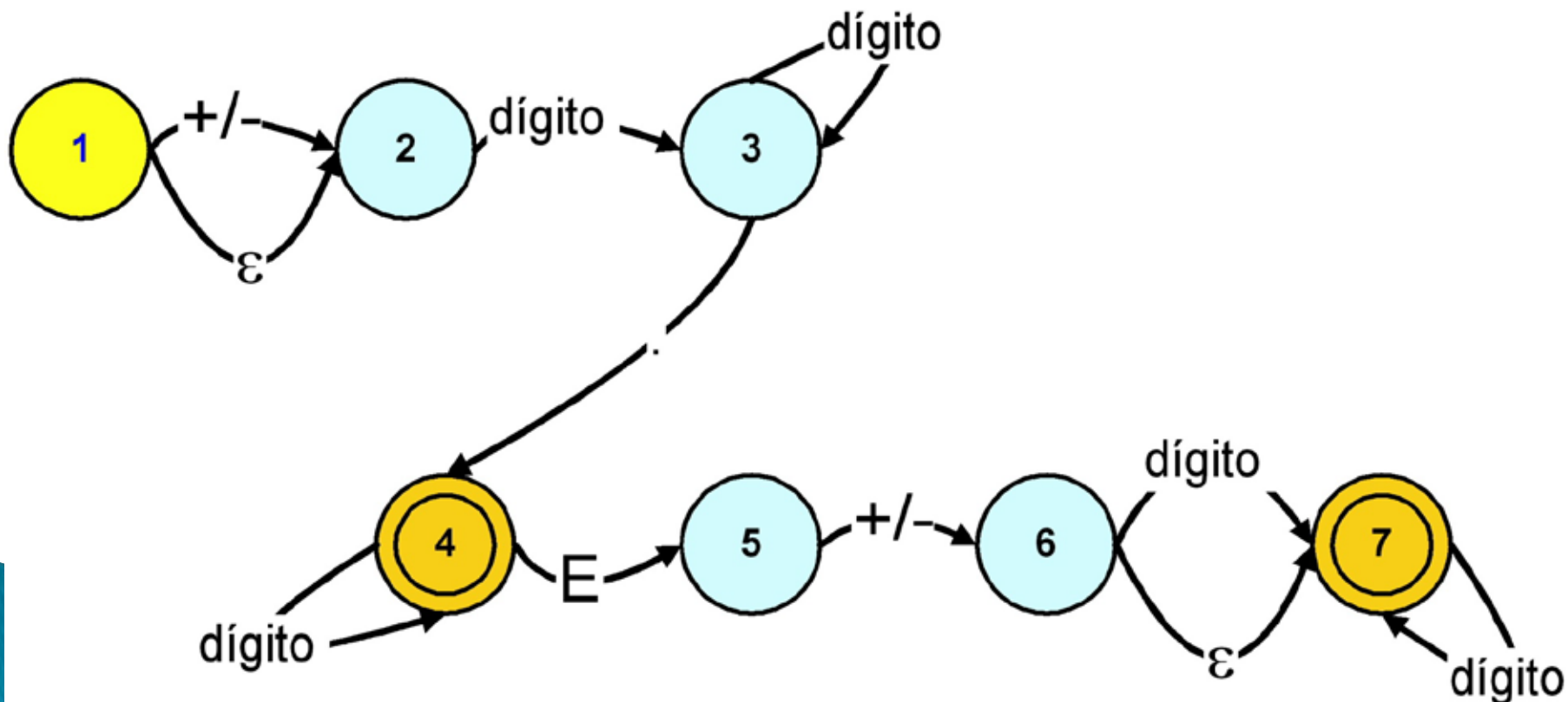
- ▶ Elementos dos Diagramas:
 - Estados: posições no diagrama
 - Transições/Lados: setas que conectam os estados
 - Rótulos: indicam os caracteres de entrada que podem aparecer após atingir-se um dado estado
- ▶ Chamamos o diagrama de *determinístico* quando o mesmo símbolo não figura como rótulo de lados diferentes que deixem o mesmo estado



