

PA1-A report

大致思路

抽象类/函数

添加了 abstract 关键词。

在 ClassDef 类中添加了成员变量 modifiers。

在 Modifiers 类中添加了 abstract 选项。

把一个类的 modifiers 的 abstract 选项选上就会变成抽象类。

把一个函数的 modifiers 的 abstract 选项选上就会变成抽象函数。

局部类型推导

添加了 var 关键词。

把 LocalVarDef 类中的 typeLit 成员变量的类型由 TypeLit 改成了 Optional<TypeLit>。用 null 来表示未确定的类型。

函数类型

添加了函数类型。

新增了一个类 TLambda 表示函数类型。

TLambda 类中的 returnType 表示返回值类型，params 表示参数。

Lambda 表达式

新增了关键词 fun 和操作符 '=>'。

新增了类 LambdaDef 用来表示一个 Lambda 表达式。

LambdaDef 类中的 params 表示参数，type 表示这个 Lambda 表达式是只包含一个表达式还是包含一个块。后面的 expr 和 block 就是表达式和块。

函数调用

修改了 Call 类以支持函数类型。

Call 类只包含两个成员变量，expr 表示调用的表达式，args 表示参数。

遇到的问题

经常拼错变量名/漏打符号，要观察很久才能发现。

要求回答的问题

Q1

A 表示一个实际的类型（例如二元运算、lambda 表达式）；B 表示一个抽象的类型（例如表达式），包含了各种 A 类型。

Q2

框架中给 ELSE 和 empty 设置了不同的优先级，ELSE 的优先级更高，所以在遇到移进/规约冲突的时候，会优先移进，即每个 ELSE 子句会与最近的 IF 匹配。

Q3

实际的框架中直接用 yacc 在单词流上进行分析构建出了抽象语法树。具体语法树是在分析时被隐式构造出来的，并在分析中被转换成了抽象语法树。

参考资料

戴言同学告诉了我可以用 modifier 实现 abstract，具体的方法和细节时我自己思考的。

参考了一些文档

通过加入优先级避免移进/规约冲突：https://www.gnu.org/software/bison/manual/html_node/Non-Operators.html#Non-Operators

Optional<T> 介绍：<https://segmentfault.com/a/1190000008692522>