



INSTITUTO TECNOLÓGICO DE AERONÁUTICA

CES-41: COMPILADORES

Laboratório II:

Construção de um analisador léxico para uma linguagem de programação usando a ferramenta Flex.

Gianluigi Dal Toso

Professor:
Fábio Carneiro Mokarzel

25 de Maio de 2020

Sobre os códigos e arquivos de dados

Anexos à este relatório está um arquivo comprimido em formato `.zip` contendo todos os códigos-fonte desenvolvidos para este laboratório, bem como o arquivo de dados utilizado para testar a implementação de cada tarefa.

Tarefa

- Usando a ferramenta Flex, implementar, um analisador léxico para a linguagem de programação COMP-ITA 2020, apresentada no documento Linguagem COMP-ITA 2020 - CES-41/2020. Tal analisador será uma ferramenta para um analisador sintático da mesma linguagem, a ser implementado num dos próximos laboratórios, usando a ferramenta Yacc.
- Gerar o código C e o código executável desse analisador.

A resolução deste laboratório foi feita tomando-se como base os exemplos vistos nos arquivos das aulas e também no arquivo `RealCharString.1` fornecido pela documentação do laboratório. No entanto, realizei algumas alterações para deixar a saída ainda mais detalhada e formatada como uma tabela. O código poderia ter sido simplificado para mostrar apenas o número referente ao ID do átomo, dessa forma diversos casos poderiam ser agrupados no código de impressão e o laboratório ficaria mais curto. No entanto, gostei da ideia de fazer uma apresentação mais detalhada dos átomos, mesmo que isso tivesse um maior custo de implementação e deixasse o código mais “*verbose*”.

Uma modificação que foi realizada quando se comparando os arquivos implementados com os arquivos fornecidos nas aulas, foi a troca das diretivas de pré-processamento para definir o valor do tipo dos atributos por um `enum`. Como o laço de repetição principal do programa depende do valor de `yyllex()`, que recebe os retornos das expressões regulares, o `enum` utilizado deve iniciar com um valor superior à zero. O Código 1 contém a seção de definições no analisador léxico implementado utilizando `enum`.

Código 1: Seção de definições do analisador léxico implementado.

```
#include <string.h>

enum tipos {
    /* Palavras reservadas */
    CARAC=1, CHAMAR, COMANDOS, ENQUANTO, ESCREVER, FALSO, FUNCAO, INT, LER, LOGIC, PARA, PRINCIPAL,
    PROCEDIMENTO, PROGRAMA, REAL, REPETIR, RETORNAR, SE, SENAO, VAR, VERDADE,
    /* Constantes */
    ID, CTINT, CTCARAC, CTREAL, CADEIA,
    /* Operadores */
    ATRIB, OR, AND, NOT, NEG, OPAD, SOMA, SUB, OPMULT, MULT, DIV, MOD, OPREL, EQ, NEQ, LT, LEQ, GT, GEQ,
    /* Controle */
    ABPAR, FPAR, ABCOL, FCOL, ABCHAV, FCHAV, ABTRIP, FTRIP, PVIG, VIRG, INVAL, FINAL
};

union {
    char string[50];
    int atr, valor;
    float valreal;
    char carac;
} yylval;
```

O Código 2 contém as expressões regulares utilizadas na implementação do analisador léxico. No demais, o código segue o padrão utilizado nas atividades do primeiro laboratório, com exceção da parte de impressão, que modifiquei para imprimir os átomos em uma tabela com informações mais detalhas. Por ser extenso demais, o código completo não será inserido neste relatório, mas pode ser consultado no arquivo `main.c` dos itens em anexo.

Código 2: Expressões regulares utilizadas pelo analisador léxico implementado.

delim	[\t\n\r]
ws	{delim}+
digito	[0-9]
letra	[A-Za-z]
ctint	{digito}+
carac1	\\. \\['\\\"\\n]
ctcarac	'{carac1}'
carac2	\\. \\["\\'\\\"\\n]
cadeia	\"{carac2}*\"
ctreal	{digito}+\\. {digito}*([Ee][+-]?{digito}+)?
caractere	{letra} {digito}
id	{letra}{caractere}*
comentario	\\/*(\\.\\n*\\.)**\\/

Arquivos de teste e resultado

Para testar a execução do código, foram utilizados três arquivos de teste. O primeiro arquivo de testes, nomeado `teste.comp` (estarei utilizando a extensão `.comp` para os arquivos da linguagem COMP-ITA) consiste de um arquivo contendo todos os possíveis átomos da linguagem e comentários, porém não representa um código sintaticamente válido. O segundo arquivo de teste, nomeado `reduzido.comp` consiste de um código curto e válido e o seu conteúdo está apresentado no Código 4. Já o terceiro arquivo de teste, nomeado `completo.comp` é o código fonte do programa apresentado no documento fornecido como manual para a linguagem COMP-ITA.

