



华南师范大学

South China Normal University

编译原理课程项目三

子实验一—熟悉 Oberon-0 语言定义

实验报告

设计题目：熟悉 Oberon-0 语言定义

院 系：计算机学院

指导老师：王欣明

专业班级：15 级数据库 2 班

完成学生：陈天一

一、讨论Oberon-0语言的特点

Oberon-0 语言是一种结构化编程语言，具有顺序、选择、循环结构，Oberon-0 与 Java、C/C++等常见语言的表达式相比：

(1) 支持的数据仅支持布尔 (BOOLEAN) 和整数 (INTEGER) 类型，不支持浮点型或大整数等类型；

(2) 不支持 FOR 和 DOWHILE 等循环类型；

(3) 具有复合数据类型：数组 (ARRAY)、记录 (RECORD)，但是对于类 (CLASS) 和结构体 (STRUCT) 等功能更强大的复合数据没有出现在 Oberon-0 中；

(4) Oberon-0 中 IF 和 WHILR 语句都对应着 END 来结尾，并且在 IF 语句中使用 THEN 来进行对 IF 和 ELSIF 的子句结尾；

(5) 在类型的转换上不支持，和其他常见语言不同；

(6) 没有主函数；

(7)在变量的声明方面，Oberon-0 中采用 VAR 说明后面的是一个变量，而 Java、C/C++则不需要；

(8) Oberon 使用函数过程 PROCEDURE 来构造函数，且用 BEGIN 和 END 来标志函数体内容，而 Java、C/C++等常用语言使用花括号 {} 来实现；

二、讨论Oberon-0文法定义的二义性

Oberon-0 文法没有二义性，在其他高级程序设计语言中常见的二义性问题主要有如下两种情况：

(1) 算数表达式计算顺序方面产生二义性：

Oberon-0 EBNF：

```
simple_expression = ["+" | "-"] term {("+" | "-" | "OR") term}  
term = factor {("*" | "DIV" | "MOD" | "&") factor} ;
```

从以上 EBNF 可以看出，乘法*、除法 DIV、求余 MOD 在加减运算之间一定会得到计算，所以算数表达式的计算顺序上，消除了这种可能的二义性。

(2) 对于 IF-ELSE-END 语句方面产生二义性：

Oberon-0 EBNF：

```
if_statement = "IF" expression "THEN"  
statement_sequence  
{ "ELSIF" expression "THEN"  
statement_sequence}  
["ELSE" statement_sequence]  
"END" ;
```

从以上 EBNF 中看出，每个 IF 语句都会对应 END 标识符，这样就不会出现 IF-ELSE 的匹配问题，在 C 语言中，可能会出现以下匹配问题：

```
IF (statement)  
    IF(statement)  
ELSE statement2;
```

此时存在文法的二义性，ELSE 的配对问题，在 Oberon-0 语法中，使用 IF-END 与 ELSIF-THEN 组合配对避免了这一情况的发生。

三、实验心得体会

(1) 通过对 Oberon-0 的程序书写，对 Oberon-0 语言有了初步的认识，对其结构、语法、数据类型等也有一些了解；

(2) 与高级程序设计语言相比较，其不像 Java 与 C++ 一样具有更方便的数据类型与编程特性；

(3) 在文法二义性上，对比 C++ 语言特性与 Tiny 语言的文法规则，分析其他语言的二义性出现问题地方，来确定该 Oberon-0 文法是否出现二义性，从中也更加熟悉了二义性判断的相关知识。