苏州大学实验报告

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 院系 | 计算机学院 | | 年级专业 | | 21计科 | | 姓名 | 方浩楠 | 学号 | 2127405048 |
| 课程名称 | | 编译原理课程实践 | | | | | | | 成绩 |  |
| 指导教师 | | 王中卿 | | 同组实验者 | | 无 | | 实验日期 | 2023.9.18 | |

# 实验目的

基于MYT算法，实现将正则表达式转换为NFA。需要实现以下正则表达式符号：

a

ab

a|b

(ab)

a\*

(ab)\*

# 实验内容

1. 输入为正则表达式，输出为一个三元组序列，每个三元组为：（起始状态，符号，结束状态）举例：

1）ab

|  |  |  |
| --- | --- | --- |
| **起始状态** | **符号** | **结束状态** |
| 0 | a | 1 |
| 1 | b | 2 |

2）ab\*

|  |  |  |
| --- | --- | --- |
| **起始状态** | **符号** | **结束状态** |
| 0 | a | 1 |
| 1 | b | 1 |

# 实验步骤和结果

本次试验主要需要匹配”\*”,”|”,”(”,”)”四个操作符.

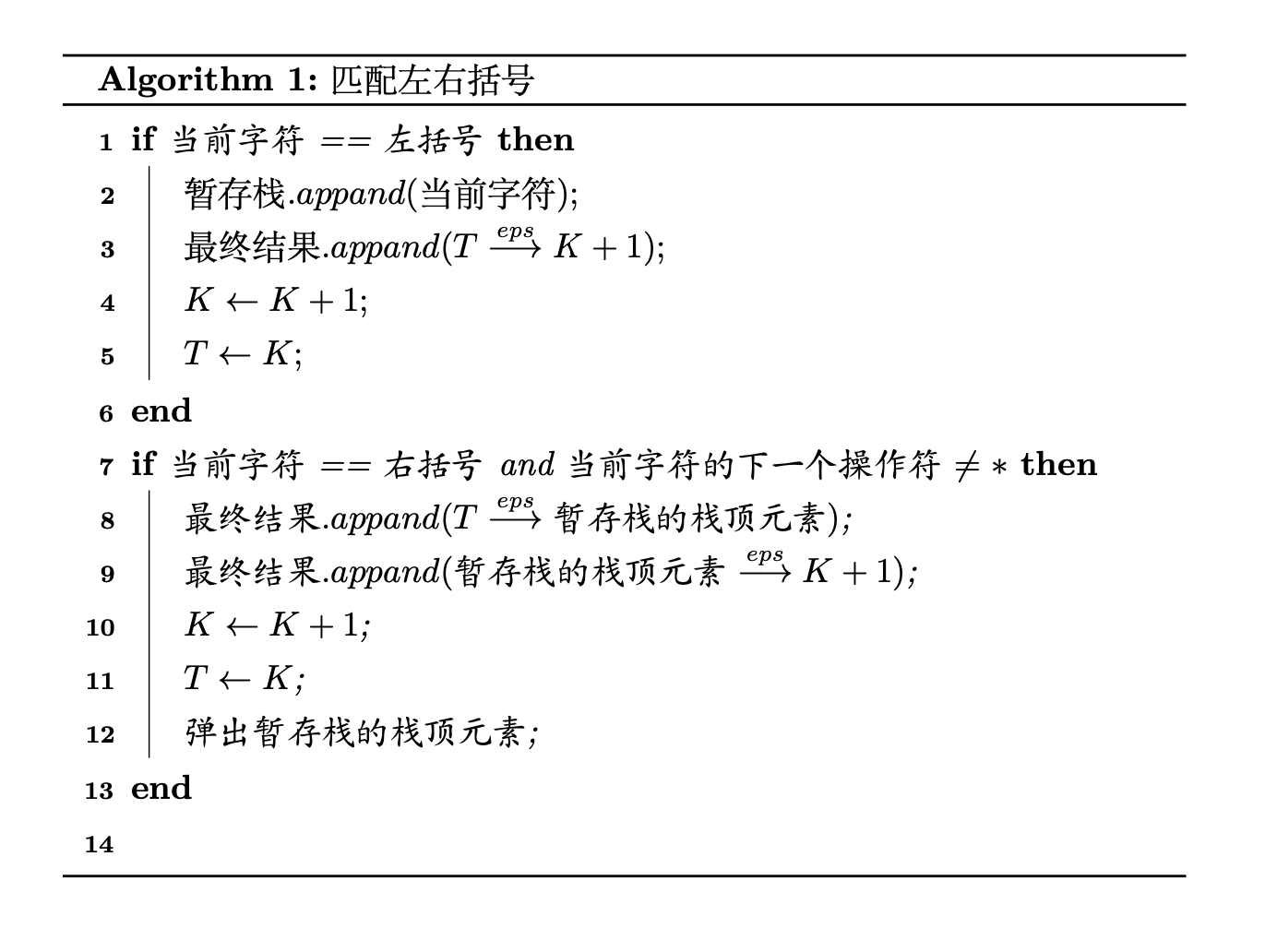
* 1. 匹配左右括号:

思路:

匹配到左括号时,需要将左右括号中的内容开始添加到暂存栈中,同时结果中要添加状态T到状态K+1的转换,转换后K加一,T变为K

若匹配到右括号,则结果中添加状态T到暂存栈中的栈顶元素的状态的转换,同时添加暂存栈的栈顶元素的状态到状态K+1的转换,转换后K加一,T变为K,同时弹出栈顶元素

伪代码:



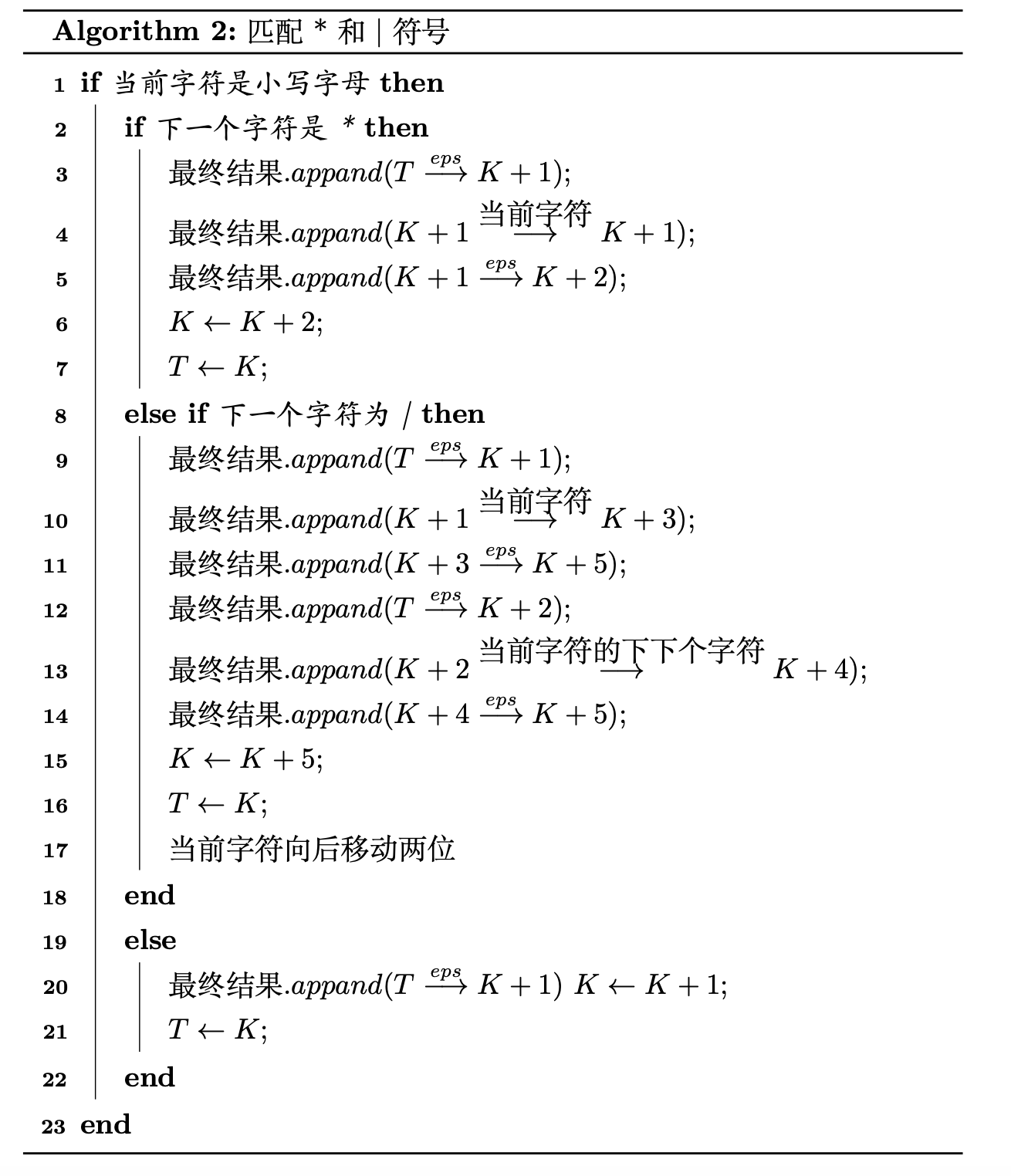
* 1. 匹配\*和|

思路:

匹配到一个字母后,若下一个字符为\*,则根据thompson构造法进行构造

