



中山大學
SUN YAT-SEN UNIVERSITY

编译原理

作业名称	Oberon-0 逆向工程工具 Rose 实验一：熟悉 Oberon-0 语言定义	任课老师	万老师		
院系	软件学院	方向	通信软件	姓名	李明宽
学号	11331173	完成日期	2013 年 11 月 24 日		
QQ	736459905	E-mail	limkuan@mail2.sysu.edu.cn		

实验一：熟悉 Oberon-0 语言定义

一、	实验目的.....	2
二、	实验过程.....	2
1.	编写一个正确的 Oberon-0 语言	2
2.	编写上述 Oberon-0 源程序的变异程序	2
3.	讨论 Oberon-0 语言的特点	3
4.	讨论 Oberon-0 文法定义的二义性	4
三、	实验成果展示.....	5
1.	目录结构与文件说明	5
2.	样例程序运行截图：	6
四、	实验心得.....	7

一、实验目的

本实验通过学习实验文档第三部分给出的 Oberon-0 语言的完整定义，做到基本熟悉 Oberon-0 语言，为接下来对这个语言的处理实验做准备。

二、实验过程

1. 编写一个正确的 Oberon-0 语言

编写正确的 Oberon-0 语言的过程中，最好尽量做到能基本覆盖所有的文法，例如变量的声明，Record 结构体的定义，过程的定义，IF 判断语句，WHILE 循环语句等等，并且要做到程序最好有实际意义。

我实现的正确的 Oberon-0 程序代码见<[Sample.obr](#)>，其中包括了注释，包含两个整型变量的结构体，以及数组的使用，在一个过程中实现了在数组中查询最大值，以及在另一个过程中实现了计算斐波那契数列第 N 项，后来为了配合函数调用关系，实现了几个没有实际作用的函数做嵌套。在主函数里，通过 IF 语句判断需要执行什么操作，基本覆盖了所有的文法。

为了检测文法设计的正确性，我使用了一个 Oberon-2 的编译器 obc（这个编译器的官方网址 http://spivey.oriel.ox.ac.uk/corner/Installing_OBC_release_2.9），这个编译器所定义的文法基本上与文档给出的类似，只是输入输出使用了 In 和 Out 这两个内置实现的函数库，并且支持字符串，因此只需要编写这个编译器能够通过的程序，然后对他进行相应的修改即可。

2. 编写上述 Oberon-0 源程序的变异程序

编写了正确的 Oberon-0 源程序后，就要写出若干含有词法、语法或者语义错误的 Oberon-0 源程序。其做法是对原有的正确源程序做出最小的改动，使之成为一个包含至少一个错误的源程序。

具体编写过程的时候，对照文档给出的所有错误 Exception 类型，为每一种可能存在的词法、语法和语义错误构造出一个变异程序。

这个部分我编写了词法错误变异程序 10 个，语法错误变异程序 10 个，语义错误变异程序 8 个一共 28 个原来正确的 Oberon-0 程序的变异程序，分别在 [LexicalException](#)、[SyntacticException](#)、[SemanticException](#) 这三个文件夹中。

3. 讨论 Oberon-0 语言的特点

1) Oberon-0 语言是一个类 Pascal 语言，语法中有很多与 Pascal 语言相类似的地方，它也比较简单，整个语言的 EBNF 文法定义只有一页，它有以下特点：

- 基本来说支持了顺序语句，条件语句，循环语句的执行。有良好的程序结构。
- Oberon-0 语言没有指针以及 Java 中经常使用的引用，因此这样对后续的解析程序编写提供了很多的方便。
- 语言的基本数据类型有 Integer 整型以及 Boolean 布尔型，要注意的是语言中不支持书写、Boolean 类型的 True 和 False。
- 对于标识符，是大小写不敏感的，并且标识符长度不允许超过 24 个字符（一般的正常的变量都不会超过的吧……）
- Oberon-0 语言支持过程的声明与调用，并且还提供了值传递以及按引用传递这两种不同的参数传递方式，感觉没有返回值，如果需要在过程中返回数据，那就返回的参数按引用传递即可。

