

形式语言与自动机实验报告

姓名：张玄逸

学号：201220194

邮箱：1822771416@qq.com

时间：2022年12月22日

完成度

完成了所有任务

设计思路

- 边与点

要构建一个图灵机，最基本的就是搭建好它的状态转移图，所以首先创建结构体。

```
struct Edge
{
    vector<char> old_letter; // 现在指向的字符
    vector<char> new_letter; // 将要写下的字符
    vector<char> new_step;   // 移动方向
    State *new_state;
    Edge *next = NULL;
};

struct State
{
    char name[50];
    Edge *edge_head = NULL; // 链式前向星
    bool final = false;     // 标记是否为终结状态
};
```

边与点的构建和图论常用算法类似，在读入所有状态名时就创建好相应的状态（点），之后的所有转移（边）都加入对应起点的边集合中。

- 读入

然后考虑读入，需要读入8个部分，其中转移函数在读入后直接添加到对应点中，而终结状态直接找到对应状态将其final改为true即可。ASCII可见字符编号为32-126，共95个字符。

```
vector<State *> state_set; // 状态集
bool in_letters[256];    // 输入字符集，bool表示是否在其中
bool all_letters[256];   // 字符集
State *start;            // 初始状态
char space_letter;       // 空格符号
int tape_num;            // 纸带数
```

读入函数使用fscanf，除纸带数以外都是单个字符依次读入。除转移函数外，其它7个的读入逻辑类似：忽略注释，读入'#'，标志字符（如Q，S），'='，若有多个则还需读入左括号和右括号，再依次读取其中的每个字符或字符串，注意当中的符号合法性判别。

- 运行

通过tm文件构建好这个图灵机后就要开始运行。首先设置所有的纸带头指向0号，为初始位置填写空格符号。然后将输入的字符串誊写到0号纸带上，最左端对齐0号位置。这里因为纸带可以向两端无限延伸，我使用了vector来模拟，奇数位代表纸带原本的负数位，偶数位代表原本的正数位，0号就对应0号。即：

vector: 0 1 2 3 4 5

纸带位置: 0 -1 1 -2 2 -3

再写好关于纸带头移动的函数move_left和move_right，因为是不定长数组，一开始只有0号位置，所以在纸带头移动后需要及时将数组长度补充至包含纸带头为止。

从初始状态开始进行while循环，直到没有可用转移时停机。每次在当前状态中找到符合纸带当前符号的边，再按照转移思路改变当前符号，移动纸带头，将当前状态更改为新状态。

输出时首先将左右两端设置为纸带的两端，再排除两端的空格符号即可。

运行效果

- 普通模式

```
user@compiler:~/mine/turing/build$ ./../bin/turing --help
usage: turing [-v|--verbose] [-h | --help]<tm><input>
user@compiler:~/mine/turing/build$ ./../bin/turing ../programs/case2.tm 12
illegal input
user@compiler:~/mine/turing/build$ ./../bin/turing ../programs/case2.tm 1
true
user@compiler:~/mine/turing/build$ ./../bin/turing ../programs/case2.tm 111
false
user@compiler:~/mine/turing/build$ ./../bin/turing ../programs/case2.tm 1111
true
user@compiler:~/mine/turing/build$
```

- verbose模式

```
user@compiler:~/mine/turing/build$ ./../bin/turing --verbose ../programs/case2.t
m 111A11
Input: 111A11
===== ERR =====
error: 'A' was not declared in the set of input symbols
Input: 111A11
      ^
===== END =====
user@compiler:~/mine/turing/build$
```

```

user@compiler:~/mine/turing/build$ ../../bin/turing --verbose ../programs/case1.t
m 11100
Input: 11100
===== RUN =====
Step   : 0
State  : 0
Index0 : 0 1 2 3 4
Tape0  : 1 1 1 0 0
Head0  : ^
Index1 : 0
Tape1  : _
Head1  : ^
-----
Step   : 1
State  : copy
Index0 : 0 1 2 3 4
Tape0  : 1 1 1 0 0
Head0  : ^
Index1 : 1
Tape1  : _
Head1  : ^
-----
Step   : 24
State  : fresh
Index0 : 0 1 2 3 4 5
Tape0  : 0 1 1 1 0 _
Head0  : ^
Index1 : 1 2 3 4 5
Tape1  : 1 1 1 0 0
Head1  : ^
-----
Step   : 25
State  : halt_accept
Index0 : 0 1 2 3 4 5
Tape0  : 0 1 1 1 0 _
Head0  : ^
Index1 : 1 2 3 4 5
Tape1  : 1 1 1 0 0
Head1  : ^
-----
Result: 01110
===== END =====
user@compiler:~/mine/turing/build$

