Manual do Usuário Compilador D+

Lucas Ferreira de Almeida

13/10/2021

Conteúdo

1 Sobre o compilador

1.1 Árvore de diretorios e Arquivos

```
   -- DS_PLUS_COMPILER
------- Entrada (Pasta aonde fica o arquivo .d para ser compilado)
------- Src (Classes da aplicação)
------------ File.cs (Classe para manusear arquivos)
------------ LEXICA.cs (Classe para realizar a analise Léxica)
------------ SINTATICO.cs (Classe para realizar a analise Sintática)
------------ Tokens.cs (Classe com os tokens)
------- doc
------------ DTs (Diagramas de transição)
----------------- Img (Diagramas de transição em imagem)
------------ Latex (Arquivo latex para gerar a documentação)
------------ mu.pdf (Manual do usuário)
------- Config.cs (Arquivo com as configurações globais da aplicação)
------- DS_PLUS_COMPILER.cs (Main)
```

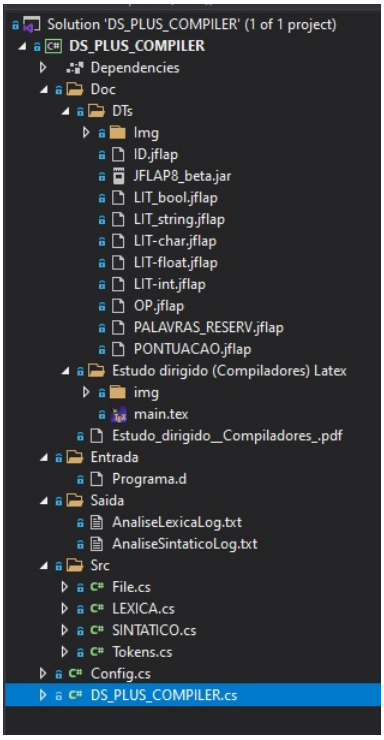

Figura 1: Árvore da aplicação.

1.2 Como utilizar

O arquivo de entrada deverá ser copiado para a pasta "Entrada", após isso é necessário rodar a aplicação. Os tokens gerados apartir do Programa.d serão printados no console e será gerados arquivos de logs na pasta Saida.

```
--------- (INICIO) PRINT LEXICO ---------------

          LEXEMA                                TOKEN

          void                          |       PR_VOID
          main                          |       PR_MAIN
          (                             |       ABRE_PARENTESES
          )                             |       FECHA_PARENTESES
          {                             |       ABRE_CHAVES
          string                        |       PR_STR
          _teste                        |       ID
          =                             |       OP_ATRI
          "teste de string"             |       LIT_STR
          ;                             |       PONTO_VIRGULA
          int                           |       PR_INT
          _inteiro                      |       ID
          =                             |       OP_ATRI
          45                            |       LIT_INT
          ;                             |       PONTO_VIRGULA
          float                         |       PR_FLT
          _real                         |       ID
          =                             |       OP_ATRI
          45.45                         |       LIT_FLT
          ;                             |       PONTO_VIRGULA
          int                           |       PR_INT
          _int2                         |       ID
          ;                             |       PONTO_VIRGULA
          _int2                         |       ID
          =                             |       OP_ATRI
          _inteiro                      |       ID
          +                             |       OP_SOMA
          _real                         |       ID
          ;                             |       PONTO_VIRGULA
          print                         |       PR_PRINT
          (                             |       ABRE_PARENTESES
          _int2                         |       ID
          )                             |       FECHA_PARENTESES
          ;                             |       PONTO_VIRGULA
          }                             |       FECHA_CHAVES
                                        |       FIM


------- (FIM) PRINT LEXICO ---------------
```

Figura 2: Print no console da analise Léxica.

2 Tokens

2.1 Identificadores

Identificador	ID

2.2 Literais

Inteiros	LIT_INT
Reais	LIT_FLT
Caractere	LIT_CHAR
String	LIT_STR
Boolean	LIT_BOOL

2.3 Palavras Reservadas

void	PR_VOID
int	PR_INT
float	PR_FLT
char	PR_CHAR
bool	PR_BOOL
if	PR_IF
then	PR_THEN
else	PR_ELSE
end-if	PR_ENDIF
for	PR_FOR
while	PR_WHILE
do	PR_DO
return	PR_RETURN
true	PR_TRUE
false	PR_FALSE
var	PR_VAR
main	PR_MAIN
scan	PR_SCAN
print	PR_PRINT

2.4 Pontuação

,	VIRGULA
;	PONTO_VIRGULA
(	ABRE_PARENTESES
)	FECHA_PARENTESES
[	ABRE_COLCHETES
]	FECHA_COLCHETES
{	ABRE_CHAVES
}	FECHA_CHAVES

2.5 Operadores

+	OP_SOMA
-	OP_SUB
*	OP_MULT
/	OP_DIV
%	OP_MOD
?	OP_TER
!	OP_NEG
.	OP_PONTO
<	OP_MENOR
>	OP_MAIOR
==	OP_IGUAL
!=	OP_DIF
<=	OP_MEN_IGUAL
>=	OP_MAI_IGUAL
=	OP_ATRI
+=	OP_ADIC_IGUAL
-=	OP_SUB_IGUAL
++	OP_INC
- -	OP_DEC
&&	OP_AND
||	OP_OR

