

第六章作业

6.2

6.6

6.7

6.8

6.10

参考答案

6.2

在符号表上的操作有四种：插入、检索、定位和重定位。

(1) 插入操作，当语义分析程序识别出一个新名字时被调用。

(2) 检索操作，有两种情况需要调用执行检索操作，①在插入操作之前，在当前符号子表范围内执行检索操作，目的是对名字进行“查重”，因为同一作用域内不允许名字重复定义；②在分析可执行语句的过程中，遇到名字引用时执行检索操作，目的是检查所引用的名字是否合法，如果合法，则从符号表中取得该名字的类型信息，检查是否与上下文要求的类型一致，或者，根据名字的类型信息和存储地址信息，辅助生成目标代码。

(3) 定位操作，当语义分析程序识别到一个块的开始时执行定位操作，目的是建立该块的符号子表，因为块的开始标志着进入了一个新的作用域。

(4) 重定位操作，当语义分析程序识别到一个块的结束时执行重定位操作，目的是从逻辑上或物理上删除该块的符号表，因为块的结束标志着退出当前作用域，退出之后，原作用域内声明的名字不能再用。

通过这些符号表操作，实现源语言的作用域规则，完成类型检查。

6.6

题目中相应名字的类型表达式如下。

(1) `array(1..100, pointer(real))`。

(2) `array(0..9, array(-10..10, integer))`

(3) `(integer→pointer(integer)) → record((a×integer)×(b×char))`

6.7

foo 的类型表达式是：`array(0..99, CELL)`。

bar 的类型表达式为：`int×CELL→PCELL`。

6.8

类型名 `table1` 代表的类型是 `pointer(char)`，类型名 `table2` 代表的类型也是 `pointer(char)`。

名字 `x` 和 `y` 的类型表达式为：`table1`

名字 `z` 和 `w` 的类型表达式为：`table2`

(1) 在名字等价的情况下，名字 `x` 和 `y` 的类型表达式是 `table1`，名字 `z` 和 `w` 的类型表达式是 `table2`，所以，`x` 和 `y` 名字等价，`z` 和 `w` 名字等价。

(2) 在结构等价的情况下，名字 `x`、`y`、`z` 和 `w` 的类型表达式都是 `pointer(char)`，所以，他们都结构等价。

(3) C语言使用的是介于名字等价和结构等价之间的一种类型等价形式，即对于 `struct` 和 `union` 采用名字等价，其他类型则采用结构等价，所以，名字 `x`、`y`、`z` 和 `w` 全部结构等价。

6.10

(1) 由于 x 和 y 是在一个声明语句中声明的两个变量名，他们具有相同的类型，类型是等价的，所以，对于语句 $x=y$ 而言，不会出现类型错误。

(2) 因为 x 和 z 是在两个不同的语句中声明的变量名，虽然结构的定义是一样的，但是 x 和 z 具有不同的类型名， x 与 z 类型不等价。所以，对于语句 $x=z$ 而言，会出现类型错误。