匹配到一个字母后,若下一个字符为|,首先需要获取到符号|后面到的字符,然后根据thompson构造法进行构造,详细过程在伪代码中

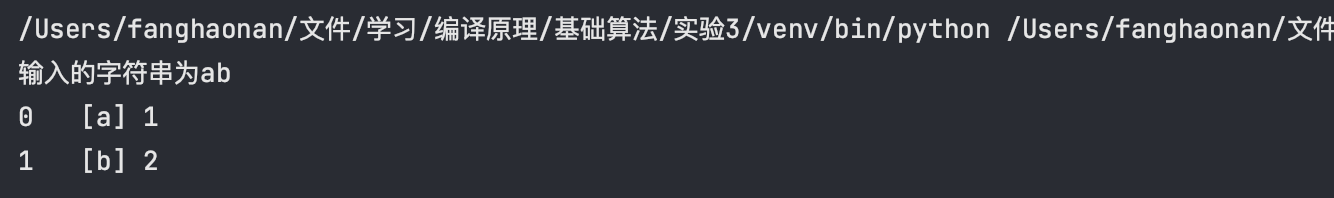
伪代码:



对正则表达式的解析:

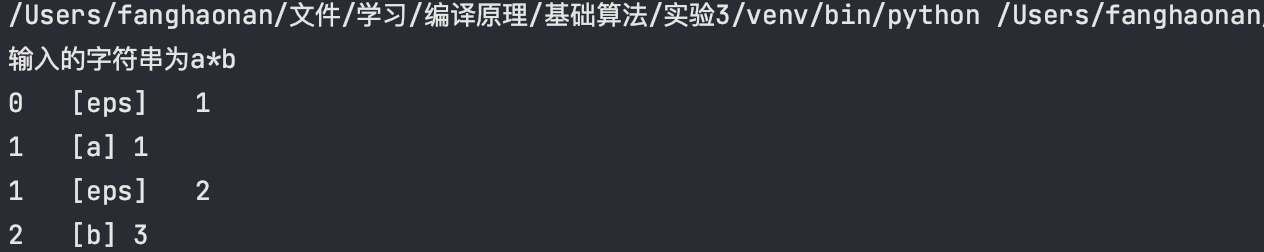
Input:ab

Output:



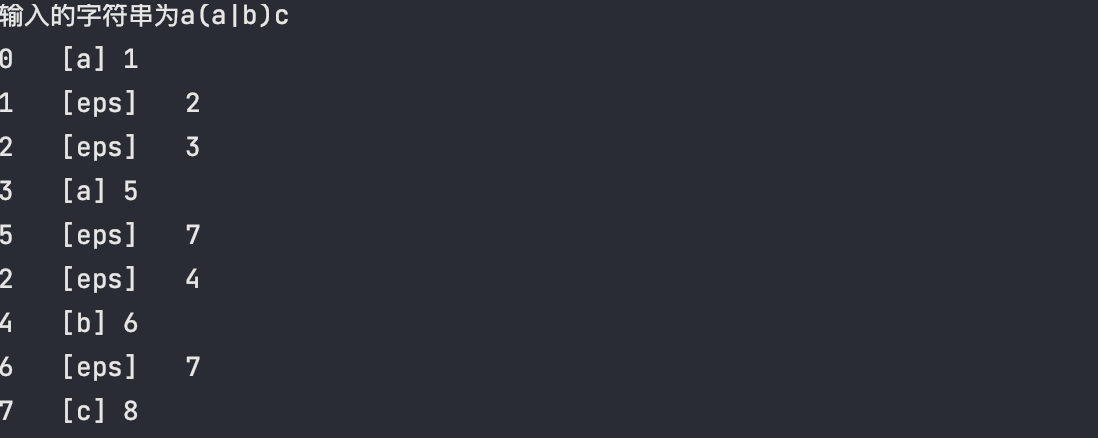
Input:a\*b

Output:



