

形式语言与自动机实验实验报告

191810140沈炫辰

1.分析设计思路

实验需要完成一个接受一定语法的输入的图灵机来处理一定的输入纸带，整个图灵机因此分为了两部分：处理图灵机程序输入的部分和处理输入纸带的部分。

对于处理图灵机程序输入的部分，定义一个图灵机class来接收程序语法中要求的各种图灵机需要的元素。利用C++中的库来处理文件输入。用string或者char的vector来存储“集合”的内容（允许重复，如果有重复则忽略），用map来存储转移函数delta，其中key为当前的状态，当前磁带头指向的值，value为写入的值，磁带头的移动方向和下一状态。

在图灵机程序内出现'*'时：

- 如果是判断磁带头指向的值，视其为通配符，如果同时有多个匹配，则选择最精确的（即star最少的），如果有匹配star一样多，则无法确定选择了哪一个。

- 如果是写入，则不写入，如果是移动，则不移动。

对于处理输入纸带的部分，先检查纸带的合法性，然后定义当前状态为初状态，然后用while循环不断执行，直到：

- 执行到了一个终结状态，程序结束，输出第一条纸带的结果

- 在当前状态没有对应的转移函数，报错

- 死循环（图灵机程序或者输入的缺陷导致）程序设定，如果程序执行了1000000步仍无结果会强制跳出

纸带规模：

- 虽然纸带无限长，为了方便，每条纸带长度定义为1000000，如果超规模会报错。

2.实验完成度、运行环境等

完成了包括解析器verbose选做部分的所有内容

实验环境：

操作系统：win10系统

编译器：mingw64 x86_64-8.1.0-posix-seh-rt_v6-rev0

代码编辑器：vs code

3.最大公约数图灵机设计思路

```
; solve gcd
; input should be 1+01+

; the finite set of states
#Q = {start, accept, copy, back, setpivot, goright, find_small, go_first,
go_second, delete_0, delete_first, delete_second}

; the finite set of input symbols
#S = {0,1}
```

```

; the complete set of tape symbols
#G = {0,1,_}

; the start state
#q0 = start

; the blank symbol
#B = _

; the set of final states
#F = {accept}

; the number of tapes
#N = 2

; the transition functions

; State start: begin
start ** ** ll setpivot

; State setpivot: set 0 in front of each number
setpivot ** 00 rr goright

; State goright: find the second number
goright 1* ** r* goright
goright 0* _* r* copy

; State copy: copy the second number to the second line
copy 1* _1 rr copy
copy _* ** ll back

; State back: back to the start
back 0* ** *l back
back *0 ** l* back
back ** ** ll back
back 00 ** rr find_small

; State find_small: find the smaller number
find_small ll ** rr find_small
find_small l_ ** rl go_first
find_small _1 ** lr go_second
find_small __ ** l* delete_0

; State go_first: go to the right most of the first number
go_first 1* ** r* go_first
go_first _* ** l* delete_first

; State delete_first: delete the first number
delete_first ll _* ll delete_first
delete_first 10 ** l* back

; State go_second: go to the right most of the second number
go_second *1 ** *r go_second
go_second *_ ** *l delete_second

; State delete_second: delete the second number
delete_second ll *_ ll delete_second
delete_second 01 ** *l back

```

```
; State delete_0: delete the pivot 0
delete_0 1* ** 1* delete_0
delete_0 0* _* r* accept
```

主要思路：辗转相减法

先在两数之前设置一个0标志位，表示两数的开始（方便回到两数开头），然后查找输入纸带的0，并将0变为_，将0以后的1移动到第二条纸带，然后两个磁头都返回两数起点，然后进行循环：找到两数中的较小（短）的（如果一样长，说明结束了，删除第一条纸带的初始的0，得到结果），然后大数磁带的头走向大数的尾，然后大数减去小数，然后双方都回到起点，然后重复。

输入必须是正整数，不能为0（会死循环），必须按格式输入1+01+，否则大概会报错，不报错也没有正确结果。