

Prof. Dr. Daniel Leal Souza

Autômatos e Compiladores (EC8MA)

Compiladores (CC6MA)

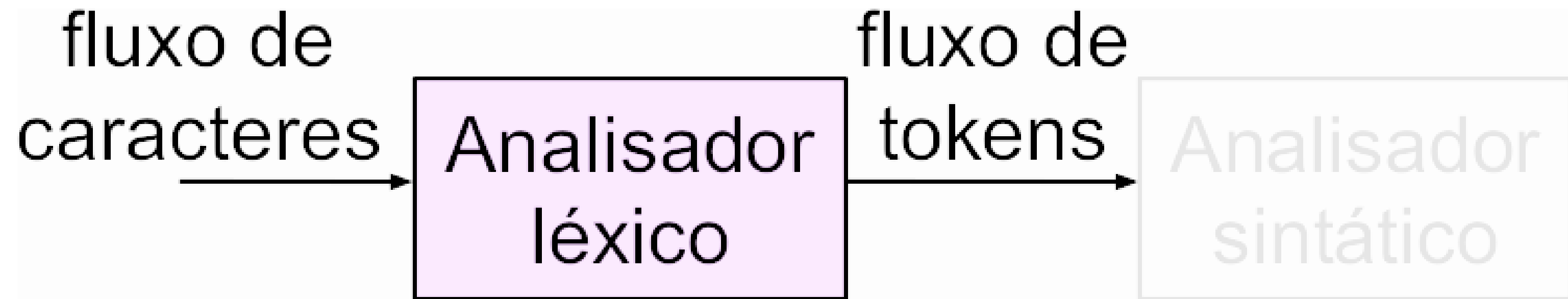
daniel.leal.souza@gmail.com
daniel.souza@prof.cesupa.br

2024



Analizador Léxico (scanning)

- Esquema:



Analizador Léxico (scanning)

- **Outras tarefas do analisador léxico**
 - Remover comentários
 - Remover espaços em branco
 - incluindo tabulação, enter, fim de linha
 - Correlacionar mensagens de erro com o programa fonte
 - Ex: número de linha, coluna.

Analizador Léxico (scanning)

- **Lexema** vs padrão vs token

- **Lexema:**

- sequência de caracteres do programa fonte

- **Por exemplo:**

```
if(var1 > 37){  
    outraVar += 54;  
    System.out.println("ok");  
}
```

- **Lexemas neste trecho:**

if	(var1	>	37
)	{	outraVar	+=	54
;	System	.	out	.
println	("ok")	;
}				

Analizador Léxico (scanning)

- Lexema vs **padrão** vs token
 - Padrão:
 - Caracteriza CLASSES de lexemas
 - Por exemplo:

```
if(var1 > 37){  
    outraVar += 54;  
    System.out.println("ok");  
}
```
 - Classes de lexemas neste trecho:

identificadores	var1, outraVar, System, out, println
constantes (numéricas)	37, 54
constantes (cadeias)	"ok"
operadores	>, +=
palavras-chave	if
...	

Analizador Léxico (scanning)

- Lexema vs **padrão** vs token
 - Um padrão é utilizado pelo analisador léxico para:
 - Reconhecer lexemas
 - E classificá-los
 - **Por exemplo:**

Classe	Lexemas	Padrão
identificadores	var1, outraVar, System, out, println	Cadeia de caracteres começando com letra
constantes (numéricas)	37, 54	Sequência de dígitos
constantes (cadeias)	"ok"	Cadeia de caracteres envolta por aspas
operadores	>, +=	O próprio lexema
palavras-chave	if	O próprio lexema
...		