Código 3: Arquivo de entrada `teste.comp`.

```
/* ignore comment */
carac chamar comandos enquanto escrever falso
funcao int ler logic para principal
procedimento programa real repetir retornar se
senao var verdade
/* ignore comment */
toso nome123 var1
1234 21 1911
'\n' 'T' '2' '1' '\.'
3.1415 2.7E-3 1.6E4
"string1" "confortably numb" "Mokarzel"
|| && ! < <= > >= == !=
+ -
* / %
/* ignore other comment */
~
= ( ) [ ] { } {{ } } ; ,
!teste ~teste2

/* some fails and tricks */
.45 # $ @ & _
/* multiline comment here
that works */
/* this /* is a tricky /* comment */
```

Código 4: Arquivo de entrada `reduzido.comp`.

```

/* Programa para demonstrar a funcionalidade
do analisador léxico implementado */
programa DemontraLab {{{
var {
    int (turma, idade)
    carac (nome[20])
    carac (inicial)
}

funcao int CalculaMedia ()
var {
    int (P1, P2, final)
}
comandos {
    ler (P1);
    ler (P2);
    final = (P1 + P2) / 2;
    retornar final;
}

principal
var {
    logic (aprovado)
    carac (c)
}
comandos {
    nome = "Toso"; idade = 25; turma = 21; inicial = 'G';
    aprovado = falso;

    escrever ("Calcular a media? (s/n): ");
    ler (c);
    se (c == 's' || c == 'S') {
        se (CalculaMedia() >= 6.5)
            aprovado = verdade;
    }
}
}}}

```

Para fins didáticos, no relatório estarei mostrando somente as saídas do arquivo contendo todos os átomos (**teste.comp**) e do programa reduzido (**reduzido.comp**), que podem ser conferidas nos Códigos 5 e 6 respectivamente. No entanto, a saída da execução para todos os arquivos de teste pode ser conferida no conteúdo dos arquivos **teste.dat**, **reduzido.dat** e **completo.dat** anexos à este relatório.

Código 5: Saída obtida pela execução do analisador léxico sobre o arquivo **teste.comp**.

TEXT0	ID: ATOMO	ATRIBUTO
carac	1: CARAC	
chamar	2: CHAMAR	
comandos	3: COMANDOS	
enquanto	4: ENQUANTO	
escrever	5: ESCREVER	

falso	6: FALSO	
+-----+	+-----+	+-----+
funcao	7: FUNCAO	
+-----+	+-----+	+-----+
int	8: INT	
+-----+	+-----+	+-----+
ler	9: LER	
+-----+	+-----+	+-----+
logic	10: LOGIC	
+-----+	+-----+	+-----+
para	11: PARA	
+-----+	+-----+	+-----+
principal	12: PRINCIPAL	
+-----+	+-----+	+-----+
procedimento	13: PROCEDIMENTO	
+-----+	+-----+	+-----+
programa	14: PROGRAMA	
+-----+	+-----+	+-----+
real	15: REAL	
+-----+	+-----+	+-----+
repetir	16: REPETIR	
+-----+	+-----+	+-----+
retornar	17: RETORNAR	
+-----+	+-----+	+-----+
se	18: SE	
+-----+	+-----+	+-----+
senao	19: SENAO	
+-----+	+-----+	+-----+
var	20: VAR	
+-----+	+-----+	+-----+
verdade	21: VERDADE	
+-----+	+-----+	+-----+
toso	22: ID	toso
+-----+	+-----+	+-----+
nome123	22: ID	nome123
+-----+	+-----+	+-----+
var1	22: ID	var1
+-----+	+-----+	+-----+
1234	23: CTINT	1234
+-----+	+-----+	+-----+
21	23: CTINT	21
+-----+	+-----+	+-----+
1911	23: CTINT	1911
+-----+	+-----+	+-----+
'\n'	24: CTCARAC	'\n'
+-----+	+-----+	+-----+
'T'	24: CTCARAC	'T'
+-----+	+-----+	+-----+
'2'	24: CTCARAC	'2'
+-----+	+-----+	+-----+
'1'	24: CTCARAC	'1'
+-----+	+-----+	+-----+
'\.'	24: CTCARAC	'\.'
+-----+	+-----+	+-----+
3.1415	25: CTREAL	3.141500
+-----+	+-----+	+-----+
2.7E-3	25: CTREAL	0.002700
+-----+	+-----+	+-----+
1.6E4	25: CTREAL	16000.000000
+-----+	+-----+	+-----+
"string1"	26: CADEIA	"string1"

