

编译原理 PA1-A 实验报告

李源隆 (2016011261)

完成的实验内容

本实验的内容是给现有的编译器框架中词法分析和语法分析的部分增加新的特性，目标是可以生成对应的抽象语法树。

对于每个新增加的特性，大体的流程是：

- ①在词法分析中引入新关键字和新使用的符号；
- ②将它们加入到辅助定义中作为终结符，同时设定其结合性和优先级；
- ③在语法分析器中添加规则，并对已有规则进行修改以应用自己的规则；
- ④对每个新加入的非终结符进行数据传输的处理；
- ⑤在语法树中添加新的特性节点，传送语法规则中的数据内容；
- ⑥修改 `printTo` 函数以使用传送的数据来输出语法分析树。

问题与解决

问题 1：在实现串行条件卫士时，曾一度不清楚如何传输列表型数据。

解决：后来发现，对于每个非终结符都会对应一个 `SemValue` 对象。而在这个对象中有着已经定义好的成员变量（如 `elist` 和 `slist`）以传输数据。最后用于输出到语法分析树的是其中的 `stmt` 成员。在知道了语法分析这一部分的工作机制之后，就解决了这个问题。

问题 2：在实现 `default` 关键字时，发现如果不设置其相对于其它运算符的优先级，会出现先进行二元运算后进行 `default` 运算的情况。

解决：设置 `default` 关键字的优先级在相反数运算和取非运算之上，符号 `'` 和 `''` 之下。另外，因为 `'` 的优先级本身很高，其它大部分数组运算都不需要考虑这个问题。（`%%` 和 `++` 运算已经规定了优先级所以也不会出问题）

问题 3：在实现数组常量 `ArrayConst` 中，曾尝试了两种语法规则。

$$\begin{aligned} \text{规则1:} & \begin{cases} \text{ArrayConst} \rightarrow [] [\text{ConstList}] \\ \text{ConstList} \rightarrow \text{Const} \mid \text{ConstList}, \text{Const} \end{cases} \\ \text{规则2:} & \begin{cases} \text{ArrayConst} \rightarrow [] [\text{ConstList} \quad \text{Const}] \\ \text{ConstList} \rightarrow \varepsilon \mid \text{ConstList} \quad \text{Const}, \end{cases} \end{aligned}$$

个人认为两个规则在语义上都没有问题，但在实践中发现前者可以成功，后者会出现 `Syntax Error`，错误位置指向非空数组的第一个逗号或者右括号（若没有逗号）

解决：最终采用了规则 1 完成了实验，但是凭笔者目前对语法分析器的了解还不清楚为何之前采用的规则 2 是不可行的，希望可以得到解答。