Analizador Léxico (scanning)

- Lexema vs padrão vs **token**

- Token: Unidade léxica correspondente a um lexema

- Estrutura de dados:

```
class Token {  
    TipoToken tipo;  
    String valor;  
}
```

<tipo, valor>

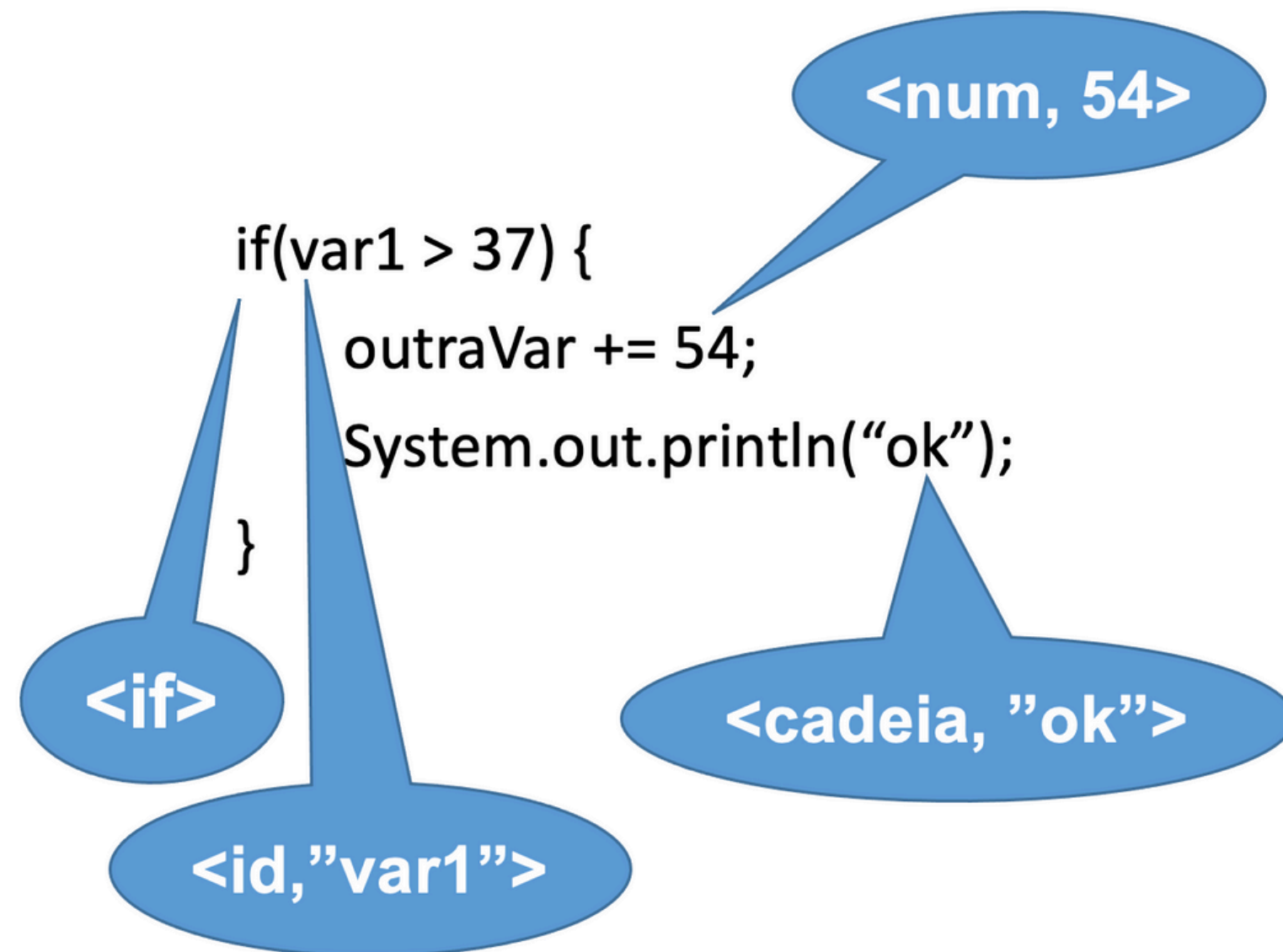
- Tipo do token: usado pelo analisador sintático

- Valor: o próprio lexema ou outras informações:

- Valor numérico, se for uma constante
- Ponteiro para a tabela de símbolos

Analizador Léxico (scanning)

- **Lexema vs padrão vs token**
 - Alguns tokens reconhecidos neste exemplo:



Classe	Padrão	Sigla
identificadores	Cadeia de caracteres começando com letra	id
constantes (numéricas)	Sequência de dígitos	num
constantes (cadeias)	Cadeia de caracteres envolta por aspas	cadeia
operadores	O próprio lexema	op
palavras-chave	O próprio lexema	o próprio lexema

DÚVIDAS?

