

课程名称：形式语言与自动机

学院	计算机科学与技术	专业（方向）	计算机科学与技术
学号	201220215	姓名	田馥雯
Email	tfuwen@qq.com	实验名称	FLA (Fall 2022) — Project

分析与设计思路

解析器

通过如下语句打开提供的.tm文件

```
1     ifstream input;
2     input.open(TM);
```

然后 `getline(input, tmp)` 逐行获取文件内容，利用 `erase()`、`find()`、`substr()` 等处理string对象的函数，将其存储在如下数据结构中

```
1  vector<string> Q;           // 状态集 Q
2  vector<char> S;             // 输入符号集 S
3  vector<char> G;             // 纸带符号集 G
4  string q0;                  // 初始状态 q0
5  char B;                     // 空格符号 B
6  vector<string> F;           // 终结状态集 F
7  int N;                      // 纸带数 N
8  vector<transition> delta; // 转移函数 delta
```

其中 `transition` 结构体包含五个部分

```
1  struct transition
2  {
3      string old_state;
4      string old_symbol_set;
5      string new_symbol_set;
6      string direction_set;
7      string new_state;
8  }; // <旧状态> <旧符号组> <新符号组> <方向组> <新状态>
```

并且在verbose模式中命名了不同类型的错误，例如

```

1 cerr << "TM file not exists" << endl;
2 cerr << "TM file incorrect format" << endl;
3 cerr << "States incorrect format" << endl;
4 cerr << "Input symbols incorrect format" << endl;
5 cerr << "Tape symbols incorrect format" << endl;
6 .....

```

需要注意的是注释和空行的解析中，我的程序默认注释符号;前必有空格符

```

1  /* 解析器 */
2  void parser(string TM);
3  void parser_Q(string input);
4  void parser_S(string input);
5  void parser_G(string input);
6  void parser_q0(string input);
7  void parser_B(string input);
8  void parser_F(string input);
9  void parser_N(string input);
10 void parser_delta(string input);
11 void handle_line_comment(string &input);

```

模拟器

定义如下数据结构

```

1 int step; // 步骤数
2 string state; // 当前状态
3 vector<int> head_index; // 读写头索引
4 vector<int> left_index; // 最左非空格符号索引
5 vector<int> right_index; // 最右非空格符号索引 + 1
6 vector<map<int, char>> tape; // 纸带（索引，符号）

```

通过当前状态和读写头对应的纸带内容在delta中寻找合适的转移函数，若未找到则break然后停机，否则修改纸带内容，移动读写头并继续寻找下一个转移函数。

在verbose模式中，要将每一步的瞬时描述打印出来

```

1  /* 模拟器 */
2  void simulator(string input);
3  void check_input_string(string input);
4  void print_instant_description();
5  int symbol_compare(string a, string b);

```

多带图灵机程序

case1

利用单磁带TM，先将除最后一个字符以外的01串进行右移，再将读写头左移到第一个字符处填入最后一个字符

case2

由除0以外的平方数必可以拆成奇数的和，例如

$$\begin{aligned}
 1 &= 1 \\
 4 &= 1 + 3 \\
 9 &= 1 + 3 + 5
 \end{aligned}$$

利用二磁带TM，当输入为空串时，直接跳转到accept状态，否则：

- 第一道磁带上的字符保持不变，第二道磁带上的字符初始化为1
- 两道磁带的指针头持续右移直到
 - 同时指向空格符：跳转到accept状态
 - 第一道磁带指向空格符：跳转到reject状态
 - 第二道磁带指向空格符：在第二道磁带上再写入两个1，并把读写头移动到第一个1的位置后，重复持续右移操作

accept状态和reject状态的写入true or false的操作与示例tm几乎相同

实验完成度

完成了多带图灵机程序解析器、多带图灵机模拟器（普通模式和 `verbose` 模式）以及两个图灵机程序

实验中遇到的问题及解决方案

如何判断纸带应该被打印出来的范围？

直接用

```
1 vector<int> left_index;      // 最左非空格符号索引
2 vector<int> right_index;     // 最右非空格符号索引 + 1
```

两个向量存储来实现注4和注5，并在读写头变化时对其修改

case2的空串问题

如果输入为空要初始化tape0为_，刚开始没管这个特殊情况

对齐问题

```
Step      : 203
State     : halt_reject
Index0    : 0 1 2 3 4
Tape0     : f a l s e
Head0     :           ^
Index1    : 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
Tape1     : 1 1 1 _ 1 1 1 1 1 1 1 1 1 1 1
Head1     :           ^
-----
Result: false
===== END =====
```

计算index位数，打印多余空格即可

```
1 int cnt = 1; // index位数，用于对齐
2 int num = j;
3 while (num /= 10)
4 {
5     cnt++;
6 }
7 print_spaces(cnt - 1);
```

对齐后如下图所示

```

-----
Step      : 193
State     : reject
Index0    : 0 1 2 3
Tape0     : 1 1 1 1
Head0     :           ^
Index1    : 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
Tape1     : 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Head1     :           ^
-----

```

总结感想

实验不难，但是自己的正确性不能判断，就把能增加的鲁棒性增加了，其他的也不太清楚，有点没底

对课程和实验的意见与建议

无