3 Diagramas de transição

3.1 Identificadores

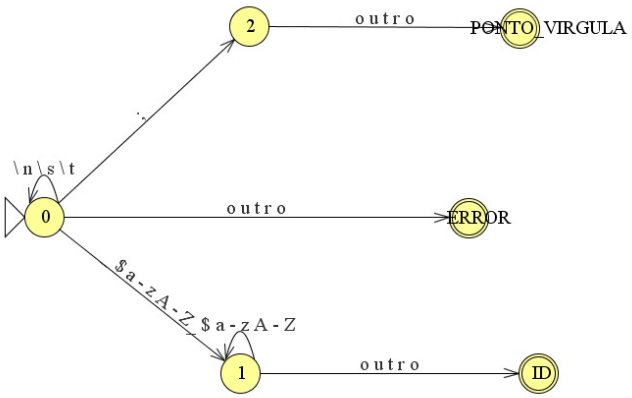

Figura 3: Indetificadores.

3.2 Literais

Figura 4: Literais (String).

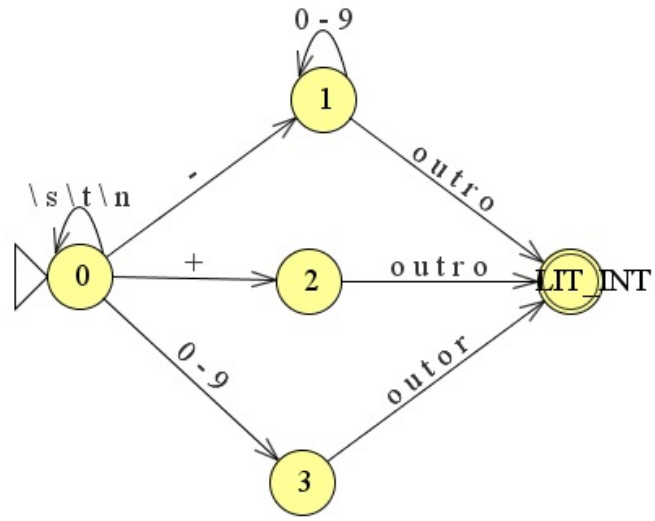

Figura 5: Literais (Inteiro).

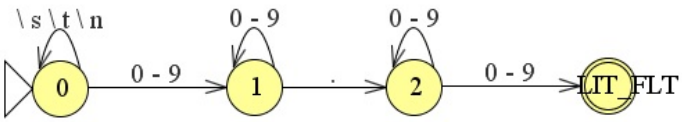

Figura 6: Literais (Reais).

Figura 7: Literais (Char).

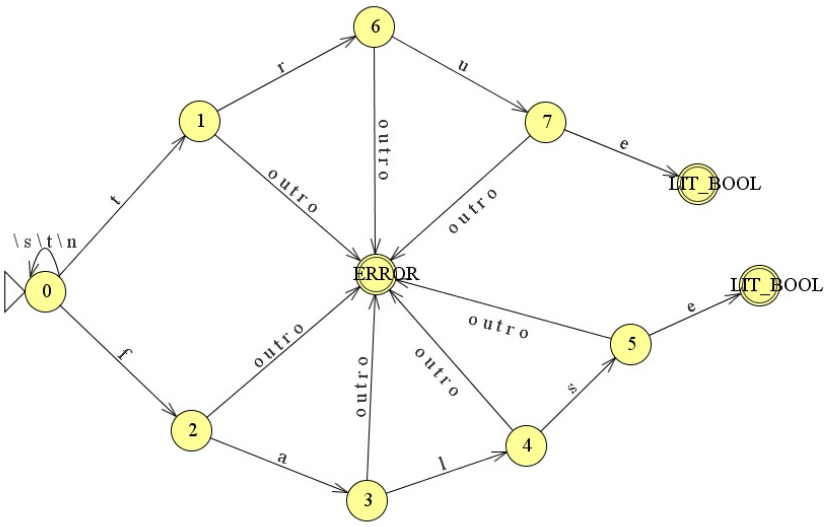

Figura 8: Literais (Boolean).

3.3 Pontuação

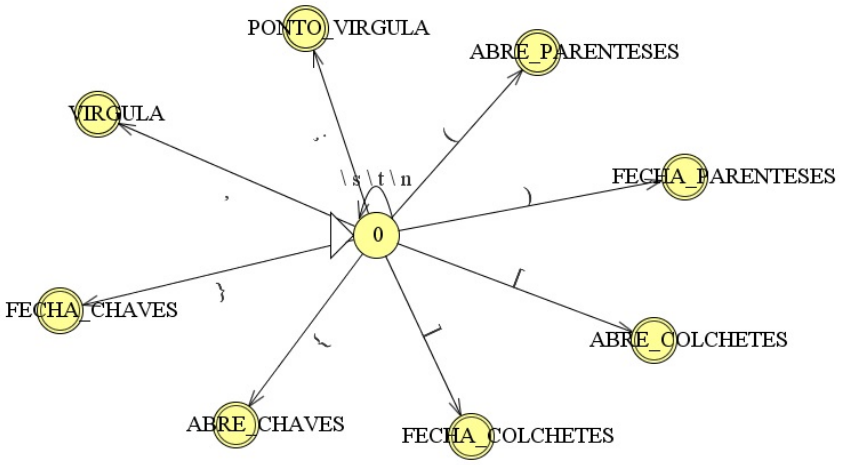

Figura 9: Literais (Boolean).