Analizador Léxico (scanning)

- Exemplo: linguagem ALGUMA
 - **AL**Goritmos **U**sados para **M**ero **A**prendizado
 - Usaremos essa linguagem nos exemplos a seguir
 - É uma linguagem de programação simples com:
 - Declaração de variáveis (inteiras e reais)
 - Expressões aritméticas (+, -, *, /)
 - Expressões relacionais (>, >=, <=, <, =, <>)
 - Expressões lógicas (E, OU)
 - Condicional (SE-ENTÃO-SENÃO)
 - Repetição (ENQUANTO)

Analizador Léxico (scanning)

- **Exemplo: linguagem ALGUMA**

:DECLARACOES

argumento: INT

fatorial: INT

:ALGORITMO

% Calcula o fatorial de um número inteiro

LER argumento

ATRIBUIR argumento A fatorial

SE argumento = 0 ENTÃO

 ATRIBUIR 1 A fatorial

ENQUANTO argumento > 1

 INICIO

 ATRIBUIR fatorial * (argumento - 1) A fatorial

 ATRIBUIR argumento - 1 A argumento

 FIM

IMPRIMIR fatorial

Analizador Léxico (scanning)

:DECLARACOES

argumento:INT

fatorial:INT

:ALGORITMO

LER argumento

fatorial = argumento

SE argumento = 0 ENTÃO

 fatorial = 1

ENQUANTO argumento > 1

 INICIO

 fatorial = fatorial*argumento-1

 argumento = argumento-1

 FIM

IMPRIMIR fatorial

Versão ANTLR

Analizador Léxico (scanning)

- Exemplo: linguagem ALGUMA

Padrão	Tipo de lexema	Sigla
DECLARACOES, ALGORITMO, INTEIRO, REAL, ATRIBUIR, A, LER, IMPRIMIR, SE, ENTAO, ENQUANTO, INICIO, FIM	Palavras-chave	3 primeiras letras
$*, /, +, -$	Operadores aritméticos	OpArit
$<, <=, >=, >, =, <>$	Operadores relacionais	OpRel
E, OU	Operadores booleanos	OpBool
:	Delimitador	Delim
(,)	Parêntesis	AP / FP
Seq, de letras ou números que começam com letra minúscula	VARIÁVEL	Var
Sequências de dígitos (sem vírgula)	NÚMERO INTEIRO	NumI
Sequências de dígitos (com vírgula)	NÚMERO REAL	NumR
Sequências de caracteres envolta por aspas	CADEIA	Str

Analizador Léxico (scanning)

- Identifique os tokens do programa abaixo, conforme os padrões da linguagem ALGUMA.
No campo “valor”, armazene o lexema se necessário

```
:DECLARACOES
argumento: INT
fatorial: INT

:ALGORITMO
% Calcula o fatorial de um número inteiro
LER argumento
ATRIBUIR argumento A fatorial
SE argumento = 0 ENTAO
  ATRIBUIR 1 A fatorial

ENQUANTO argumento > 1
  INICIO
    ATRIBUIR fatorial * (argumento - 1) A fatorial
    ATRIBUIR argumento - 1 A argumento
  FIM
IMPRIMIR fatorial
```

Padrão	Sigla
DECLARACOES, ALGORITMO, INTEIRO, REAL, ATRIBUIR, A, LER, IMPRIMIR, SE, ENTAO, ENQUANTO, INICIO, FIM	3 primeiras letras
*, /, +, -	OpArit
<, <=, >=, >, =, <>	OpRel
E, OU	OpBool
:	Delim
(,)	AP / FP
Sequências de letras ou números que começam com letra	Var
Sequências de dígitos (sem vírgula)	NumI
Sequências de dígitos (com vírgula)	NumR
Sequências de caracteres envolta por aspas	Str