2) 关键字与保留字的区别：

实验文档中举了一些关于保留字以及关键字的例子，说明了 IF、THEN、ELSIF 是保留字，而 INTEGER、WRITE、WRITELN 是关键字，我不是很赞同这个说法，我认为 IF、THEN、ELSIF、INTEGER 都是关键字，而 WRITE、WRITELN 因为是预定义的函数名，因此应该属于标识符。保留字（英语：Reserved word），有时也叫关键字（keyword），是编程语言中的一类语法结构。在特定的编程语言里，这些保留字具有较为特殊的意义，并且在语言的格式说明里被预先定义。通常，保留字包括用来支持类型系统的原始数据类型的标记，并可以用来识别诸如循环结构、语句块、条件、分支等程序结构（摘自维基百科 <http://zh.wikipedia.org/wiki/保留字>），因此我觉得在文法定义过程中出现了的标识符就是关键字，而文法中没有出现的，不具有具体含义的，但不能作为标识符的字符串就称作保留字。

但如果考虑到后面语法分析的过程，应该说有具体含义的词就是关键字，而没有具体含义的词就是保留字，例如 IF、ELSIF、END 等，只是表明了程序的结构，而没有具体的含义，但是一些有具体含义的词如 ARRAY、INTEGER、RECORD 是关键词。

3) Oberon-0 语言与常见语言如 JAVA、C/C++的相同点如下：

- 都含有顺序语句、选择语句以及循环语句这三种基本的语句。
- 都能自定义结构体类型。C/C++使用 struct 关键字，Oberon-0 使用 RECORD。
- 都是强类型语言，变量使用前必须声明。

4) Oberon-0 语言与常见语言的不同点如下:

- 在类型方面, C/C++支持类型转换, 并且比 Oberon-0 多了字符类型的数据。
- 在程序的结构方面, Oberon-0 没有 FOR 循环和 DO...WHILE 循环, 而且 C/C++ 使用大括号来划分程序结构, Oberon 使用 END 语句作为语句段的结束。
- 函数和过程方面, C++支持函数的返回值, Oberon-0 好像没有函数的返回值, 如果需要在函数中返回数据, 需要使用按引用传递参数返回。
- C/C++有指针, 对于一些对内存数据的操作会更加灵活。

4. 讨论 Oberon-0 文法定义的二义性

Oberon-0 文法定义没有二义性。

一般的一些高级语言中, 常见的二义性问题是 else-悬空问题, 就是如果有出现如下的程序段:

```
IF expression THEN
    IF expression THEN
        statement
    ELSE statement
```

这样下面一个 ELSE 语句就不知道应该和哪个 IF 语句相匹配, 然而在 C/C++中采用了大括号划分清楚语句的界限, 在 Oberon-0 语言中, 在每一个 IF 语句后面结束的时候都要使用 END 语句, 这样就划分清楚 ELSE 应该属于哪一个 IF 语句。从而解决了这个常见的二义性问题。

而且, 对于运算符的优先级, 在文法中采用了两层的文法推导式的定义, 优先定义了+/-等优先级较低的运算符, 然后再定义优先级较高的*、DIV、MOD 等运算符, 这样保证进行语法推导的时候优先按+/-展开, 也就是说计算过程中优先计算*、DIV 等, 这样符合了运算符之间的优先级, 也消除了这个优先级导致的二义性。

