

# Compiladores

## Análise Léxica

### A Linguagem *TINY*

Prof. Dr. Luiz Eduardo G. Martins  
(adaptado por Profa Dra Ana Carolina Lorena)  
UNIFESP

# A Linguagem *TINY*

- A linguagem *TINY* será usada como exemplo para ilustrar características essenciais de um compilador
  - Ver seção 1.7 do livro “Compiladores: Princípios e Práticas”, Kenneth C. Louden

# A Linguagem *TINY*

- Características gerais da linguagem TINY:
  - Sequências de declarações separadas por ponto-e-vírgula (com exceção da declaração final)
  - Todas as variáveis são do tipo inteiro
  - Variáveis declaradas pela atribuição de valores
  - Há duas declarações de controle: *if* e *repeat*
  - *If* tem como opcional uma parte *else*, e precisa terminar com a palavra-chave *end*

# A Linguagem *TINY*

- Características gerais da linguagem TINY:
  - Há declarações para entrada e saída  
*read*  
*write*
  - Permite comentários - aparecem entre chaves
  - Permite expressões aritméticas com inteiros  
*+* *-* *\** */* (divisão inteira)
  - Uma expressão aritmética pode ter constantes, variáveis e parênteses

# A Linguagem *TINY*

- Características gerais da linguagem TINY:
  - Permite expressões booleanas
    - Expressão booleana composta por uma comparação de duas expressões aritméticas
    - Operadores relacionais:  $<$  e  $=$
  - A linguagem não contempla:
    - Procedimentos
    - Matrizes
    - Estruturas
    - Valores em ponto flutuante

# A Linguagem *TINY*

- Exemplo de um programa em *TINY*

```
{ Sample program
  in TINY language -
  computes factorial
}
read x; { input an integer }
if 0 < x then { don't compute if x <= 0 }
  fact := 1;
  repeat
    fact := fact * x;
    x := x - 1
  until x = 0;
  write fact { output factorial of x }
end
```

# A Linguagem *TINY*

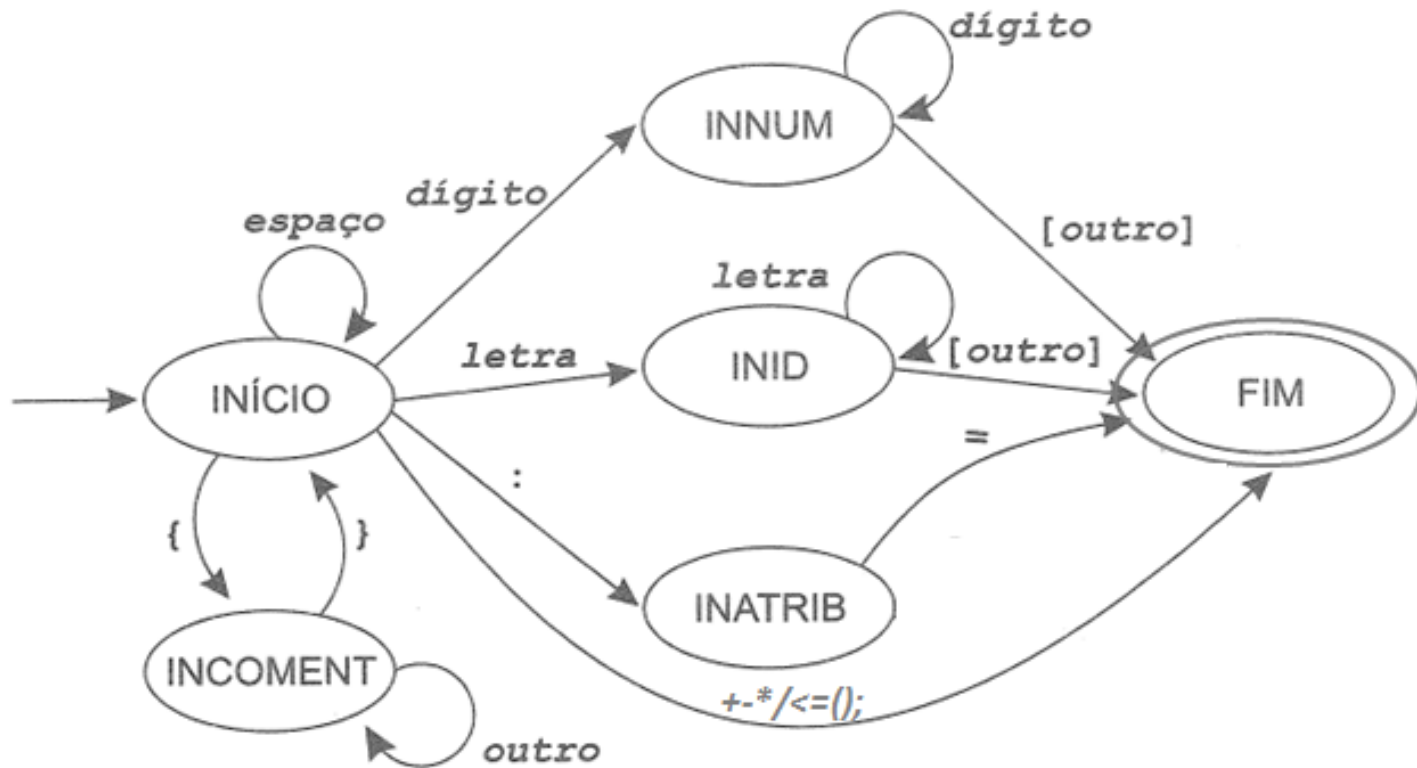
- Estrutura léxica da linguagem *TINY*

Tabela 2.1 Marcas da linguagem TINY

Palavras Reservadas	Símbolos Especiais	Outras
<i>if</i>	+	<i>número</i>
<i>then</i>	-	(1 ou mais
<i>else</i>	*	dígitos)
<i>end</i>	/	
<i>repeat</i>	=	
<i>until</i>	<	<i>identificador</i>
<i>read</i>	(	(1 ou mais
<i>write</i>	)	letras)
	;	
	:=	

# A Linguagem *TINY*

- AFD para o analisador léxico de *TINY*



*[outro]* : não avança  
caractere



# A Linguagem *TINY*

- Exemplo:

```
{ Sample program
  in TINY language -
  computes factorial
}
read x; { input an integer }
if 0 < x then { don't compute if x <= 0 }
  fact := 1;
  repeat
    fact := fact * x;
    x := x - 1
  until x = 0;
  write fact { output factorial of x }
end
```

# A Linguagem *TINY*

- Exemplo:

```
TINY COMPILATION: sample.tny
1: { Sample program
2:   in TINY language -
3:   computes factorial
4: }
5: read x; { input an integer }
   5: reserved word: read
   5: ID, name = x
   5: ;
6: if 0 < x then { don't compute if x <= 0 }
   6: reserved word: if
   6: NUME, val = 0
   6: <
   6: ID, name = x
   6: reserved word: then
7: fact := 1;
   7: ID, name = fact
   7: :=
   7: NUM, val = 1
   7: ;
...
```

# A Linguagem *TINY*

- Bibliografia consultada

Capítulo 2 e Apêndice B: LOUDEN, K. C. **Compiladores: princípios e práticas**. São Paulo: Pioneira Thompson Learning, 2004