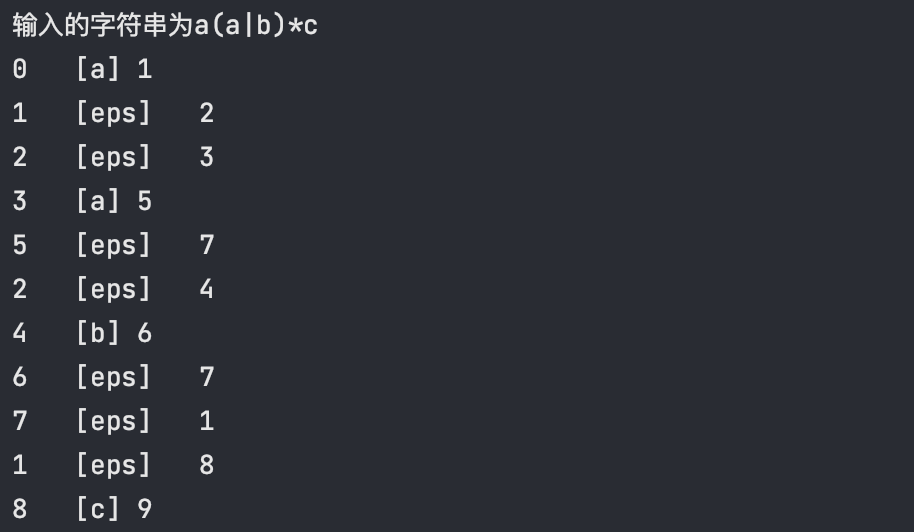
Input: a（a|b）c

Output:



Input:a(a|b)\*c

Output:



# 实验总结

本次试验让我了解到了如何将正则表达式转换为NFA,并且了解了如何匹配正则表达式中的左右括号,星号和或符号

PS:源代码:

*#! /usr/bin/env python*

*# coding=utf-8*

S = [] *# Status*

ST = [] *# temp status*

T = 0

K = 0

input = 'a(a|b)\*c'

n = len(input)

def get\_current(input, i):

return input[i]

def get\_next(input, i):

if i < len(input) - 1:

return input[i + 1]

else:

return ''

def add\_to\_status(t0, ch, t1):

S.append((t0, ch, t1))

def check\_vocab(ch):

if 'a' <= ch <= 'z':

return True

else:

return False

def check\_union(ch):

if i < len(input) - 1:

return input[i + 1] == '|'

else:

return False

i = 0

while i < len(input):

ch = get\_current(input, i)

sym = get\_next(input, i)

if check\_vocab(ch) == True:

if sym == '\*':

add\_to\_status(T, 'eps', K + 1)

add\_to\_status(K + 1, ch, K + 1)

add\_to\_status(K + 1, 'eps', K + 2)

K = K + 2

T = K

elif check\_union(ch):

add\_to\_status(T, 'eps', K + 1)

add\_to\_status(K + 1, ch, K + 3)

add\_to\_status(K + 3, 'eps', K + 5)

add\_to\_status(T, 'eps', K + 2)

add\_to\_status(K + 2, get\_next(input, i + 1), K + 4)

add\_to\_status(K + 4, 'eps', K + 5)

K = K + 5

T = K

i += 2 *# skip next character*

else:

add\_to\_status(T, ch, K + 1)

K = K + 1

T = K

if ch == '(':

ST.append(T)

add\_to\_status(T, 'eps', K + 1)

K = K + 1

T = K

if ch == ')':

if sym != '\*':

pass

else:

add\_to\_status(T, 'eps', ST[-1])

add\_to\_status(ST[-1], 'eps', K + 1)

K = K + 1

T = K

ST.pop()

i += 1

print(f"输入的字符串为{input}")

for t0, ch, t1 in S:

print('%s\t[%s]\t%s' % (t0, ch, t1))