

编译原理课程实验报告

实验 3 - 递归向下方法的语法分析器

实验时间: 18th Nov, 2016

1) 实验内容与目的

实验 3 要求使用递归下降法编写语法分析程序，使之与实验 2 中的词法分析器结合，根据文法判断输入的单词序列是否为文法的句子。

2) 实现分析

经文法改写消除左递归、提取左公因式并解决“悬空 - else ”问题后得到如下文法，其中 `e` 代表空串：

```
program → block
block   → { stmts }
stmts   → stmt stmts
        | e
stmt    → if ( bool ) stmt elsepart
        | id = expr ;
        | while ( bool ) stmt
        | do stmt while ( bool ) ;
        | break ;
        | block
elsepart→ else stmt    //Limit: use when lookahead = ELSE_TKN
        | e
bool    → expr comp    //Extracted left common factor
comp    → < expr
        | <= expr
        | > expr
        | >= expr
        | e
expr    → term rest     //Eliminated left recursive
rest    → + term rest
        | - term rest
        | e
term    → factor section //Eliminated left recursive
section → * factor section
        | / factor section
        | e
factor  → ( expr )
        | id
        | num
```

需要注意的是，这里将原文法改写为了 LL(1) 文法，并且在“悬空 - else”的解决上对 `elsepart` 的匹配加以限制，当 `lookAhead` 为 `ELSE_TKN` 时才选择第一个产生式，从而语法分析器总是可以通过 `lookAhead` 的值选取唯一的产生式。

在发现错误时，语法分析器会将全局变量 `status` 加一，若最后 `status` 的值不为 0，语法分析器会报告输入的单词序列不是指定文法的句子，否则语法分析器会报告可以接受输入的单词序列。

词法分析器的实现见 `tokenizer.lex`，Flex 生成的结果为 `lex.yy.c`，语法分析器源代码实现见 `parser.cpp`。

3) 实验结果

实验所采用的 flex 版本为 flex 2.5.35, C 编译器为 LLVM version 8.0.0, 系统环境为 macOS 10.12.2.

- 接受的示例 `test_succ.c`：

```
{
    i = 2;
    do {
        if(c != 2 ) c = 5;
        else {
            c = c + bpp;
            if(app) {
                bpp = bpp * (1 / c);
                c = bpp;
            }
            /* Remarks can be */
            break;
        }
    } while (p <= /* everywhere. */ 2);
}
```

语法分析器可以正确识别各个词素并接受这个序列。

- 拒绝的示例 `test_fail.c`：

```

{
    i = 2;
    if (p == 2) {
        if(cite >= perm) {
            cite = perm;
        } else
            break;
        } else {
        if (p <= 0)
            cite = aol;
            perm = aol;
        /* The Following else should fail */
        else {
            perm = aol; p = aol;
        }
    }
}
}

```

部分程序输出：

```

...
Remark:/* The Following else should fail */
ELSE:else
match error
[INFO] lookAhead 611 else but excepting 535
match error
[INFO] lookAhead 611 else but excepting 535

=== PARSER REJECTED INPUT FILE ===

```

这里第三个 `else` 因为找不到合适的 `if` 导致语法出错。

4) 尚需考虑的问题

关于“悬空 - else”问题的解决，实验指导中提出了引入 `unmatched_stmt` 与 `matched_stmt` 两个新的非终结符的方案，但是对于递归下降方法，在进行实验时没有找到一个有效的能够根据 `lookAhead` 判断应该采取哪种产生式的方法，这一问题还需考虑。