三、实验成果展示

1. 目录结构与文件说明

obc-linux-2.9.3.tar.gz	验证 Oberon-0 语言正确性的编译器
Oberon-0.doc	实验报告文档
Oberon-0.pdf	实验报告文档 PDF 格式
readme.txt	README 说明文档
Sample	正确的程序编译出来的 Linux 可执行文件
Sample.exe	正确的程序编译出来的 Win 可执行文件
testcases	测试样例文件夹
Sample.obr	正确的 Oberon-0 源程序
LexicalException	词法错误变异程序文件夹
Sample001	异常 IllegalSymbolException
Sample.02	异常 IllegalIntegerException
Sample.03	异常 IllegalIntegerRangeException
Sample.00	异常 IllegalOctalException
Sample.005	异常 IllegalIdentifierLengthException
Sample.006	异常 MismatchedCommentException
Sample.007	异常 IllegalIntegerException
Sample.008	异常 MismatchedCommentException
Sample.009	异常 MismatchedCommentException
Sample.010	异常 IllegalOctalException
SemanticException	语义错误变异程序文件夹
Sample.001	异常 TypeMismatchedException
Sample.002	异常 TypeMismatchedException
Sample.003	异常 TypeMismatchedException
Sample.004	异常 ParameterMismatchedException
Sample.005	异常 ParameterMismatchedException
Sample.006	异常 ParameterMismatchedException
Sample.007	异常 ParameterMismatchedException
Sample.008	异常 TypeMismatchedException
SyntacticException	语法错误变异程序文件夹
Sample.001	异常 MissingLeftParenthesisException
Sample.002	异常 MissingLeftParenthesisException
Sample.003	异常 MissingLeftParenthesisException
Sample.004	异常 MissingRightParenthesisException
Sample.005	异常 MissingRightParenthesisException
Sample.006	异常 MissingRightParenthesisException
Sample.007	异常 MissingOperatorException
Sample.008	异常 MissingOperatorException
Sample.009	异常 MissingOperandException
Sample.010	异常 MissingOperandException

4 directories, 35 files

2. 样例程序运行截图：

- 运行正确的 Oberon 程序，选项选择 1 计算一个数组的最大值：

```
➔ ex1 ./Sample
1 - Compute the maximum number
2 - Compute Fibonacci sequence
3 - Test the call of functions
Other - Quit
What do you want to do? : 1
Please input the number of student: 5
Please input each student's score:
1 5 10 6 2
The user 3 get the highest score.The score is 10
```

- 选项中选择 2，计算斐波那契数列第 n 项：

```
The user 3 get the highest score.The score is 10
1 - Compute the maximum number
2 - Compute Fibonacci sequence
3 - Test the call of functions
Other - Quit
What do you want to do? : 2
Compute Fibonacci sequence...
Please input the position to compute: 10
The Fibonacci number of that position is 55
```

- 选项中选择 3，测试函数嵌套调用，然后输入非 123 之外的数字结束程序：

```
1 - Compute the maximum number
2 - Compute Fibonacci sequence
3 - Test the call of functions
Other - Quit
What do you want to do? : 3
This is Fun3, The parameter is 4
This is Fun2, The parameter is 0
This is Fun2, The parameter is 1
This is Fun1, The parameter is 0
This is Fun2, The parameter is 2
This is Fun1, The parameter is 0
This is Fun1, The parameter is 1
This is Fun2, The parameter is 3
This is Fun1, The parameter is 0
This is Fun1, The parameter is 1
This is Fun1, The parameter is 2
1 - Compute the maximum number
2 - Compute Fibonacci sequence
3 - Test the call of functions
Other - Quit
What do you want to do? : 4
➔ ex1
```

四、 实验心得

这是 Project3 逆向工程的第一部分实验，通过这部分实验，基本了解了 Oberon-0 语言的一些特性以及使用方法，并且与其他高级语言的对比也明白 Oberon-0 语言与类 C/C++ 语言的相同与不同点，为接下来的词法分析以及语法分析做好了准备。

通过变异测试程序的编写，也对软件测试技术中变异测试有了一定的了解，他的目标变异测试的目标是筛选更有效的测试数据,其思路是通过变异操作在程序中植入一个错误,变异后的程序称为变异程序，通过检查测试数据能否发现变异程序中的这些错误,可判断测试数据的有效性。这种测试思路对我们的以后的程序设计有很大的启发。

Thanks\(^o^)/~