

CES41: Série1

Gianluigi Dal Toso
Divisão de Ciência da Computação
Instituto Tecnológico de Aeronáutica
São José dos Campos, Brasil
gianluigi.toso@gmail.com

I. GRAMÁTICA

Realizando a expansão da gramática fornecida pelo enunciado, conclui-se que um programa consiste basicamente de uma linguagem de atribuição. Um programa escrito nessa linguagem terá essencialmente a estrutura apresentada pelo Código 1.

Código 1: Exemplo de código válido pela expansão da gramática fornecida.

```
1  programName {  
2      toso = 25;  
3      turma = 21;  
4      anoInicio = 1965;  
5      respostaUniverso = 42;  
6  
7      var1 = ( anoInicio + respostaUniverso ) + ( 256 + toso );  
8      var2 = var1 + ( turma ) + ( anoInicio + ( var1 + toso ) );  
9  }
```

Este código será o arquivo de dados que será utilizado para os testes das análises que seguirão.

II. QUESTÃO 1: ANÁLISE LÉXICA

Antes de iniciar a fase de implementação do analisador léxico, construiu-se os diagramas de transição referentes à etapa de análise léxica para auxiliar a implementação. A Figura 4 ilustra o diagrama de transição léxico completo para essa gramática. e as Figuras 1, 2, e 3 são recortes didáticos do diagrama completo para melhor compreensão.

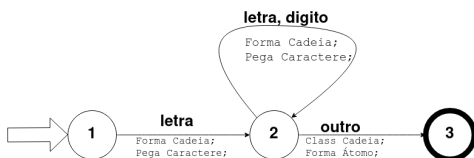


Figura 1: Reconhecimento de cadeias do tipo ID.

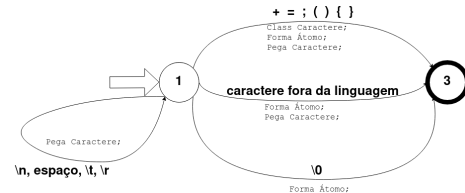


Figura 2: Reconhecimento de caracteres únicos.

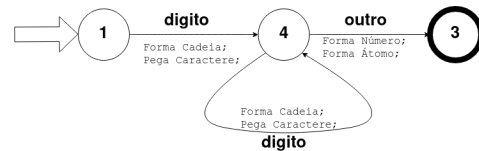


Figura 3: Reconhecimento de cadeias do tipo CTE.

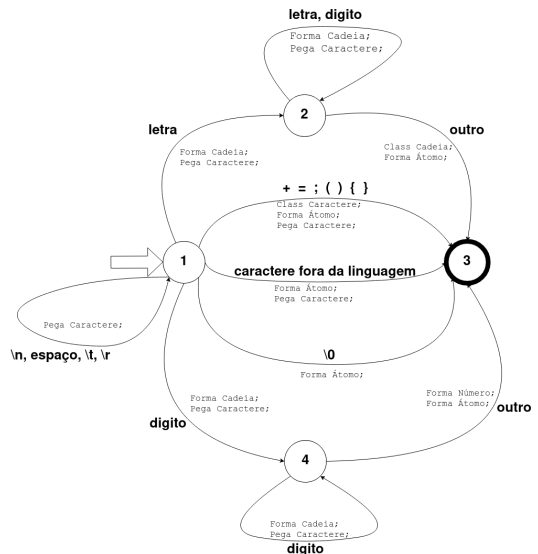


Figura 4: Diagrama completo com todas as transições

A implementação do analisador léxico foi realizada em linguagem C, seguindo a mesma ideia apresentada pelos slides de aula (Capítulo 3). O número máximo de caracteres para uma cadeia foi definido como sendo de 30 (trinta) caracteres e definido pela diretiva de pré-processamento MAXCADEIA. Foi definido um número pequeno apenas para tornar a tabela de exibição dos átomos mais elegante e didática e tal limite pode ser facilmente modificado.

O código completo está anexo à este relatório, em formato comprimido (.zip), juntamente com os arquivos de teste utilizados. O código principal do analisador léxico está contido no arquivo `analisadorLex.c`.

Como esta fase trata apenas de identificar os átomos que compõe a estrutura do programa e sua estrutura ainda não é avaliada, será utilizado como entrada de testes uma modificação do código exemplo para a gramática apresentado anteriormente (Código 1) adicionando à este código alguns caracteres inválidos e também fragmentos lexicamente corretos apesar de sintaticamente inválidos. O conteúdo do arquivo de entrada de dados com esse código modificado é apresentado pelo Código 2.

