
编译原理实验指导书

实验二 语义分析

实验目的

1. 巩固对语义分析的基本功能和原理的认识。
2. 能够基于语法指导翻译的知识进行语义分析。
3. 理解并处理语义分析中的异常和错误。

实验内容

在词法分析和语法分析程序的基础上编写一个程序，对C--源代码进行语义分析和类型检查，并打印分析结果。与实验一不同的是实验二不再借助已有的工具，所有的任务都必须通过手写代码来完成。

在本次实验中，对C--语言做如下假设，你可以认为这些就是C--语言的特性：

- 1) 假设1：整型（int）变量不能与浮点型（float）变量相互赋值或者相互运算。
- 2) 假设2：仅有int型变量才能进行逻辑运算或者作为if和while语句的条件；仅有int型和float型变量才能参与算术运算、
- 3) 假设3：任何函数只进行一次定义，无法进行函数声明。
- 4) 假设4：所有变量（包括函数的形参）的作用域都是全局的，即程序中所有变量均不能重名。
- 5) 假设5：结构体间的类型等价机制采用名等价（Name Equivalence）的方式。
- 6) 假设6：函数无法进行嵌套定义。
- 7) 假设7：结构体中的域不与变量重名，并且不同结构体中的域互不重名。

以上假设1假设7也可视为要求，违反即会导致各种语义错误，不过我们只对后面讨论的17种错误类型进行考察。此外，你可以安全地假设输入文件中不包含注

释、八进制数、十六进制数以及指数形式的浮点数，也不包含任何词法或语法错误（除了特别说明的针对选做要求的测试）。

你的程序需要对输入文件进行语义分析（输入文件中可能包含函数、结构体、一维和高维数组）并检查如下类型的错误

■ 变量（包括数组、指针、结构体）或过程未经声明就使用

错误类型1：变量在使用时未经定义。

错误类型2：函数在调用时未经定义。

错误类型14：访问结构体中未定义过的域。

错误类型17：直接使用未定义过的结构体来定义变量。

■ 变量（包括数组、指针、结构体）或过程名重复声明

错误类型3：变量出现重复定义或变量与前面定义过的结构体名字重复。

错误类型4：函数出现重复定义（即同样的函数名被多次定义）。

错误类型15：结构体中域名重复定义（指同一结构体中）。

错误类型16：结构体的名字与前面定义过的结构体或变量的名字重复。

■ 运算分量类型不匹配

错误类型5：赋值号两边的表达式类型不匹配。

错误类型7：操作数类型不匹配或操作数类型与操作符不匹配（例如整型变量与数组变量相加减，或数组（或结构体）变量与数组（或结构体）变量相加减）。

■ 操作符与操作数之间的类型不匹配

错误类型6：赋值号左边出现一个只有右值的表达式。

错误类型12：数组访问操作符 “[...]” 中出现非整数（例如 `a[1.5]`）。

错误类型10：对非数组型变量使用数组访问操作符。

错误类型11：对非过程名使用过程调用操作符。

错误类型13：对非结构体类型变量使用 “.” 操作符。

错误类型9：过程调用时实参与形参的数目或类型不匹配。

错误类型8：`return` 语句的返回类型与函数定义的返回类型不匹配。

其中，要注意两点：一是关于数组类型的等价机制。同C语言一样，只要数组的基类型和维数相同我们就认为类型是匹配的，例如 `int a[10][2]` 和 `int b[5][3]` 即

属于同一类型；二是我们允许类型等价的结构体变量之间的直接赋值（见测试样例）。这时的语义是，对应的域相应赋值（数组域也如此，按相对地址赋值直至所有数组元素赋值完毕或目标数组域已经填满）。

实验要求

（1）输入格式

程序的输入是一个包含C--源代码的文本文件，该源代码中可能会有语义错误。程序需要能够接收一个输入文件名作为参数。例如，假设程序名为、输入文件名为test1、程序和输入文件都位于当前目录下，那么在Linux命令行下运行`.cc test1`即可获得以test1作为输入文件的输出结果。

（2）输出格式

实验二要求通过标准输出打印程序的运行结果。对于那些没有语义错误的输入文件，你的程序不需要输出任何内容。对于那些存在语义错误的输入文件，程序应当输出相应的错误信息，这些信息包括错误类型、出错的行号以及说明文字，其格式为：

Error type 【错误类型】 at Line 【行号】: 【说明文字】.

说明文字的内容没有具体要求，但是错误类型和出错的行号一定要正确，因为这是判断输出的错误提示信息是否正确的唯一标准。请严格遵守实验要求中给定的错误分类，否则将影响你的实验评分。

输入文件中可能包含一个或者多个错误（但每行最多只有一个错误），程序需要将它们全部检查出来。当然，有时输入文件中的一个错误会产生连锁反应，导致别的地方出现多个错误（例如，一个未定义的变量在使用时由于无法确定其类型，会使所有包含该变量的表达式产生类型错误），我们只会去考察你的程序是否报告了较本质的那个错误（如果难以确定哪个错误更本质一些，建议你报告所有发现的错误）。但是，如果源程序里有错而你的程序没有报错或报告的错误类型不对，又或者源程序里没有错，但你的程序却报错，都会影响你的实验评分。

（3）测试环境

程序将在如下环境中被编译并运行（同实验一）：

1) GNU Linux Release: Ubuntu 12.04, kernel version 3.2.0-29。

-
- 2) GCC version 4.6.3。
 - 3) GNU Flex version 2.5.35。
 - 4) GNU Bison version 2.5。

(4) 现场验收

要求当堂完成实验内容，并进行现场验收。未当堂完成验收者将扣除相应分数。

(5) 提交要求

实验二结束后**一周内**提交如下内容：

- 1) Flex、Bison以及C语言的可被正确编译运行的源程序。
- 2) 份PDF格式的实验报告，内容包括：
 - a) 程序实现了哪些功能？简要说明如何实现这些功能。清晰的说明有助于助教对你的程序所实现的功能进行合理的测试。
 - b) 程序应该如何被编译？可以使用脚本、makefile或逐条输入命令进行编译，请详细说明应该如何编译你的程序。无法顺利编译将导致助教无法对你的程序所实现的功能进行任何测试，从而丢失相应的分数。
 - c) 实验报告的长度不得超过三页。所以实验报告中需要重点描述的是程序中的亮点，是你认为最个性化、最具独创性的内容，而相对简单的、任何人都可以做的内容则可不提或简单地提一下，尤其要避免在报告中贴代码。实验报告中所出现的最小字号不得小于五号字（或英文11号字）。

实验评分标准

一、课堂表现（10分）

1. 出勤情况（迟到，早退，缺席）
2. 是否遵守课堂纪律

二、操作表现（50分）

1. 功能齐全，结果正确无误（25分）
2. 程序输出格式规范（5分）
3. 讲解清楚明了，具有良好的演示效果（10分）
4. 当堂按时完成（10分）

三、实验报告（40分）