Diagramas de Transição

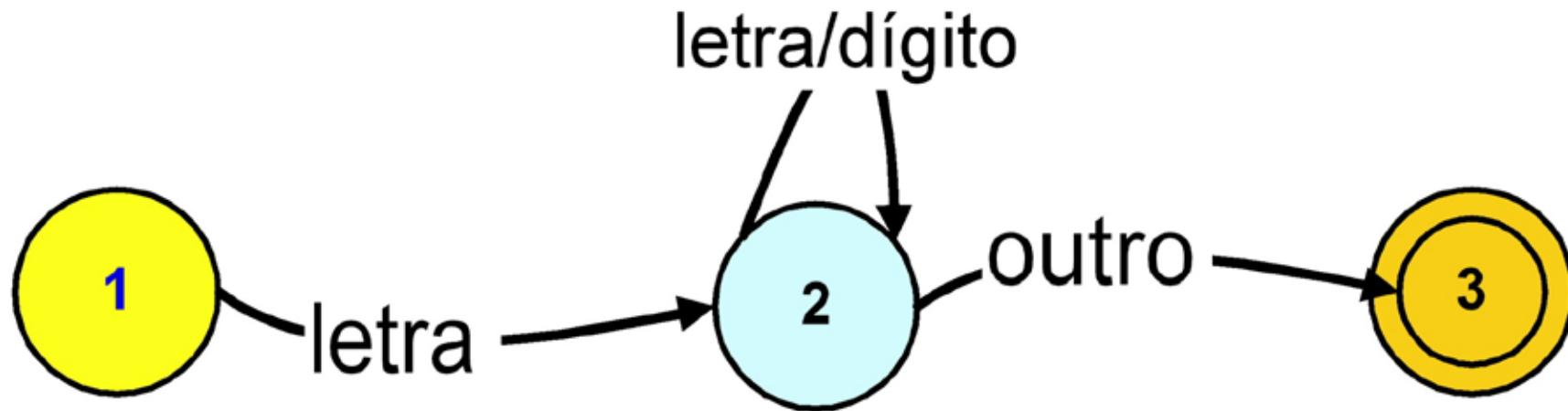
Supondo a que um número real possa ser dado por:

$\langle \text{sinal} \rangle \langle \text{parte_inteira} \rangle . \langle \text{parte_fracionária} \rangle E \langle \text{expoente} \rangle$



Diagramas de Transição

- ▶ Para reconhecimento de identificadores que começam com uma letra e depois possuem uma ou mais letra ou dígito



Autômatos Finitos

- ▶ Os símbolos que deverão ser reconhecidos na análise léxica são representáveis por gramáticas regulares
- ▶ Há uma correspondência unívoca entre gramáticas regulares e *autômatos finitos*



Autômatos Finitos

- ▶ Máquinas (rotinas) que podem ser utilizadas para reconhecer *strings* de uma dada linguagem.
- ▶ São compostos por:
 - Um conjunto de **estados**, alguns dos quais são denominados **estados finais**.
 - A medida que caracteres da string de entrada são lidos, o controle da máquina passa de um estado a outro
 - Um conjunto de **regras de transição** entre os estados