Código 2: Arquivo `entradaLex.dat` utilizado para testar o analisados léxico.

```

1  programName {
2      toso = 25;
3      turma = 21;
4      anoInicio = 1965;
5      respostaUniverso = 42;
6
7      var1 = ( anoInicio + respostaUniverso ) + (256 + toso);
8      var2 = var1 + (turma) + (anoInicio + (var1 + toso));
9
10     ? codigo valido lexicamente    !
11     & mas invalido sintaticamente %
12     # 1323123 3123123 3123123 31 $
13     1994 + 25 = invalidoToso
14     ...
15 }

```

O Código 3 e 4 apresentam as saídas provenientes da execução do código implementado para o analisador léxico comprovando seu funcionamento. A execução foi realizada em um ambiente *Linux* rodando em uma arquitetura *amd64*.

Código 3: *Output* do analisador léxico no `stdout`.

```

1  === A N A L I S E   L E X I C A ===
2
3  Nome do arquivo: entradaLex.dat
4
5  Analise do arquivo 'entradaLex.dat' encerrada
6
7  Ver atomos no arquivo 'atomosmp'

```

Código 4: Arquivo `atomosmp`, saída do analisador léxico.

```

Arquivo de entrada: entradaLex.dat
+-----+
| #ID | ATOMO   | VALOR                                |
+-----+
| 0   | ID      | programName                         |

```

```

+-----+
| 4   | ACHAV   | {                                    |
+-----+
| 0   | ID      | toso                                |
+-----+
| 3   | ATRIB   | =                                    |
+-----+
| 1   | CTE     | 25                                  |
+-----+
| 8   | PVIRG   | ;                                    |
+-----+
| 0   | ID      | turma                              |
+-----+
| 3   | ATRIB   | =                                    |
+-----+
| 1   | CTE     | 21                                  |
+-----+
| 8   | PVIRG   | ;                                    |
+-----+
| 0   | ID      | anoInicio                          |
+-----+
| 3   | ATRIB   | =                                    |
+-----+
| 1   | CTE     | 1965                               |
+-----+
| 8   | PVIRG   | ;                                    |
+-----+
| 0   | ID      | respostaUniverso                   |
+-----+
| 3   | ATRIB   | =                                    |
+-----+
| 1   | CTE     | 42                                  |
+-----+
| 8   | PVIRG   | ;                                    |
+-----+
| 0   | ID      | var1                               |
+-----+
| 3   | ATRIB   | =                                    |
+-----+
| 6   | APAR    | (                                    |
+-----+
| 0   | ID      | anoInicio                          |
+-----+
| 2   | OPAD    | +                                    |
+-----+
| 0   | ID      | respostaUniverso                   |
+-----+
| 7   | FPAR    | )                                    |
+-----+
| 2   | OPAD    | +                                    |
+-----+
| 6   | APAR    | (                                    |
+-----+
| 1   | CTE     | 256                                  |
+-----+
| 2   | OPAD    | +                                    |
+-----+
| 0   | ID      | toso                                |
+-----+
| 7   | FPAR    | )                                    |
+-----+

```

8	PVIRG	;	
+-----+			
0	ID	var2	
+-----+			
3	ATRIB	=	
+-----+			
0	ID	var1	
+-----+			
2	OPAD	+	
+-----+			
6	APAR	(
+-----+			
0	ID	turma	
+-----+			
7	FPAR)	
+-----+			
2	OPAD	+	
+-----+			
6	APAR	(
+-----+			
0	ID	anoInicio	
+-----+			
2	OPAD	+	
+-----+			
6	APAR	(
+-----+			
0	ID	var1	
+-----+			
2	OPAD	+	
+-----+			
0	ID	toso	
+-----+			
7	FPAR)	
+-----+			
7	FPAR)	
+-----+			
8	PVIRG	;	
+-----+			
10	INVAL	?	
+-----+			
0	ID	codigo	
+-----+			
0	ID	valido	
+-----+			
0	ID	lexicamente	
+-----+			
10	INVAL	!	
+-----+			
10	INVAL	&	
+-----+			
0	ID	mas	
+-----+			
0	ID	invalido	
+-----+			
0	ID	sintaticamente	
+-----+			
10	INVAL	%	
+-----+			
10	INVAL	#	
+-----+			
1	CTE	1323123	

+-----+			
1	CTE	3123123	
+-----+			
1	CTE	3123123	
+-----+			
1	CTE	31	
+-----+			
10	INVAL	\$	
+-----+			
1	CTE	1994	
+-----+			
2	OPAD	+	
+-----+			
1	CTE	25	
+-----+			
3	ATRIB	=	
+-----+			
0	ID	invalidoToso	
+-----+			
10	INVAL	.	
+-----+			
10	INVAL	.	
+-----+			
10	INVAL	.	
+-----+			
5	FCHAV	}	
+-----+			
9	FINAL		
+-----+			