201220180 李全昊 FLA\_LAB实验报告

一. 实验完成度

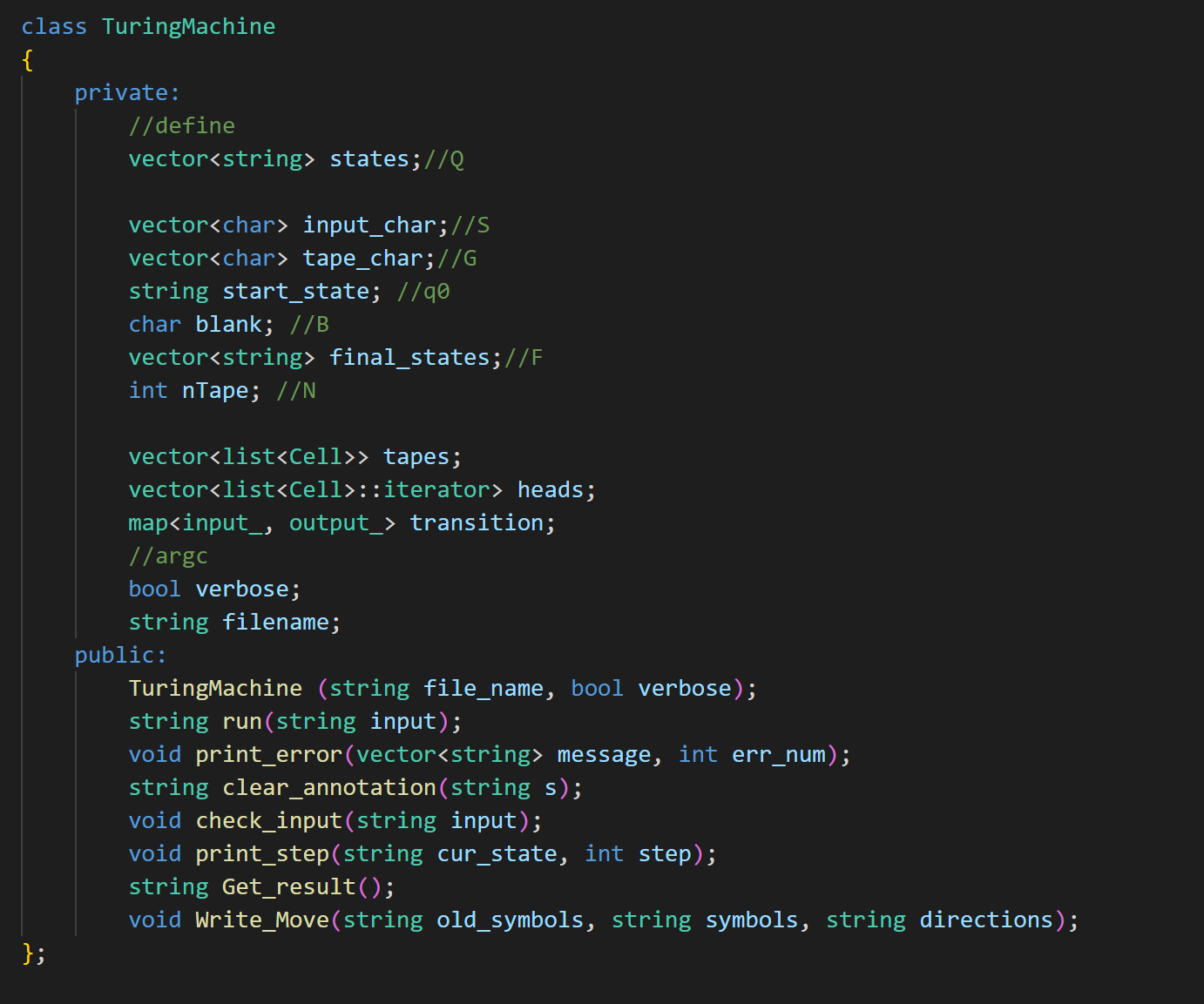
完成了三个阶段的所有任务

二. 分析与设计思路

总体步骤是按照stage1-3来写代码的；

1. stage1要求我实现一个多带图灵机程序解析器；在这个阶段主要对应的是我的TuringMachine类的初始化函数；

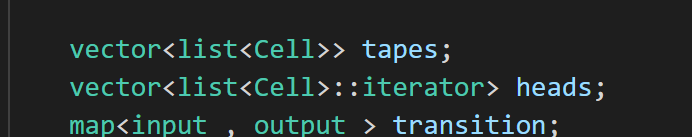
我的数据结构构造如下图所示：



在初始化函数中进行的主要是读入tm文件进行的字符串处理的操作；读入字符串并且初始化我的图灵机类的成员Q, S, G, q0等。在处理字符串的过程中如果碰到了相对应的错误会调用prit\_error函数进行错误的输出（比如文件打不开等错误）；

这里的verbose成员用来指示当前是否为verbose状态，如果是的话会进行更加详细的输出。

2. stage2要求我们的图灵机程序经过解析器解析后，将得到一个对应的多带图灵机模拟器。模拟器读入一个字符串(即命令行参数中的<input>) 作为图灵机的输入。在这个阶段主要对应的是我的run(string input)函数。在这个函数里我们接收命令行传来的input参数，然后进行图灵机的运行；整体运行的过程其实就是由tm文件指定的transition fuction来决定的。

