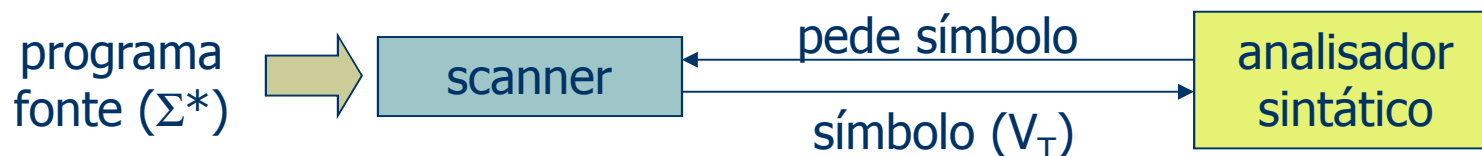


## 2.2 - Análise Léxica

- ♦ Seja  $P$  um programa escrito na linguagem de programação  $L \subseteq V_T^*$ .  
Exemplo: Em Pascal,  $V_T = \{ \text{begin, end, while, integer, :=, *, /, ...} \}$
- ♦ Entretanto, não existem disponíveis os símbolos de  $V_T$ . Existe apenas:  
 $\Sigma = \{ a, b, c, ..., 0, 1, ..., :, =, *, /, ... \}$
- ♦ O programa  $P$  que é fornecido ao compilador é, na realidade, uma sequência de  $\Sigma^*$ . O analisador léxico transforma essa sequência em uma sequência de  $V_T^*$ .



- ♦ O scanner pode ser programado como uma subrotina, chamada pelo analisador sintático sempre que este necessitar de um novo símbolo.



# Análise Léxica

- Sejam:  $\Sigma = \{ a, b, c, \dots, 0, 1, \dots, :, =, *, /, \dots \}$   
 $V_T = \{ <id>, <int>, /, +, -, *, //, (, ), BEGIN, END, ABS \}$

onde:

$<id> ::= <letra> |$   
 $\quad <id><letra> |$   
 $\quad <id><dígito>$   
 $<int> ::= <dígito> |$   
 $\quad <int><dígito>$   
 $<letra> ::= A | B | \dots | Z$   
 $<dígito> ::= 0 | 1 | \dots | 9$

Seja a seguinte representação  
para os símbolos de  $V_T$ :

O analisador léxico deverá retirar os  
comentários que são sequências de  $\Sigma^*$   
delimitadas por  $/*$  e  $*/$ . Os brancos  
entre os símbolos deverão ser ignorados  
também.

símbolo	nome	representação interna
indefinido	\$UND	0
identificador	\$ID	1
inteiro	\$INT	2
BEGIN	\$BEGIN	3
END	\$END	4
ABS	\$ABS	5
/	\$SLASH	6
+	\$PLUS	7
-	\$MINUS	8
*	\$STAR	9
(	\$LPAR	10
)	\$RPAR	11
//	\$SLSL	12

# Análise Léxica

---

- ♦ A saída do analisador léxico será da forma: (SIMB, INFO), onde SIMB é a representação interna e INFO é a informação semântica (caracteres componentes) de cada token.
- ♦ Exemplo: Se o texto:

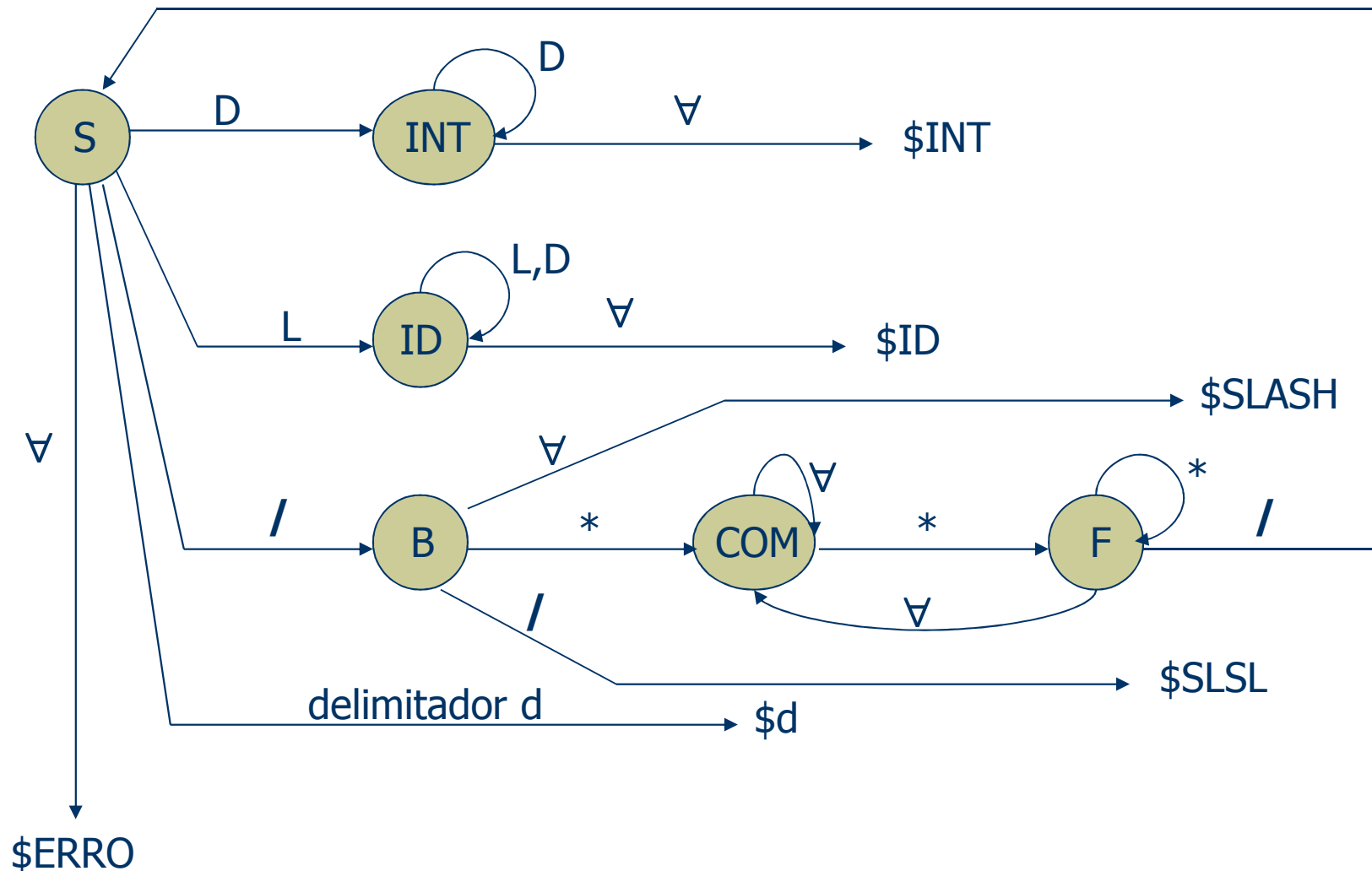
BEGIN A + /\* COMENTARIO \* + \*/ BC // END 11

for dado como entrada para o “scanner”, este deverá devolver:

	SIMB	INFO
1	3	'BEGIN'
2	1	'A'
3	7	'+'
4	1	'BC'
5	12	'//'
6	4	'END'
7	2	'11'

# Análise Léxica

- Vamos supor, inicialmente, que não existem as palavras reservadas (BEGIN, END e ABS, nesse caso). Podemos construir o seguinte autômato:



# Análise Léxica

---

- ◆ Problemas com esse scanner:
  - a) palavras reservadas não são reconhecidas
  - b) o caractere seguinte ao símbolo reconhecido pode ou não ter sido lido.Será necessário então:
  - a) tabelas de palavras reservadas e de delimitadores
  - b) convenção sobre o caractere seguinte ao símbolo reconhecido (por exemplo: ler sempre)
- ◆ Para construir um novo scanner, sejam as variáveis e procedimentos:

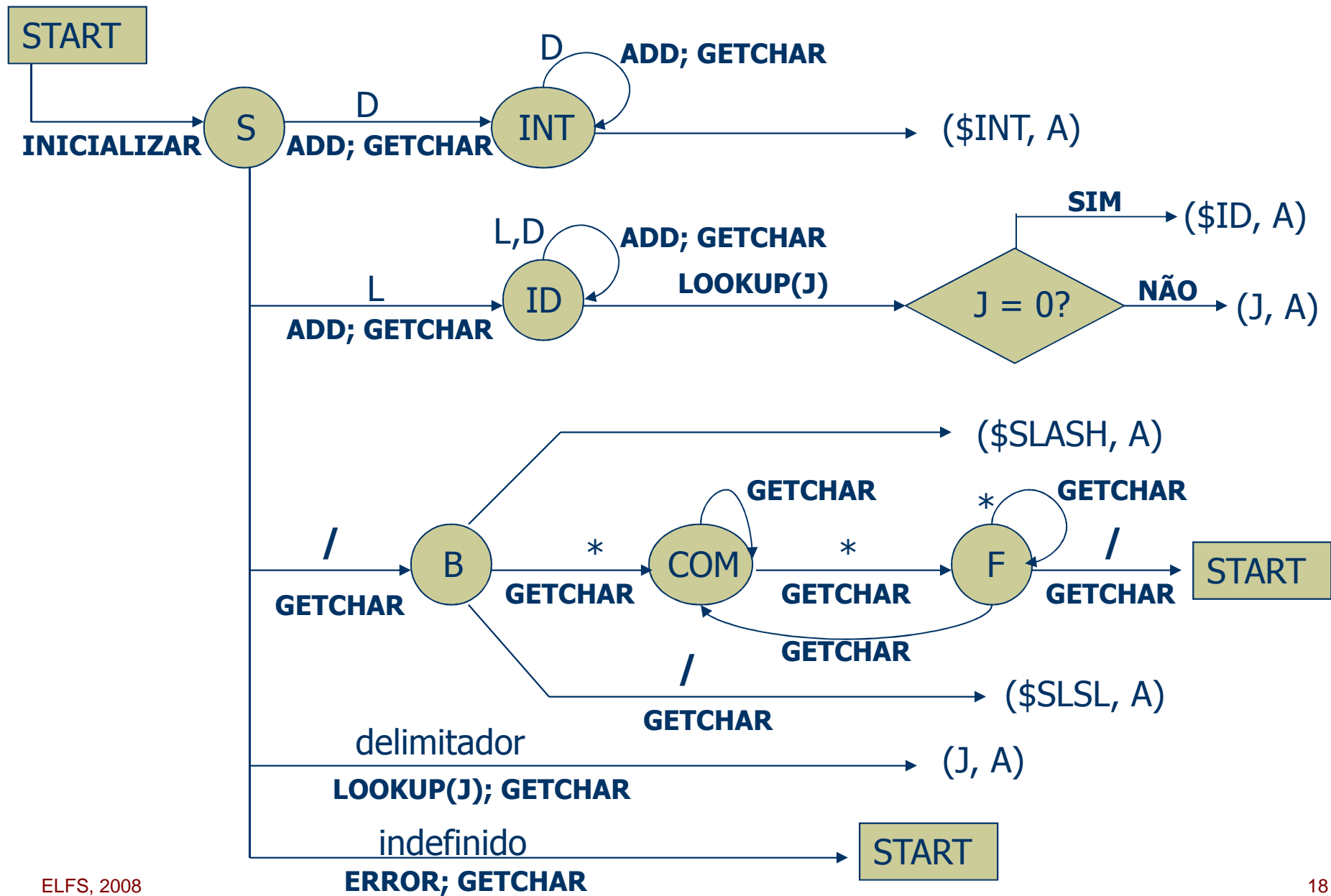
Tipo	Nome	Descrição
caractere	CHAR	Guarda o caractere sendo lido.
integer	CLASS	Contém a classe do caractere sendo lido (dígito = 1; letra = 2; / = 3; delimitador = 4).
string	A	Contém a sequência de caracteres que forma o símbolo.
procedure	GETCHAR	Percorre o texto, guarda o caractere em CHAR e a classe em CLASS; efetua operações de E/S quando necessário (ler e imprimir uma linha; mudar de linha; detectar fim de arquivo).

## Análise Léxica

---

Tipo	Nome	Descrição
procedure	GETNONBLANK	Obtém o próximo caractere diferente de " " por meio de chamadas sucessivas a GETCHAR.
procedure	LOOKUP	Procura numa tabela de palavras reservadas, retornando 0 (zero) se o identificador não está na tabela, ou a representação interna da palavra reservada. Os delimitadores simples podem ser armazenados nessa tabela.
procedure	ADD	Concatena CHAR ao string A.

# Análise Léxica



# Análise Léxica

---

- ♦ Observações:
  - INICIALIZAR é um procedimento que efetua as inicializações necessárias;
  - na entrada do scanner um caractere já deve ter sido lido (ou seja, GETCHAR deve ser chamado antes).
- ♦ Alguns pontos importantes:
  - 1) o scanner deve construir o maior símbolo possível.

Exemplo: ABC12 é um único identificador e não um identificador seguido de uma constante.
  - 2) existem alguns problemas:
    - a)  $\begin{array}{l} \text{DO10I}=1.15 \\ \text{DO10I}=1,15 \end{array} \quad \left. \vphantom{\begin{array}{l} \text{DO10I}=1.15 \\ \text{DO10I}=1,15 \end{array}} \right\} \text{ será preciso analisar o contexto para saber qual é a atribuição e qual é o "DO"}$
    - $\text{X} = \text{FOC}(3) \quad \left. \vphantom{\text{X} = \text{FOC}(3)} \right\} \text{ FOC é função ou vetor?}$
    - b) palavras reservadas que podem ser usadas como identificadores.