

图灵机实验报告

盛朱恒 MF20330066

1、分析与设计思路

代码主体为三个类，主要变量与方法如下：

一个Delta类的对象是一个转移函数，一个Tape类的对象是一条纸带，一个TuringMachine类的对象是一个图灵机。

```
class Delta{
public:
    string old_states;
    string old_signs;
    string new_signs;
    string directions;
    string new_states;
    void Print() const {
        ...
    }
    bool operator < (const Delta &b) const{
        ...
    }
};
```

```
class Tape{
public:
    map<int,char> tape;
    int left,right,pointer;
    char B;
    Tape(char b){
        left = right = pointer = 0;
        B = b;
    }
    void Left(int l){
        if(l < left) left = l;
    }
    void Right(int r){
        if(r > right) right = r;
    }
    void PrintText(){
        ...
    }
    void Print(int index){
        ...
    }
};
```

```
class TuringMachine{
private:
    bool verbose;
    set<string> Q;
```

```

set<char> S;
set<char> G;
string q0;
char B;
set<string> F;
int N;
set<Delta> D;
//-----//
vector<Tape> Tapes;
string State;

bool CheckinQ(string s){
    ...
}
bool CheckinS(char s){
    ...
}
bool CheckingG(char s){
    ...
}
bool CheckinF(string s){
    ...
}
void PrintParser(){
    ...
}
void PrintTapes(int step, string state){
    ...
}
int Run(){
    ...
}
public:
    TuringMachine(char *tm_filename, bool v){
        ...
    }
    void CheckParser(){
        ...
    }
    void CheckInput(char *input){
        ...
    }
    void Input(char *input){
        ...
    }
};

```

1) 解析器

解析部分采用字符串处理的方法，由于字符串输入可能性过多，因此假设输入的文件格式为标准的语法，语法不会出错，只有图灵机本身可能定义出错。

首先，解析文本，得到**状态集 Q**，**输入符号集 S**，**纸带符号集 G**，**初始状态 q0**，**空格符号 B**，**终结状态集 F**，**纸带数 N**，**转移函数 delta**。这一部分在TuringMachine的构造函数中完成。

接下来，检查解析是否正确，包括：**S**是否是**G**的子集，**F**是否是**Q**的子集，**q0**是否属于**Q**，**B**是否属于**G**，对每个**delta** ("**<旧状态>** **<旧符号组>** **<新符号组>** **<方向组>** **<新状态>**")，**<旧状态>**以及**<新状态>**是否属于**Q**，**<旧符号组>**以及**<新符号组>**中的符号是否属于**G**，**<方向组>**是否是"**l,r,***"之一，**<旧符号组>**、**<新符号组>**以及**<方向组>**的长度是否为**N**，各个集合中是否有重复元素，各个**delta**是否有冲突。

2) 模拟器

模拟器部分的步骤如下：

首先，读取输入串，检查是否符合规定，即字符是否属于**S**。

然后，初始化纸带，将输入串复制到第一条纸带上。

最后，运行图灵机，直至停机。

运行时，根据当前状态，查找**delta**函数，进行对应操作。

核心部分纸带采用map记录，根据index即可得到对应字符，index可为负值。同时辅以记录曾经记录东西的最左端与最右端，解决纸带的字符中间有空格符号出现的情况。

3) 图灵机程序

case1

$$L_1 = \{a^i b^j a^i b^j | i, j > 0\}$$

思路：

采用两条纸带。开始时，第一条纸带上为 $a^i b^j a^i b^j$ 。然后，将第一条纸带上的前半部分剪切到第二条纸带上。此时，两条纸带都有 $a^i b^j$ ，且都指着开头。然后，开始对比两条纸带上的字符，一样的就删除。最后，两条纸带应该都空，就是对的。其他的任何情况，都是出错。

如果是对的，此时纸带1是空的，写上true即可。

如果是错的，首先需要清空纸带1，然后再写上false。

case2

$$L_2 = \{1^m \times 1^n = 1^{mn} | m, n \in N^+\}$$

思路：

采用3条纸带。开始时，第1条纸带上为 $1^m \times 1^n = 1^{mn}$ 。然后，将第1串1剪切到第2条纸带，将第2串1剪切到第3条纸带。处理完后，三条纸带上分别是 1^{mn} , 1^m , 1^n 。然后，开始循环，第三条纸带从左往右或者从右往左遍历，每次遍历一轮，第二条纸带删除一个1，第一条纸带则是每移动一下，删除一个1。最后，3条纸带应该都空，就是对的。其他的任何情况，都是出错。

如果是对的，此时纸带1是空的，写上true即可。

如果是错的，首先需要清空纸带1，然后再写上false。

2、实验完成度

全部完成。开启-v后，可以输出解析器的错误信息以及模拟器的瞬时描述。

case1

```
szh@ubuntu:~/Documents/TuringProject$ ./turing programs/case1.tm aaabbbbaaabb  
true
```

```
szh@ubuntu:~/Documents/TuringProject$ ./turing programs/case1.tm aabbba  
false
```

case2

```
szh@ubuntu:~/Documents/TuringProject$ ./turing programs/case2.tm 11x111=111111  
true
```

```
szh@ubuntu:~/Documents/TuringProject$ ./turing programs/case2.tm 11x111=1111  
false
```

```
szh@ubuntu:~/Documents/TuringProject$ ./turing programs/case2.tm 11x11x111=  
false
```

verbose

```
Step    : 21  
Index0  : 6 7 8 9 10 11  
Tape0   : f a l s e  
Head0   :  
Index1  : 1 2 3 4  
Tape1   : _ b b b  
Head1   : ^  
State   :halt_reject  
-----  
Result: false  
===== END =====
```

error

```
szh@ubuntu:~/Documents/TuringProject$ ./turing programs/case2.tm 11x11x111=2  
illegal input
```

```
szh@ubuntu:~/Documents/TuringProject$ ./turing -v programs/case2.tm 11x11x111=2  
Input: 11x11x111=2  
===== ERR =====  
error: '2' was not declared in the set of input symbols  
Input: 11x11x111=2  
          ^  
===== END =====
```

3、实验中遇到的问题及解决方案

1) 寻找合适的纸带数据结构

使用了map，在性能上不如线性数据结构，但是对负值的index可以很好地适应。

2) 输入为空串时可能会导致问题

在模拟器初始化时，空串作特殊处理。纸带map中没有的index默认对应的字符为空格B。

3) 纸带的指针可能会指向空格B，可能会打印不到

修改为，不论如何指针部分需要打印出来。

4、总结感想

这个大作业实验很有意思。

5、对课程和实验的意见与建议

各个课程的大作业时间可以错开一点吗。。。