

# lab1 实验报告

PB20111630 张艺耀

## 实验要求

- 本次实验需要完成一个完整的 Cminus-f 解析器，包括基于 `flex` 的词法分析器和基于 `bison` 的语法分析器。
- 根据 `Cminus-f` 的词法补全 `src/parser/lexical_analyzer.l` 文件，完成词法分析器。
- 完成 `src/parser/syntax_analyzer.y` 语法分析器。

## 实验难点

### 词法分析器

个人认为最难的部分是词法分析器中浮点数和关于无嵌套的多行注释的正则表达式的书写。

关于无嵌套的多行注释的正则表达式，我一开始的写法是：

```
"\\\\"*"(("[\\\\"]*[\\\\"])*(("[\\\\"]*)([\\\\"]*\\\\"([\\\\"]*)*)*"\\\\"\\\\"
```

它指的是在前后两个注释符号之间不存在与之相同的符号，这样写较为冗余

之后改为（参考自StackOverflow）：

```
\\\\"*("[^"]|(\\"+[^"]*))\\"*\\\\"
```

其正确性如下：

```
1 The full explanation and derivation of that regex is excellently elaborated upon here.
2 In short:
3 "/" marks the start of the comment
4 ( [^"] | (\\"+[^"]*) )* says accept all characters that are not * (the [^"] ) or accept a
sequence of one or more * as long as the sequence does not have a ' ' or a / following it
(the (*+[^"])). This means that all *****... sequences will be accepted except for *****/
since you can't find a sequence of * there that isn't followed by a * or a /.
5 The *****/ case is then handled by the last bit of the RegEx which matches any number of *
followed by a / to mark the end of the comment i.e \"+\\"
```

以及在flex模式+动作的书写过程中，先后顺序很重要。

比如匹配一个特定的字符串的正则表达式要尽量写在前面，否则会出现 `warning: rule can not be matched` 的情况，这是指我们书写的这些表达式会被之前的正则表达式匹配到。

又如如下三行代码，如果把53行与55行顺序调换，则会先匹配到整数再匹配浮点数，这样会多出来一个句子。

## 语法分析器

词法分析器的写法较为简单，只需要无脑套公式就行了。

注意参数个数问题防止漏参数（实验过程中由于漏了param的第四个参数导致bug）

## 实验设计

根据实验文档补全 `lexical_analyzer.l` 和 `syntax_analyzer.y` 其中需要补全正则表达式的书写，编写 token和type以及对应规则，补全union中的node定义。

最后通过给出的测试脚本分别测试easy normal hard的测试样例。

## 实验结果验证

```
root@bbd8c5a5521e:/labs/2022fall-compiler_cminus/tests/parser# ./test_syntax.sh hard yes
[info] Analyzing assoc.cminus
[info] Analyzing gcd.cminus
[info] Analyzing hanoi.cminus
[info] Analyzing if.cminus
[info] Analyzing selectionsort.cminus
[info] Analyzing You_Should_Pass.cminus
[info] Comparing...
[info] No difference! Congratulations!
[info] Analyzing FAIL_local-decl.cminus
error at line 4 column 5: syntax error
[info] Analyzing func.cminus
[info] Analyzing if.cminus
[info] Analyzing local-decl.cminus
[info] Analyzing skip_spaces.cminus
[info] Comparing...
[info] No difference! Congratulations!
root@bbd8c5a5521e:/labs/2022fall-compiler_cminus/tests/parser#
```

自行设计的测试：

```
1  /* I am NO.1 */
2  int main(void) {
3      int a;
4      int b;
5      float c;
6      a = 1;
7      b = 2;
8      c = 1.1;
```

```
9
10     a = a + b;
11
12     return a;
13
14 }
```

lexer输出如下:

1	Token	Text	Line	Column (Start,End)
2	262	int	1	(1,4)
3	285	main	1	(5,9)
4	279	(	1	(9,10)
5	264	void	1	(10,14)
6	280	)	1	(14,15)
7	283	{	1	(16,17)
8	262	int	2	(3,6)
9	285	a	2	(7,8)
10	277	;	2	(8,9)
11	262	int	3	(3,6)
12	285	b	3	(7,8)
13	277	;	3	(8,9)
14	266	float	4	(3,8)
15	285	c	4	(9,10)
16	277	;	4	(10,11)
17	285	a	5	(3,4)
18	276	=	5	(5,6)
19	286	1	5	(7,8)
20	277	;	5	(8,9)
21	285	b	6	(3,4)
22	276	=	6	(5,6)
23	286	2	6	(7,8)
24	277	;	6	(8,9)
25	285	c	7	(3,4)
26	276	=	7	(5,6)
27	287	1.1	7	(7,10)
28	277	;	7	(10,11)
29	285	a	9	(3,4)
30	276	=	9	(5,6)
31	285	a	9	(7,8)
32	259	+	9	(9,10)
33	285	b	9	(11,12)
34	277	;	9	(12,13)
35	263	return	11	(3,9)
36	285	a	11	(10,11)
37	277	;	11	(11,12)
38	284	}	13	(1,2)

