

Compiladores

Aula 6

Análise Léxica

Implementação de AFD

Prof. Dr. Luiz Eduardo G. Martins

UNIFESP

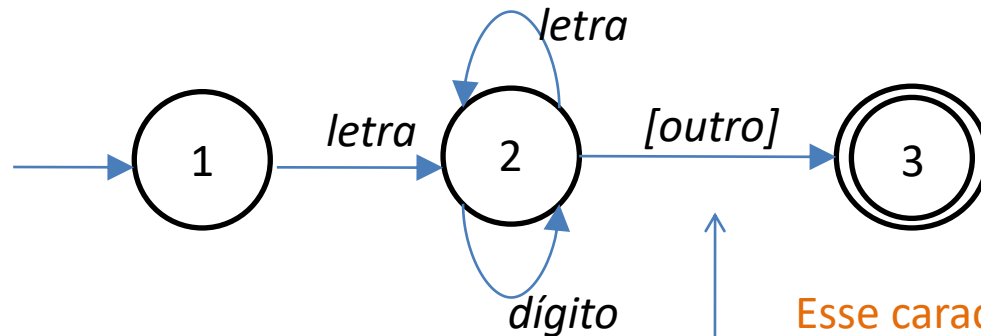
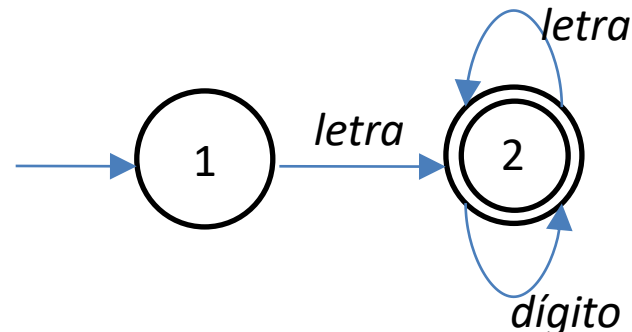


Implementação de AFD

- Há muitas formas de traduzir um AFD em código, considere o seguinte AFD para reconhecimento de identificadores

ER para identificadores

$\text{letra}(\text{letra} \mid \text{dígito})^*$



Esse caractere
não será
consumido pelo
autômato

Implementação de AFD

- Podemos simular esse AFD escrevendo o código da seguinte forma:

{início – estado 1}

If próximo caractere for letra **then**

avance entrada;

 {estado 2}

while próximo caractere for letra ou dígito

avance entrada; {permanece no estado 2}

end while;

 {passa para o estado 3 sem avançar entrada}

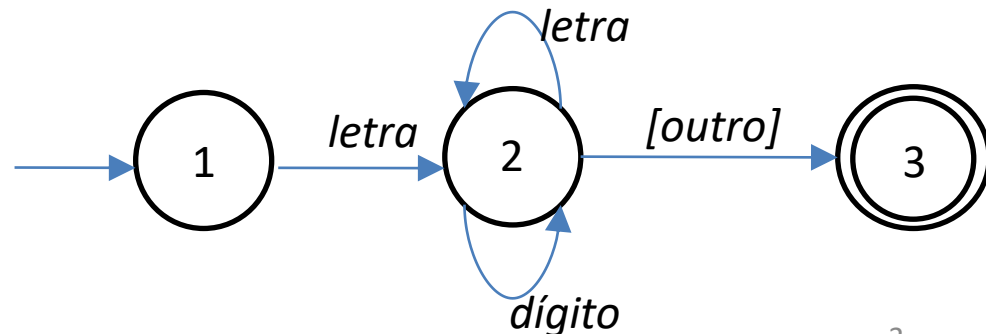
aceitação;

else

 {erro ou outros casos}

end if;

