

Decaf/Mind PA2 实验报告

一、实验思路

在我们完成了 PA1 的任务之后，我们得到了一系列的语法符号，以及这些根据这些符号出现的位置所建立起来的抽象语法树，在 PA2 我们需要做的就如 README 里面说的：对得到的 AST 进行遍历，建立起符号表，然后进行语义的检查。符号表的建立过程实验框架已经帮我们完成了，当然因为新增特性，所以没有面面俱到，比如对于 repeatLoop 的作用域就没有建立，对于 Double 的处理也没有照顾到，需要我们自行添加一下。剩下留给我们的就是对一部分的语法符号的语义类型兼容性的检查了，如果不兼容，进行相应的报错处理。

首先，最快的方法正如 README 里面所说的：根据各种报错的类型，逐一考虑各种情形。这其中稍微复杂一点的就是关于避免连环报错的处理了。

我的实验过程是这样的：

首先在留给我们补充的位置，添加给各种语法符号的 type 赋值的代码，这其中可能会需要先递归一下它的子节点，比如 visitNewArray，就需要先检查其 elementType，length 子节点，为其 type 赋值，然后为该节点赋值。这一遍工作完成之后，所有的 type 都不会是 null 了，那么运行的时候就不会产生 NullPointerException 了。

接下来就是在每个需要补充代码的地方结合所有的 Error 类以及测例对各种情况进行报错处理，仍以 visitNewArray 为例，如果 length.type 不是 INT，那就需要报告 BadNewArrayLength 错误。

这一遍工作完成后，所有的测例就都能够通过了，接下来我有仔细结合 Decaf/Mind 语言规范里面的描述，对各种情形进行考虑，以免自己有漏掉的地方。比如：数组没有子类或者类型提升之说：如果 T2 继承自 T1，则 T2[] 类型的数组不兼容于 T1[] 类型的数组，这一点给出的测例里面就没有涉及到，所以这一遍我把前面很多直接用 equal 来判断类型的地方改成了用 compatible 来判断。这一遍改了至少有 4 个地方，感觉还是有必要的，另外有些语言特性里面规定的错误，在语法分析的阶段就进行了报错，因此也就不需要在本阶段进行处理了。

二、遇到的问题

主要还是避免连环报错的问题，最直接的还是体现在 exprerror.decaf 测例里面，对于 $a = \text{Main} + 1$ 的处理，这里肯定是不兼容的，如果这里把该表达式的 type 赋为 left.type，那么后面就会对 $a > d$ 再次报错，所以如果在表达式判断不兼容时，需要置为 Error，而兼容时置为其 type。