"confortably numb"	26: CADEIA	"confortably numb"	
"Mokarzel"	26: CADEIA	"Mokarzel"	
	28: OR		
&&	29: AND		
!	30: NOT		
<	39: OPREL	42: LT	
<=	39: OPREL	43: LEQ	
>	39: OPREL	44: GT	
>=	39: OPREL	45: GEQ	
==	39: OPREL	40: EQ	
!=	39: OPREL	41: NEQ	
+	32: OPAD	33: SOMA	
-	32: OPAD	34: SUB	
*	35: OPMULT	36: MULT	
/	35: OPMULT	37: DIV	
%	35: OPMULT	38: MOD	
~	31: NEG		
=	27: ATRIB		
(46: ABPAR		
)	47: FPAR		
[48: ABCOL		
]	49: FCOL		
{	50: ABCHAV		
}	51: FCHAV		
{{{	52: ABTRIP		
}}}	53: FTRIP		
;	54: PVIG		
,	55: VIRG		
!	30: NOT		
teste	22: ID	teste	

~	31: NEG	
+-----+	+-----+	+-----+
teste2	22: ID	teste2
+-----+	+-----+	+-----+
.	56: INVAL	
+-----+	+-----+	+-----+
45	23: CTINT	45
+-----+	+-----+	+-----+
#	56: INVAL	
+-----+	+-----+	+-----+
\$	56: INVAL	
+-----+	+-----+	+-----+
@	56: INVAL	
+-----+	+-----+	+-----+
&	56: INVAL	
+-----+	+-----+	+-----+
_	56: INVAL	
+-----+	+-----+	+-----+

Código 6: Saída obtida pela execução do analisador léxico sobre o arquivo `reduzido.comp`.

+-----+	+-----+	+-----+
TEXTO	ID: ATOMO	ATRIBUTO
+-----+	+-----+	+-----+
programa	14: PROGRAMA	
+-----+	+-----+	+-----+
DemontraLab	22: ID	DemontraLab
+-----+	+-----+	+-----+
{{{	52: ABTRIP	
+-----+	+-----+	+-----+
var	20: VAR	
+-----+	+-----+	+-----+
{	50: ABCHAV	
+-----+	+-----+	+-----+
int	8: INT	
+-----+	+-----+	+-----+
(46: ABPAR	
+-----+	+-----+	+-----+
turma	22: ID	turma
+-----+	+-----+	+-----+
,	55: VIRG	
+-----+	+-----+	+-----+
idade	22: ID	idade
+-----+	+-----+	+-----+
)	47: FPAR	
+-----+	+-----+	+-----+
carac	1: CARAC	
+-----+	+-----+	+-----+
(46: ABPAR	
+-----+	+-----+	+-----+
nome	22: ID	nome
+-----+	+-----+	+-----+
[48: ABCOL	
+-----+	+-----+	+-----+
20	23: CTINT	20
+-----+	+-----+	+-----+
]	49: FCOL	
+-----+	+-----+	+-----+

)	47: FPAR	
+-----+	+-----+	+-----+
carac	1: CARAC	
+-----+	+-----+	+-----+
(46: ABPAR	
+-----+	+-----+	+-----+
inicial	22: ID	inicial
+-----+	+-----+	+-----+
)	47: FPAR	
+-----+	+-----+	+-----+
}	51: FCHAV	
+-----+	+-----+	+-----+
funcao	7: FUNCAO	
+-----+	+-----+	+-----+
int	8: INT	
+-----+	+-----+	+-----+
CalculaMedia	22: ID	CalculaMedia
+-----+	+-----+	+-----+
(46: ABPAR	
+-----+	+-----+	+-----+
)	47: FPAR	
+-----+	+-----+	+-----+
var	20: VAR	
+-----+	+-----+	+-----+
{	50: ABCHAV	
+-----+	+-----+	+-----+
int	8: INT	
+-----+	+-----+	+-----+
(46: ABPAR	
+-----+	+-----+	+-----+
P1	22: ID	P1
+-----+	+-----+	+-----+
,	55: VIRG	
+-----+	+-----+	+-----+
P2	22: ID	P2
+-----+	+-----+	+-----+
,	55: VIRG	
+-----+	+-----+	+-----+
final	22: ID	final
+-----+	+-----+	+-----+
)	47: FPAR	
+-----+	+-----+	+-----+
}	51: FCHAV	
+-----+	+-----+	+-----+
comandos	3: COMANDOS	
+-----+	+-----+	+-----+
{	50: ABCHAV	
+-----+	+-----+	+-----+
ler	9: LER	
+-----+	+-----+	+-----+
(46: ABPAR	
+-----+	+-----+	+-----+
P1	22: ID	P1
+-----+	+-----+	+-----+
)	47: FPAR	
+-----+	+-----+	+-----+
;	54: PVIG	
+-----+	+-----+	+-----+
ler	9: LER	
+-----+	+-----+	+-----+
(46: ABPAR	