Indicado apenas
para sistemas de
varredura muito
simples



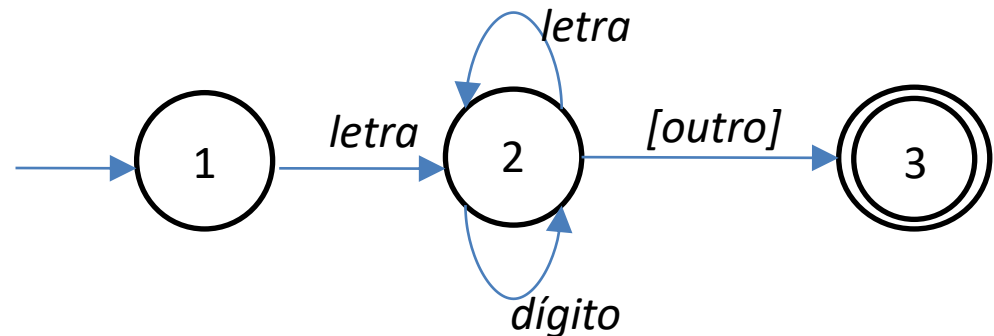
Implementação de AFD

- Uma alternativa melhor de implementação, seria usar uma variável para manter o estado corrente e tratar as transições como declarações *case*

```
estado := 1; {início}
while estado = 1 ou 2
  case estado of
    1: case caractere de entrada of
        letra: avance entrada; estado := 2;
        default estado := 4 {erro ou ignore};
      end case;
    2: case caractere de entrada of
        letra, dígito: avance entrada;
        default estado := 3;
      end case;
  end case;
end while;
If estado = 3 then aceitação else erro ou ignore;
```

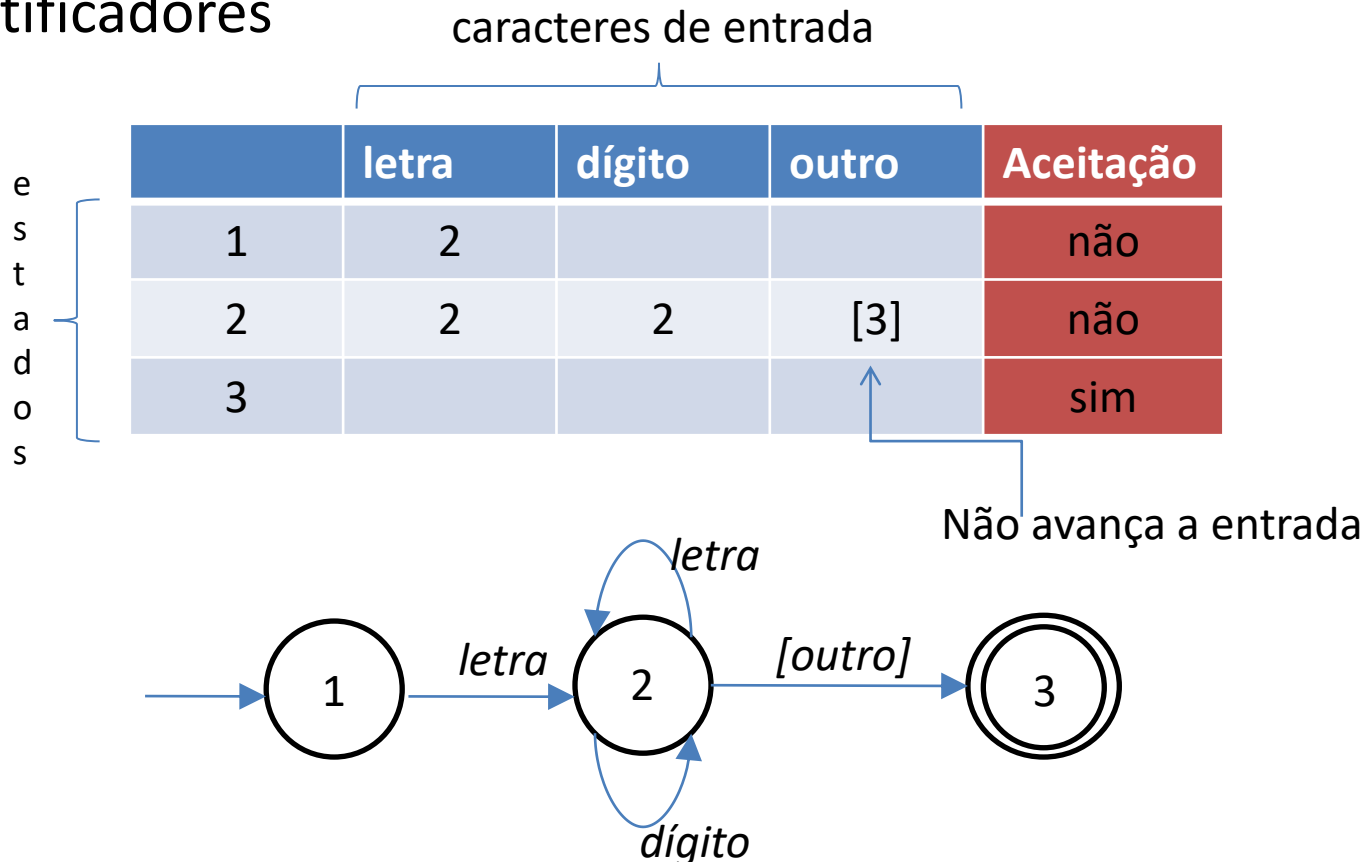
Exemplos:

x=0;
nota01=7;
nota02=6;