```

遇到的问题及解决

- 文件结束的判定

之前一直以为feof就是直接判断FILE*p指针是否已经到了文件末尾，结果出现了错误。原因是feof其实判定的是最后一次fscanf读到的是不是文件末尾，也就是上一次读取是否失败。

- 读取下一行

之前我使用了一个函数skip_char(x)，也就是忽略之后的所有符号直到遇见x（注释会除去），如果一直到文件末尾都没有遇见则报错。此函数用于比如读入"#Q"之后还要读入一个 '=' 和 '{' 的时候。但在读入转移函数时不能使用skip_char('\n')，虽然每个转移函数占据一行，但最后一个转移之后如果没有注释则可能不会有换行符，出错。新创建函数try_char，逻辑大致与skip_char相同，只是如果到文件末尾也没有找到符号x，则返回false而不是直接报错。

意见与建议

- 希望不要有语法错误

虽然后面添加了FAQ，说明了一些必要的和不必要考虑的语法错误，但注意到“如在必须的位置缺少 '{'、'}'、','、'#' 等情况”里面也还是有等这个字。仍然不能明确分辨到底有哪些语法错误，比如光是“缺少某个符号”这一点，就可以还有缺 '='，缺 'Q' 或 'S'，甚至是缺换行符。另外还有“纸带数是否可能输入0或负数”“转移函数是否可能和纸带数不对应”“结束时true/false是否应该从纸带0号位开始”“结束时纸带头位置有无要求”等等之类的问题。

我也看到助教在群里回答了很多同学的一些语法疑问，但既然同学需要在群里问，就说明手册并没有把语法错误规定清楚。有些人没来得及看聊天记录也不知道有这些答疑，光按自己理解来做就很可能和助教想的不一樣。

助教在群里说“如果不想费心，字符串处理时遇到了任何意料外的情况都可以直接报个syntax error 就一走了之”，但问题不是发现错误如何处理，而是到底该如何发现错误。这几乎就是让同学自己猜测所有的语法错误，那就比编译原理还编译原理了。

哪怕是专门处理语法错误的编译原理，里面出现的错误类型都是按123清楚列好的，不会超出范围（如下图）。如果自动机一定要考虑语法错误，希望可以直接列出有且仅有有哪些语法错误，如果像这次一样只是说哪些有哪些没有，那基本上不可能考虑得全面，毕竟语法错误几乎无穷无尽，助教一直打补丁也挺累的。

以上假设1至7也可视为要求，违反即会导致各种语义错误，不过我们只对后面讨论的17种错误类型进行考察。此外，你可以安全地假设输入文件中不包含注释、八进制数、十六进制数、以及指数形式的浮点数，也不包含任何词法或语法错误（除了特别说明的针对选做要求的测试）。

你的程序需要对输入文件进行语义分析（输入文件中可能包含函数、结构体、一维和高维数组）并检查如下类型的错误：

- 1) 错误类型1：变量在使用时未经定义。

45

-
- 2) 错误类型2：函数在调用时未经定义。

- 3) 错误类型3：变量出现重复定义，或变量与前面定义过的结构体名字重复。

- 4) 错误类型4：函数出现重复定义（即同样的函数名出现了不止一次定义）。

- 5) 错误类型5：赋值号两边的表达式类型不匹配。