P2	22: ID	P2	
)	47: FPAR		
;	54: PVIG		
final	22: ID	final	
=	27: ATRIB		
(46: ABPAR		
P1	22: ID	P1	
+	32: OPAD	33: SOMA	
P2	22: ID	P2	
)	47: FPAR		
/	35: OPMULT	37: DIV	
2	23: CTINT	2	
;	54: PVIG		
retornar	17: RETORNAR		
final	22: ID	final	
;	54: PVIG		
}	51: FCHAV		
principal	12: PRINCIPAL		
var	20: VAR		
{	50: ABCHAV		
logic	10: LOGIC		
(46: ABPAR		
aprovado	22: ID	aprovado	
)	47: FPAR		
carac	1: CARAC		
(46: ABPAR		
c	22: ID	c	
)	47: FPAR		
}	51: FCHAV		
comandos	3: COMANDOS		

{	50: ABCHAV	
+-----+	+-----+	+-----+
nome	22: ID	nome
+-----+	+-----+	+-----+
=	27: ATRIB	
+-----+	+-----+	+-----+
"Toso"	26: CADEIA	"Toso"
+-----+	+-----+	+-----+
;	54: PVIG	
+-----+	+-----+	+-----+
idade	22: ID	idade
+-----+	+-----+	+-----+
=	27: ATRIB	
+-----+	+-----+	+-----+
25	23: CTINT	25
+-----+	+-----+	+-----+
;	54: PVIG	
+-----+	+-----+	+-----+
turma	22: ID	turma
+-----+	+-----+	+-----+
=	27: ATRIB	
+-----+	+-----+	+-----+
21	23: CTINT	21
+-----+	+-----+	+-----+
;	54: PVIG	
+-----+	+-----+	+-----+
inicial	22: ID	inicial
+-----+	+-----+	+-----+
=	27: ATRIB	
+-----+	+-----+	+-----+
'G'	24: CTCARAC	'G'
+-----+	+-----+	+-----+
;	54: PVIG	
+-----+	+-----+	+-----+
aprovado	22: ID	aprovado
+-----+	+-----+	+-----+
=	27: ATRIB	
+-----+	+-----+	+-----+
falso	6: FALSO	
+-----+	+-----+	+-----+
;	54: PVIG	
+-----+	+-----+	+-----+
escrever	5: ESCREVER	
+-----+	+-----+	+-----+
(46: ABPAR	
+-----+	+-----+	+-----+
"Calcular a media? (s/n): "	26: CADEIA	"Calcular a media? (s/n): "
+-----+	+-----+	+-----+
)	47: FPAR	
+-----+	+-----+	+-----+
;	54: PVIG	
+-----+	+-----+	+-----+
ler	9: LER	
+-----+	+-----+	+-----+
(46: ABPAR	
+-----+	+-----+	+-----+
c	22: ID	c
+-----+	+-----+	+-----+
)	47: FPAR	
+-----+	+-----+	+-----+
;	54: PVIG	

se	18: SE	
(46: ABPAR	
c	22: ID	c
==	39: OPREL	40: EQ
's'	24: CTCARAC	's'
	28: OR	
c	22: ID	c
==	39: OPREL	40: EQ
'S'	24: CTCARAC	'S'
)	47: FPAR	
{	50: ABCHAV	
se	18: SE	
(46: ABPAR	
CalculaMedia	22: ID	CalculaMedia
(46: ABPAR	
)	47: FPAR	
>=	39: OPREL	45: GEQ
6.5	25: CTREAL	6.500000
)	47: FPAR	
aprovado	22: ID	aprovado
=	27: ATRIB	
verdade	21: VERDADE	
;	54: PVIG	
}	51: FCHAV	
}	51: FCHAV	
}}}	53: FTRIP	