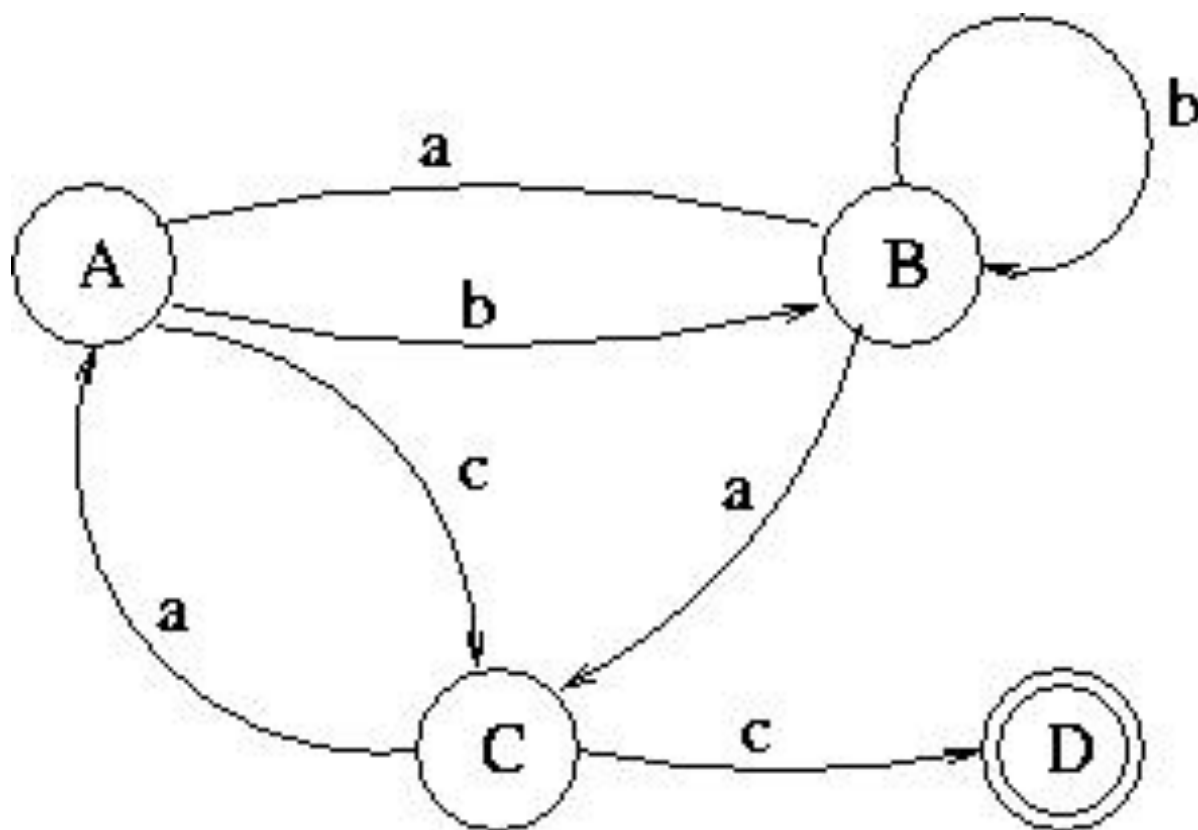
Autômatos Finitos

- ▶ Formalmente um autômato é descrito por cinco características:
 - Um conjunto finito de estados
 - Um alfabeto de entrada finito
 - Um conjunto de transições
 - Um estado inicial
 - Um conjunto de estados finais
- ▶ Quando, partindo de um estado inicial, varrendo a sentença dada (de acordo com as regras de transições), consegue-se atingir um estado final, a sequência dada é parte da linguagem



Autômatos Finitos

- ▶ Forma gráfica de representação de um Autômato Finito



Autômatos Finitos

- ▶ Forma tabular de representação do mesmo autômato

	a	b	c
A	B	B	C
B	C	B	-
C	A	-	D
D	-	-	-



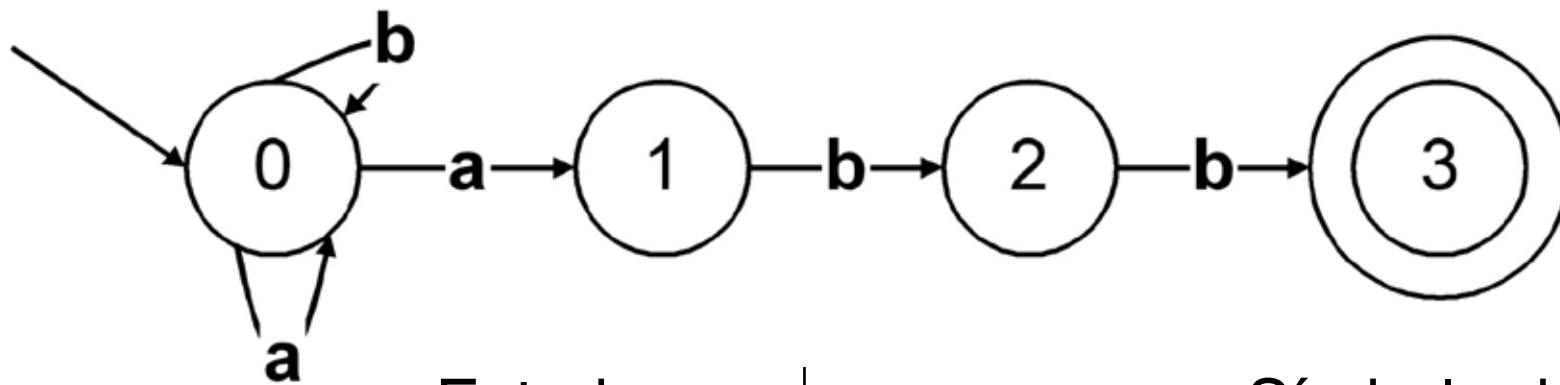
AF Não-determinísticos

- ▶ Um autômato de estados finito não-determinístico é:
 - Aquele em que pode ocorrer a transição vazia (aquela em que o autômato pode passar de um estado a outro sem que ocorra a entrada de um caracter do string)
 - Aquele onde podem ocorrer indeterminações na ação, isto é, determinados estados podem ter mais do que uma transição possível para dado caracter



AF Não-Determinísticos

- Exemplo: reconhecimento da linguagem $(a|b)^*abb$



Estado	Símbolo de Entrada	
	a	b
0	{0,1}	{0}
1		{2}
2		{3}

Exercício 1

- ▶ Para cada um dos padrões abaixo crie um autômato para reconhecê-lo:
 - Números Reais – Ex: 99.99, 99999.99, – 123.12, +123.4
 - Strings – Ex: “Tales Bitelo Viegas”
 - Comentários – Ex: /* */
 - Números Inteiros – Ex: 1234, 567, 9, 0, –123



Exercício 2

- ▶ Agora crie um único autômato que reconheça os 4 padrões anteriores