3.4 Palavras Reservadas

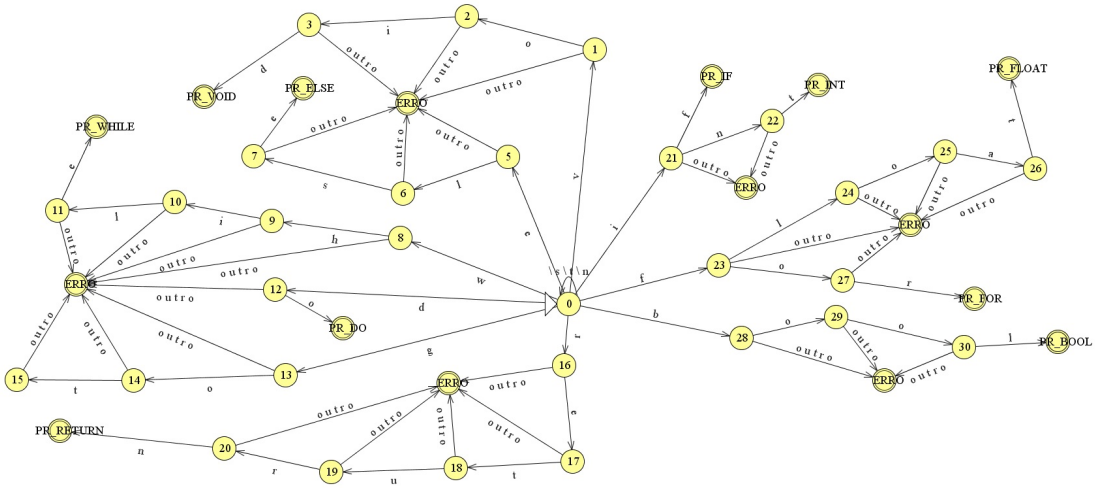

Figura 10: Literais (Boolean).

3.5 Operadores

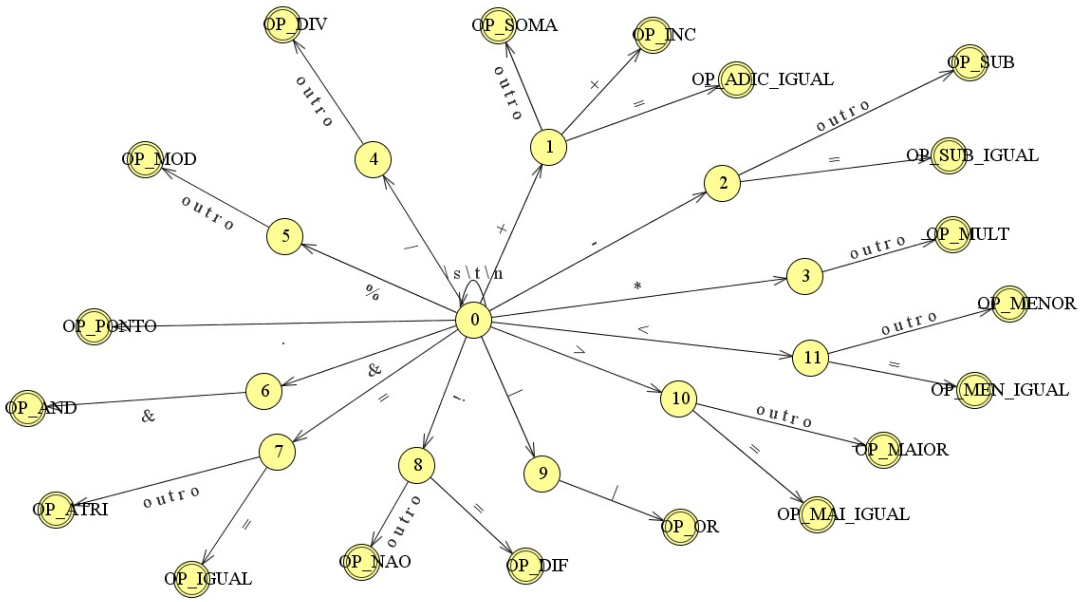

Figura 11: Literais (Boolean).

4 Validações Lexica e Sintática

4.1 Lexica

Tipo	Mensagem
Erro	`'Erro': Operador ou pontuação inválida. Lexema = 'Lexema'`
Erro	`'Erro': Char inválido. Lexema = 'Lexema'`
Erro	`'Erro': Comando não identificado. Lexema = 'Lexema'`

4.2 Sintática