PA1-A 实验报告

2018011289 包涵

实验简述

- 思路：阅读理解代码，模仿框架的实现方式，将新语法的EBNF翻译成代码。
- 翻译过程
    - 抽象类
        - 修改了syntax/ast.rs中的`ClassDef`，加入了标志是否是抽象类的`abstract_: bool`量；对应修改了`print/ast.rs`中`ClassDef`打印方式，当`abstract_`为`true`时，打印"ABSTRACT"
        - 修改了`syntax/ast.rs`中的`FuncDef`，加入了`abstract_: bool`，把`body`改为了`Option<Block<'a>>`；对应修改了print的方式
    - 局部类型推断
        - 在ty.rs中给`SynTyKind`增加了`None`的类型，并在打印时输出对应字符
    - First-class Functions
        - 增加新type：在ty.rs中给`SynTyKind`增加了Lambda类型，并对应在print/ast.rs中进行打印
        - 表达式：在syntax/ast.rs中给`ExprKind`增加了`Lambda`类型，增加了`Lambda`和`LambdaBody`两个`struct`；Lambda在print/ast.rs中仿照其他expr进行打印，但是lambda表达式有两种function body，这通过`LambdaBody`的`Printable`来区分；如果`LambdaBody`中的`expr: Option<Box<Expr<'a>>>`是`None`，则输出其中`body: Option<Block<'a>>`的内容，否则输出`expr`的内容
        - 调用：直接修改产生式和对应fn即可
- 挑战和解决方法
    - 一开始用了整体框架，但是修改FuncDef时，其他模块会报错，完全不知道该怎么改，后来换了PA1-A的专用代码，就好了

问题回答

1. (Rust) 有一部分 AST 结点是枚举类型。若结点 B 是枚举类型，结点 A 是它的一个 variant，那么语法上会不会 A 和 B 有什么关系？

   答：文法上，节点A应该是节点B生成的。

2. 原有框架是如何解决空悬 else (dangling-else) 问题的？

   答：原有框架通过将else关键字的优先级设为最高，保证了if语句总是先结合最近的else语句，进而消除了歧义。

3. PA1-A的流程是：输入程序 lex 完得到一个终结符序列，然后构建出具体语法树，最后从具体语法树构建抽象语法树。 这个概念模型与框架的实现有什么区别？

   答：框架的实现中，rule都是具体语法树的生成规则，使用产生式进行匹配的过程，是程序获得CST的过程。但是并没有存储具体语法树，而是在rule对应的functions中进行了处理，直接生成了AST的节点。