FLA-LAB 实验报告
——201220176 刘兴

一、*实验完成度：100%(大概？)*

1. 完成了解析器，并且可以指定 --v|--verbose 参数。
2. 完成了模拟器，实现了普通模式和 verbose 模式。
3. 完成了两个图灵机程序。

二、*设计思路*

1. 命令解析：

只需要对传进来的参数进行相应的解析即可。通过 parseCommand 函数进行解析，若未发现错误，则返回解析出来的图灵机程序文件名和输入。

2. 图灵程序解析：

实现了一个 TuringMachine 类。其私有成员主要包括：

```
bool verbose;
    string input;
    int numOfTape;
    unordered_set<string> state;     //状态集 Q
    unordered_set<string> finalState;    //接收状态集 F
    unordered_set<string> inputChar;     //输入符号集 S
    unordered_set<string> tapeChar;      //纸袋符号集 G
    string startState;  //开始状态
    string curState;     //当前状态
    unordered_map<pair<int,int>,char,pair_hash> tape;      //根据(磁道数,
下标)来读取符号
    vector<int>head;    //磁头指向下标
    int step;    //记录移动步数
    vector<pair<int,int>> limit;     //维护所有磁道的左右边界,左闭右开，如
(0,0)代表全空。注意会受到 head 的影响
    unordered_map<pair<string,string>,vector<string>,pair_hash>
transferFunc;    //转移函数,<当前状态，当前符号组>为 key,<新符号组，方向组，新
状态>为 value.
```

通过对传入的图灵程序进行解析，将相关信息存储在 TuringMachine 的成员中，有关图灵程序的解析便结束。

3. 图灵机的运行：

根据当前状态和当前符号组，寻找到相应转移函数，然后更新状态、磁带符号以及磁头移动情况即可。

```
while(true)
    {
        string curTapestr = getTapestr(head);//获得当前符号组
        //遍历状态转移函数
        auto itr =transferFunc.begin();
```

```
        while(itr!=transferFunc.end())
        {
            if(itr->first.first!=curState)
            {
                ++itr;
                continue;
            }
            if(checkMatch(curTapestr,itr->first.second))
                break;
            ++itr;
        }
        //若未找到对应 move，则停止
        if(itr==transferFunc.end())
            {
                printResult();
                return;
            }
        //修改当前状态，并对磁头进行读写
        curState=itr->second[2];
        wirteAndmove(itr->second);
        ++step;
        if(verbose)
            printCurStep();
    }
```

4．图灵程序的编写：

（1）第一题思路：

①使用两磁带图灵机，将输入拷贝至第二磁带上

②按照要求先将第二磁带的末端，拷贝到第一磁带的首位

③接下来将第二磁带按顺序拷贝到第一磁带上即可

（2）第二题思路

①使用三磁带图灵机。

②第二磁带和第三磁带同时增加，每次最多增加 1 位。

③通过第二磁带和第三磁带实现一个逐位的乘法（，第三磁头和第一磁头同时移动，第三磁带扫描完一遍，第二磁头移动一格）。

④若三个磁头同时消耗完成，则返回 true;若磁带 2、3 消耗完而磁带 1 仍有剩余，则进行步骤 2；其他情况返回 false.

三、遇到的问题与解决方案

此次实验较为简单，并未遇到什么特别难的问题。

1．最为繁琐的地方，莫过于对图灵程序的解析。为了维护程序的鲁棒性，需要考虑各种各样的输入错误，这花费了较多的时间。

2．数据结构的设计方面，有 STL 的帮助，通过 set 和 map 很好实现。

3．打印的时候需要保证格式的美观，最初读头的移动与边界很难协调，后面想

明白后，干脆弄了个变量 limit 来存储每条磁带的边界，然后专门在有关读写头的操作里进行维护。

四、总结感想

这次实验总体来说，难度不算很大，但是非常繁琐。因为要完成各种模式的匹配，同时还需要维护程序的鲁棒性，要考虑各种出错情况，在这些细碎的地方花费了大量的时间。

五、意见与建议

无。