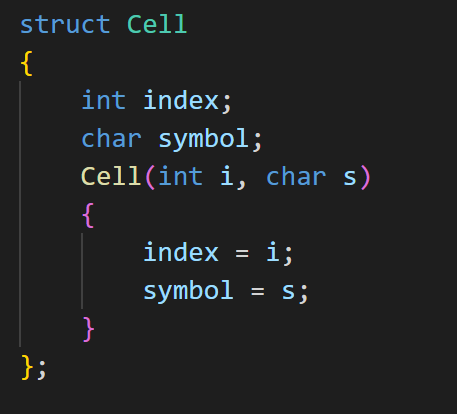


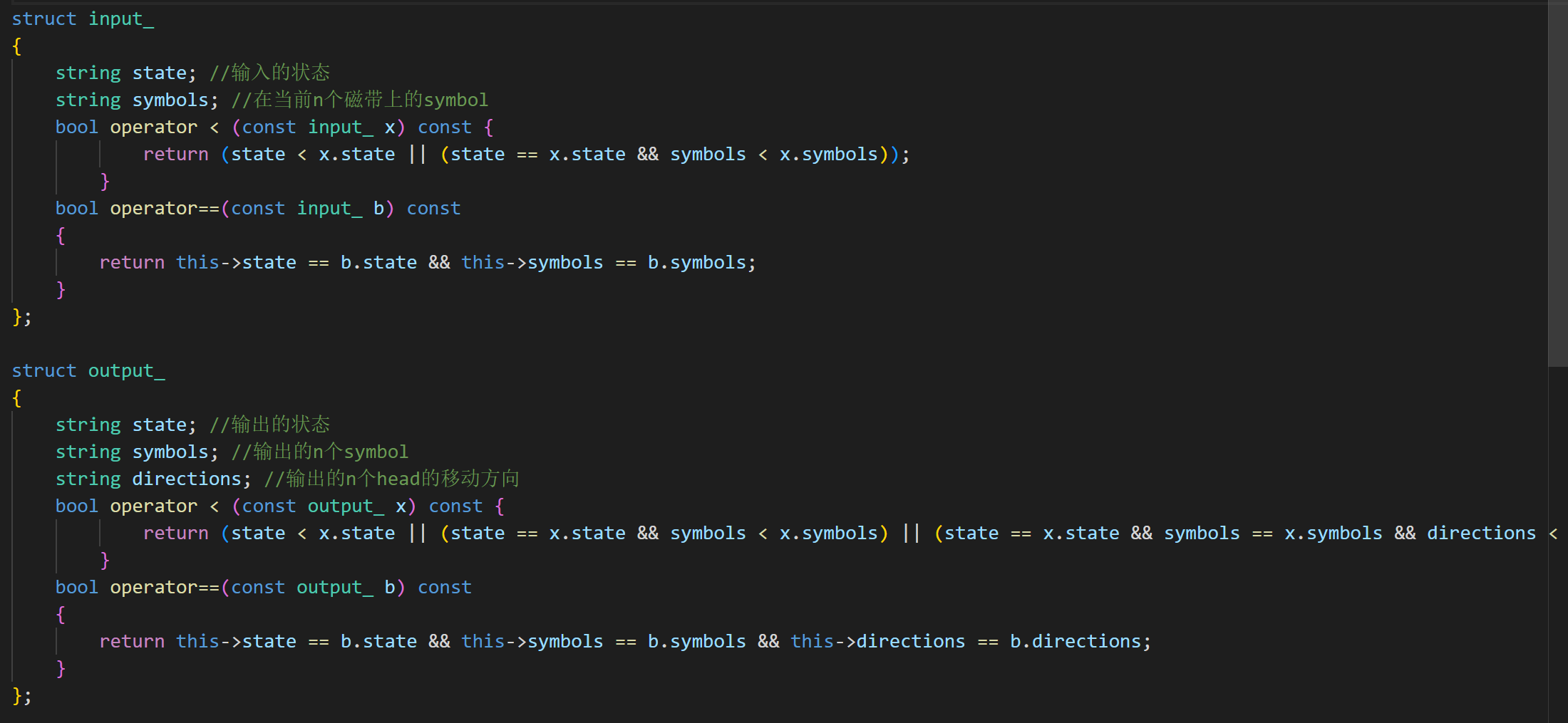
在这里我把每个tape作为一个list，因为他是一个双向的链表，符合我们对纸带的理解。每一个纸带上的单元是一个结构体Cell;里面包含有对应的index和char类型的元素。

Tapes是包含了多个tape的vector;

同理每个head是每个tape对应的磁头；

而transition存有我们tm文件指定的所有转移函数，是一个map结构；他的key value分别是我自己定义的结构体input\_和output\_。





要特别注意的是这里作为key的input\_结构体必须要好好写他的小于号重载；因为这在map中是非常重要的，用来排序key和比较key！我当时debug了很久。

3. stage3要求我们自己构思两个图灵机；并且编写对应的tm文件；这本身的思路我就不在此介绍了；值得一提的是，在测试的时候发现我没有对\*号的匹配进行处理，因此在思考之后，我决定对于每一个规则，我都去试试看当前的head底下的symbol是否可以通过对于\*的转换满足，如果不行那就彻底不行了（在此之前已经对普通的规则寻找过了）。

还需要指出的是\*不能匹配blank!所以我们应该在循环的过程中，如果发现不能匹配，而且有一个head底下是blank，那就彻底不能匹配了。

三. 总结感想

这次试验难度不是很大，但是比较繁琐，在配环境的时候也遇到了一些问题，在询问了助教gg之后得到了解决！Windows下build得到的是sln文件要用VS去编译；Linux下直接得到makefile ; make就行了。