Analizador Léxico (scanning)

- **Resposta:**

<Delim> <DEC> <Var,"argumento"> <Delim> <INT> <Var,"fatorial">
<Delim> <INT> <Delim> <ALG> <LER> <Var,"argumento"> <ATR>
<Var,"argumento"> <A> <Var,"fatorial"> <SE> <Var,"argumento">
<OpRel,"="> <Numl,"0"> <ENT> <ATR> <Numl,"1"> <A> <Var,"fatorial">
<ENQ> <Var,"argumento"> <OpRel,">"> <Numl,"1"> <INI> <ATR>
<Var,"fatorial"> <OpArit,"*"> <AP> <Var,"argumento"> <OpArit,"-">
<Numl,"1"> <FP> <A> <Var,"fatorial"> <ATR> <Var,"argumento">
<OpArit,"-"> <Numl,"1"> <A> <Var,"argumento"> <FIM> <IMP>
<Var,"fatorial"> <EOF>

Analizador Léxico (scanning)

- **Lexema vs padrão vs token**
 - Classes típicas de tokens
 1. Um token para cada palavra-chave (obs: o padrão é o próprio lexema ou a própria palavra-chave)
 2. Tokens para os operadores
 - Individualmente ou em classes (como aritméticos vs relacionais vs booleanos)
 3. Um token representando todos os identificadores
 4. Um ou mais tokens representando constantes, como números e cadeias literais
 5. Tokens para cada símbolo de pontuação
 - Como parênteses, dois pontos, etc...

Analizador Léxico (scanning)

- **Lexema vs padrão vs token**
 - Erros simples podem ser detectados
 - Ex: 123x&\$33
 - Mas para muitos erros o analisador léxico não consegue sozinho
 - Por exemplo:
while (i>3) { }
 - É um erro léxico, mas como saber sem antes analisar a sintaxe?
 - A culpa é dos identificadores
 - O padrão dos identificadores é muito abrangente, então quase tudo se encaixa
 - Em outras palavras: quase sempre existe um padrão que reconhece o lexema

Analizador Léxico (scanning)

- Lexema vs padrão vs token
 - Erros simples podem ser detectados

- Ex: 107.8077

- Mas

-

-

- A c

-

- Em

- lexe

Erros léxicos aparecem apenas quando não existe um padrão capaz de reconhecer a entrada!

Analizador Léxico (scanning)

- Identifique os tokens do programa abaixo, conforme os padrões da linguagem ALGUMA.
No campo “valor”, armazene o lexema se necessário

:DECLARACOES
argumento: INT
fatorial: INT

:ALGORITMO
% Calcula o fatorial de um número inteiro
LER argumento
ATRIBUIR argumento A fatorial

SE argumento == 0 ENTAO
ATRIBUIR 1 A **F@atorial**

ENQUANTO argumento > 1
INICIO
ATRIBUIR fatorial * (argumento - 1) A fatorial
ATRIBUIR argumento - 1 A argumento
FIM
IMPRIMIR fatorial

Dois tokens
<=><=>

Erro léxico!

Padrão	Sigla
DECLARACOES, ALGORITMO, INTEIRO, REAL, ATRIBUIR, A, LER, IMPRIMIR, SE, ENTAO, ENQUANTO, INICIO, FIM	3 primeiras letras
*, /, +, -	OpArit
<, <=, >=, >, =, <>	OpRel
E, OU	OpBool
:	Delim
(,)	AP / FP
Sequências de letras ou números que começam com letra	Var
Sequências de dígitos (sem vírgula)	NumI
Sequências de dígitos (com vírgula)	NumR
Sequências de caracteres envolta por aspas	Str

DÚVIDAS?

Referências

Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman.
Compiladores - princípios, técnicas e ferramentas. Pearson, 2007.
José Neto, João. Apresentação à compilação. 2ª edição. Rio de Janeiro: Elsevier, 2016.
Notas de aula do professor Daniel Lucrédio - UFSCar.