

编译原理 · parser-stage · 实验报告

计01 容逸朗 2020010869

实验内容

实验目标

结合课堂上学习的 LL(1) 分析方法，完成一个手工实现的递归下降语法分析器。

具体实现

本次实验只需要修改 `src/frontend/my_parser.cpp` 的 `p_Type`, `p StmtList`, `p_Statement`, `p_VarDecl`, `p_Return`, `p_If`, `p_Expression`, `p_Assignment`, `p_LogicalAnd` 及 `p_Relational` 函数即可。

需要实现的语法

```
1 type      : Int
2 StmtList  : (Statement | VarDecl ';')*
3 statement : if | return | ( expression )? ';' | '{' StmtList '}'
4 declaration : type Identifier ('=' expression)?
5 return     : 'return' expression ';'
6 if         : 'if' '(' expression ')' statement ( 'else' statement )?
7 expression : assignment
8 assignment : Identifier '=' expression | conditional
9 logical_and : equality { '&&' equality }
10 relational : additive { '<' additive | '>' additive | '≤' additive | '≥' additive }
```

- 对于特定的终结符（如 `Int`, `Identifier`, `;` 等），可以直接调用 `lookahead()` 函数；
- 对于非终结符，可以直接调用对应的 parser；（例如 `assignment` 的 `p_Assignment()`）
- 对于含有 `()?` 和 `|` 的语句，可以配合 `if` 和 `next_token.type` 来判断下一个符号是否符合对应的语句条件；
- 对于含有 `{ }` 和 `()*` 的语句，可以配合 `while` 和 `next_token.type` 来不断判断下一个符号是否符合条件，若符合则选择之；
- 形如 `{ '<' | '>' | '≤' }` 的语句，可以使用 `switch` 判断符号类型并选择对应的语句。

思考题

1. 在框架里我们使用 EBNF 处理了 `additive` 的产生式。请使用课上学习的消除左递归、消除左公因子的方法，将其转换为不含左递归的 LL(1) 文法。（不考虑后续 `multiplicative` 的产生式）

```

1 | additive : additive '+' multiplicative
2 |         | additive '-' multiplicative
3 |         | multiplicative

```

- 所求文法如下：

```

1 | additive : multiplicative additive2
2 | additive2 : '+' multiplicative additive2
3 |           | '-' multiplicative additive2
4 |           | /**empty**/

```

2. 对于我们的程序框架，在自顶向下语法分析的过程中，如果出现一个语法错误，可以进行错误恢复以继续解析，从而继续解析程序中后续的语法单元。请尝试举出一个出错程序的例子，结合我们的程序框架，描述你心目中的错误恢复机制对这个例子，怎样越过出错的位置继续解析。（注意目前框架里是没有错误恢复机制的。）

- 一个出错程序的例子如下：（[failcases/step6/badparse.c](#)）

```

1 | int main() {
2 |     return 1 ? 2 : { 3; };
3 | }

```

- 程序扫描至（2,20）时，出现了 `{`。由于 `conditional` 语句不包含此符号，因此跳过并分析下一个符号即可；
- 由于 `3` 和 `;` 都是可行的，此时程序可以继续分析；
- 程序扫描至（2,25）时遇到了 `}`，由于没有匹配的表达式，此时同样采用舍弃策略；
- 对于剩下的 `;` 识别为空语句即可（尽管不会被执行到）。

3. 指出你认为的本阶段的实验框架/实验设计的可取之处、不足之处、或可改进的地方。

- 本阶段的实验结合了课上所学的知识，在做实验的过程会有学以致用用的感觉。除此之外，由于实验框架的注释十分清晰，因此完成本阶段任务的体验较好。

参考

实现代码的过程中没有参考额外资料。