Implementação de AFD

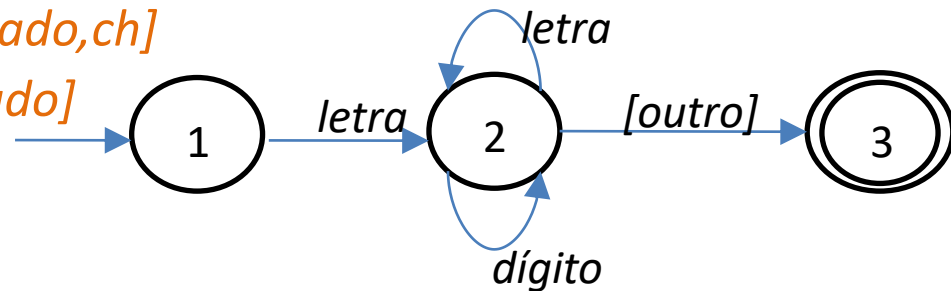
- Algoritmos **dirigidos por tabela** proporcionam a implementação de um código genérico para AFD (máquina de estados)
- Considere a tabela a seguir para o AFD de reconhecimento de identificadores



Implementação de AFD

- Um algoritmo dirigido por tabela para o AFD de reconhecimento de identificadores poderia utilizar as seguintes matrizes:

- Uma matriz de inteiros $T[\text{estado}, \text{ch}]$
- Uma matriz booleana $\text{Avance}[\text{estado}, \text{ch}]$
- Uma matriz booleana $\text{Aceita}[\text{estado}]$



estado := 1;

erro := F;

ch := próximo caractere de entrada;

if *ch* **not** letra **then** erro := V;

while **not** Aceita[estado] **and** **not** erro

novοestado := T[estado, ch];

if Avance[estado, ch] **then** *ch := próximo caractere de entrada;*

estado := novοestado;

end while;

if Aceita[estado] **then** aceitação;

Exemplos:

$x = 0;$

$n1 = 2;$

$n2 = 3;$

	letra	dígito	outro	Aceitação
1	2 V			F
2	2 V	2 V	3 F	F
3				V

Implementação de AFD

- Cabe ao sistema de varredura (analisador léxico) ler os caracteres do código-fonte e organizá-los em unidades lógicas para as outras partes do compilador (como o analisador sintático)
- É necessário implementar, como parte do sistema de varredura, uma rotina que leia os caracteres do arquivo de entrada (programa fonte), e retorne o caractere lido

Implementação de AFD

- Como alternativa, considere a função *proximoChar()* apresentada no código a seguir:

```
#include <fstream>
#include <iostream>

ifstream arq("teste.txt");

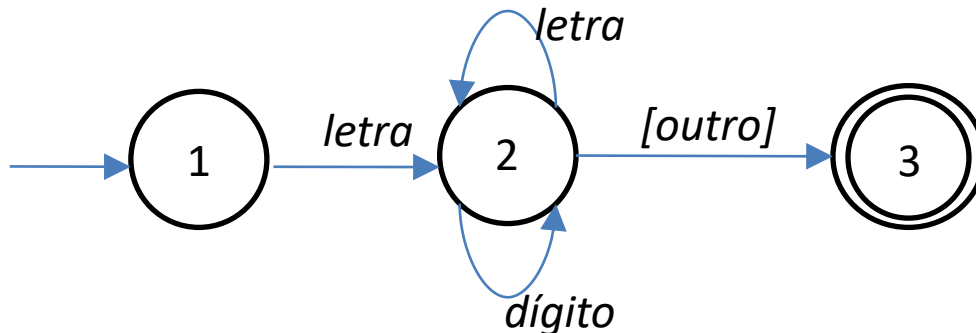
char proximoChar()
{
    char ch;
    arq.get(ch); // lê um caractere do arquivo texto
    return(ch);
}

int main()
{
    char ch;
    while(!arq.eof())
    {
        ch = proximoChar();
        cout << ch;
    }
    return 0;
}
```


Implementação de AFD

- Atividade:

Escrever um programa (em C ou C++) que leia o arquivo sort.txt. Implemente o algoritmo dirigido por tabela para reconhecer identificadores (conforme AFD abaixo). Gerar um programa de saída, substituindo todos os identificadores pelo token **ID**.



letra = [a-zA-Z]

dígito = [0-9]

ID = letra(letra | dígito)*

Implementação de AFD

- Bibliografia consultada

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

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