

实验三、自动生成语法分析程序 (JavaCUP)

实验三、自动生成语法分析程序 (JavaCUP)

实验步骤 2.1、下载自动生成工具 JavaCUP

实验步骤 3.2、配置和试用 JavaCUP

实验步骤 3.3、生成 Oberon-0 语法分析和语法制导翻译程序

2.3.4 实验步骤 3.4、讨论不同生成工具的差异

实验步骤 2.1、下载自动生成工具 JavaCUP

上官网下载即可，下载页面截图如下：



Legal issues

实验步骤 3.2、配置和试用 JavaCUP

1. 执行jflex下的例子cup下的ycalc.cup

```

[→ cup java -jar java-cup/java-cup-11b.jar ycalc.cup
----- CUP v0.11b 20160615 (GIT 4ac7450) Parser Generation Summary -----
  0 errors and 0 warnings
 11 terminals, 6 non-terminals, and 14 productions declared,
 producing 23 unique parse states.
  0 terminals declared but not used.
  0 non-terminals declared but not used.
  0 productions never reduced.
  0 conflicts detected (0 expected).
Code written to "parser.java", and "sym.java".
----- (CUP v0.11b 20160615 (GIT

```

2. 样例程序运行结果

```

<terminated> Main (2) [Java Application] /Library/Java/
2 + 4 = 6
5 * ( 6 - 3 ) + 1 = 16
6 / 3 * 5 + 20 = 30
4 * 76 / 31 = 9
1 - 1 - 1 = -1
|

```

实验步骤 3.3、生成 Oberon-0 语法分析和语法制导翻译程序

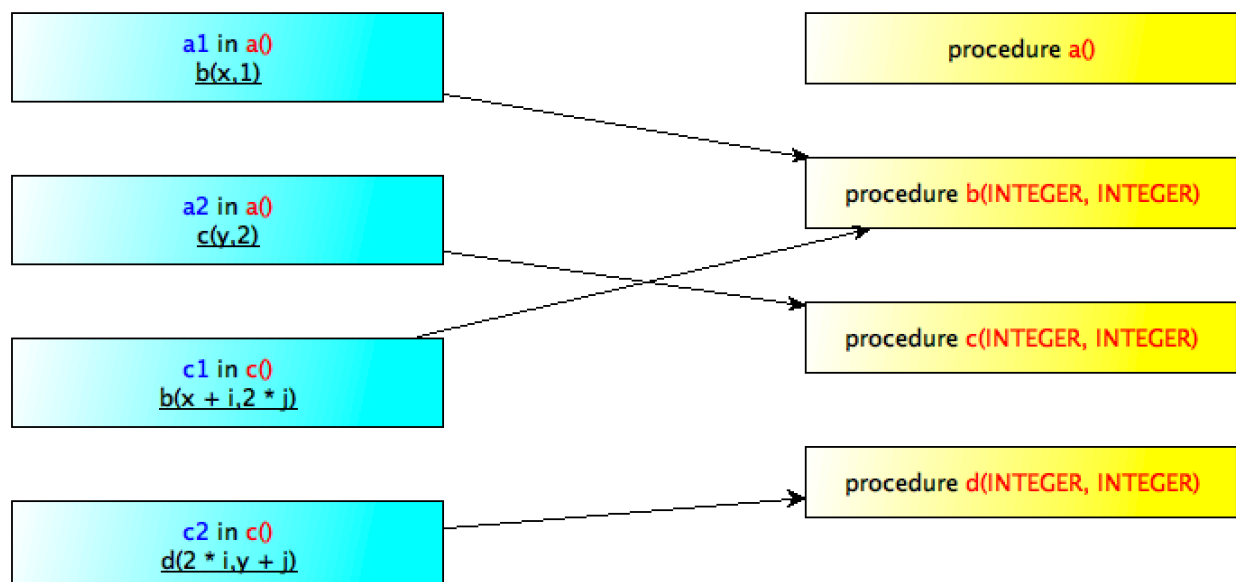
1. 按照文档提供的 Oberon-0 语言的 EBNF 编写 oberon.cup，实现语法分析（此时未实现忽略语法错误检测，便于调试调用图生成工具）。

```

[→ src git:(master) ✕ java -jar java-cup/java-cup-11b.jar oberon.cup
Warning : Terminal "WRITE" was declared but never used
Warning : Terminal "WRITELN" was declared but never used
Warning : Terminal "READ" was declared but never used
----- CUP v0.11b 20160615 (GIT 4ac7450) Parser Generation Summary -----
 0 errors and 3 warnings
48 terminals, 38 non-terminals, and 87 productions declared,
producing 163 unique parse states.
3 terminals declared but not used.
0 non-terminals declared but not used.
0 productions never reduced.
0 conflicts detected (0 expected).
Code written to "parser.java", and "sym.java".
----- SEMI- (CUP v0.11b 20160615 (GIT 4
ac7450))

```

2.添加代码，阅读调用图使用文档，实现调用图生成功能。



3.添加检测语法错误的代码，实现语法错误检测

```
../testcase/SomeFuncs.007:
13 line 15 column
Syntactic Exception: Missing LeftParenthesis Exception.

../testcase/SomeFuncs.008:
14 line 14 column
Syntactic Exception: Missing Right Parenthesis Exception.

../testcase/SomeFuncs.009:
14 line 9 column
Syntactic Exception: Missing Operator Exception.

../testcase/SomeFuncs.010:
11 line 12 column
Syntactic Exception: Missing Operand Exception.

../testcase/SomeFuncs.011:
6 line 8 column
Semantic Exception: Type Mismatched Exception.
```

2.3.4 实验步骤 3.4、讨论不同生成工具的差异

1. Bison和Cup在消除二义性上都是分两步进行的。首先是按照优先级规则消除部分移进/归约冲突，其次是按照确定性规则解决余下的冲突，对sr冲突，优先移进，对rr冲突，选用列在文法最前面的语法规则进行归约。但在实际操作文法时可能会产生不完全一致的冲突解决结果。
2. 在描述语言文法的规则上，两者有较大不同。在 Bison 中，你必须指明参加具体 reduce 动作的 token，或者指明 non terminal 符号所在的位置，reduce 之后的结果是通过\$\$返回的，在 javaCUP 中，你可以给参加具体 reduce 动作的 token 或者其他的 nonterminal 一个具体的“变量”，然后根据这个变量来参加具体的动作规约。返回的结果都是用 RESULT 来返回的，而且我们可以通过声明阶段赋予 RESULT 所属 terminal 或者 non terminal 的类型，这样我们就可以通过取变量的方式来获取归约结果的值了。
3. 结合的词法分析工具不一样，别结合使用的是 lex 和 jflex。