parser输出如下:

```
1 | >--+ program
```

```

2 | >--+ declaration-list
3 | | >--+ declaration
4 | | | >--+ fun-declaration
5 | | | | >--+ type-specifier
6 | | | | | >--* int
7 | | | | >--* main
8 | | | | >--* (
9 | | | | >--+ params
10 | | | | | >--* void
11 | | | | >--* )
12 | | | | >--+ compound-stmt
13 | | | | | >--* {
14 | | | | | >--+ local-declarations
15 | | | | | | >--+ local-declarations
16 | | | | | | | >--+ local-declarations
17 | | | | | | | | >--+ local-declarations
18 | | | | | | | | | >--* epsilon
19 | | | | | | | | | >--+ var-declaration
20 | | | | | | | | | >--+ type-specifier
21 | | | | | | | | | | >--* int
22 | | | | | | | | | | >--* a
23 | | | | | | | | | | >--* ;
24 | | | | | | | | >--+ var-declaration
25 | | | | | | | | >--+ type-specifier
26 | | | | | | | | | >--* int
27 | | | | | | | | | >--* b
28 | | | | | | | | | >--* ;
29 | | | | | | | >--+ var-declaration
30 | | | | | | | >--+ type-specifier
31 | | | | | | | | >--* float
32 | | | | | | | | >--* c
33 | | | | | | | | >--* ;
34 | | | | | >--+ statement-list
35 | | | | | | >--+ statement-list
36 | | | | | | | >--+ statement-list
37 | | | | | | | | >--+ statement-list
38 | | | | | | | | | >--+ statement-list
39 | | | | | | | | | | >--+ statement-list
40 | | | | | | | | | | >--* epsilon
41 | | | | | | | | | | >--+ statement
42 | | | | | | | | | | >--+ expression-stmt
43 | | | | | | | | | | | >--+ expression
44 | | | | | | | | | | | | >--+ var
45 | | | | | | | | | | | | | >--* a
46 | | | | | | | | | | | | | >--* =
47 | | | | | | | | | | | | | >--+ expression
48 | | | | | | | | | | | | | >--+ simple-expression
49 | | | | | | | | | | | | | | >--+ additive-expression
50 | | | | | | | | | | | | | | >--+ term
51 | | | | | | | | | | | | | | | >--+ factor
52 | | | | | | | | | | | | | | | >--+ integer
53 | | | | | | | | | | | | | | | | >--* 1

```

```

54 | | | | | | | | | | | | | >--* ;
55 | | | | | | | | | | | >--+ statement
56 | | | | | | | | | | | >--+ expression-stmt
57 | | | | | | | | | | | >--+ expression
58 | | | | | | | | | | | >--+ var
59 | | | | | | | | | | | >--* b
60 | | | | | | | | | | | >--* =
61 | | | | | | | | | | | >--+ expression
62 | | | | | | | | | | | >--+ simple-expression
63 | | | | | | | | | | | >--+ additive-expression
64 | | | | | | | | | | | >--+ term
65 | | | | | | | | | | | >--+ factor
66 | | | | | | | | | | | >--+ integer
67 | | | | | | | | | | | >--* 2
68 | | | | | | | | | | | >--* ;
69 | | | | | | | | | | | >--+ statement
70 | | | | | | | | | | | >--+ expression-stmt
71 | | | | | | | | | | | >--+ expression
72 | | | | | | | | | | | >--+ var
73 | | | | | | | | | | | >--* c
74 | | | | | | | | | | | >--* =
75 | | | | | | | | | | | >--+ expression
76 | | | | | | | | | | | >--+ simple-expression
77 | | | | | | | | | | | >--+ additive-expression
78 | | | | | | | | | | | >--+ term
79 | | | | | | | | | | | >--+ factor
80 | | | | | | | | | | | >--+ float
81 | | | | | | | | | | | >--* 1.1
82 | | | | | | | | | | | >--* ;
83 | | | | | | | | | | | >--+ statement
84 | | | | | | | | | | | >--+ expression-stmt
85 | | | | | | | | | | | >--+ expression
86 | | | | | | | | | | | >--+ var
87 | | | | | | | | | | | >--* a
88 | | | | | | | | | | | >--* =
89 | | | | | | | | | | | >--+ expression
90 | | | | | | | | | | | >--+ simple-expression
91 | | | | | | | | | | | >--+ additive-expression
92 | | | | | | | | | | | >--+ additive-expression
93 | | | | | | | | | | | >--+ term
94 | | | | | | | | | | | >--+ factor
95 | | | | | | | | | | | >--+ var
96 | | | | | | | | | | | >--* a
97 | | | | | | | | | | | >--+ addop
98 | | | | | | | | | | | >--* +
99 | | | | | | | | | | | >--+ term
100 | | | | | | | | | | | >--+ factor
101 | | | | | | | | | | | >--+ var
102 | | | | | | | | | | | >--* b
103 | | | | | | | | | | | >--* ;
104 | | | | | | | | | | | >--+ statement
105 | | | | | | | | | | | >--+ return-stmt

```

106									>--*	return
107									>--+	expression
108									>--+	simple-expression
109									>--+	additive-expression
110									>--+	term
111									>--+	factor
112									>--+	var
113									>--*	a
114									>--*	;
115									>--*	}

## 实验反馈

建议把正则表达式书写语法详细介绍一下，比如使用 `[] " "` 和不加任何修饰符的表达式存在何种区别等。