Regular Expression to E-NFA

By Thompson's Construction in C

CAU Computer science & engineering

20134220 Jeong hyun, Woo

1. How to execute

In Windows,

1. Execute CMD
2. Go to 'Win' Directory
3. Usage : >Automata.exe [regexp] ex> Automata.exe ab(a+b)*cde*

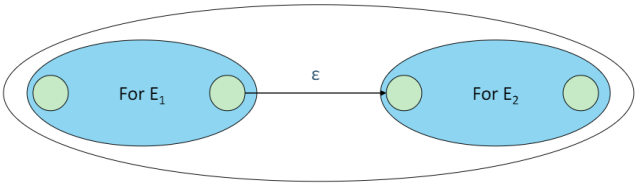
In Linux(Unix),

1. Execute Terminal
2. Go to 'Unix' Directory
3. $make
4. $ls (Check 'Automata')
5. Usage : $./Automata [regexp] ex> ./Automata ab(a+b)*ab

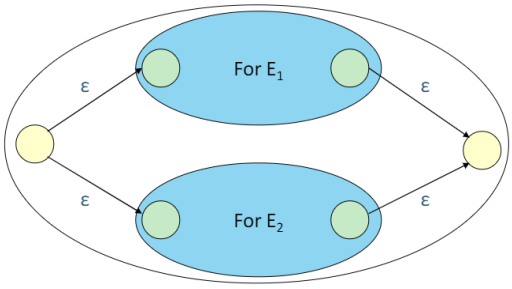
2. Flow of program

- Get input (regular expression) from user (argv[1])
- Extract alphabets from input && Check that the regular expression is correct or not (regex_extract_alphabet, regex_input_validation)
  - If not correct, return
- Insert additional symbol that represents 'concatenation operator (.)' to original regular expression. (regex_insert_concat)
- Convert concat-added regular expression to postfix. (regex_to_postfix)
- Create an NFA with postfix expression. (postfix_to_nfa) & Print E-NFA
  - Function 'nfa_symbol' works as

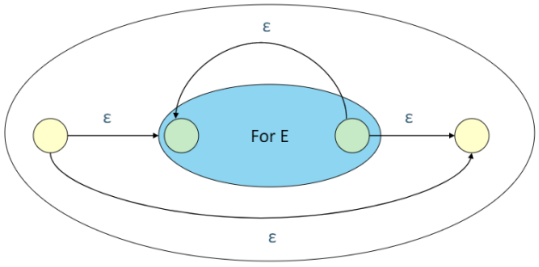
  - Function 'nfa_concat' works as



  - Function 'nfa_union' works as



  - Function 'nfa_kleene' works as



3. Result

※ 'E' in Delta means Epsilon (Not a symbol!)

In Windows,

~Win>Automata.exe ab+b*(cdef)

```
C:₩Users₩우정현₩Desktop₩[오토마타]20134220우정현₩Win>Automata.exe ab+b*(cdef)
 Input RE : ab+b*(cdef)

 Sigma          = { a b c d e f }
 Q              = { q0 q1 q2 q3 q4 q5 q6 q7 q8 q9 q10 q11 q12 q13 q14 q15 q16 q17 q18 q19 }
 Start-state    = q0
 Final-states   = { q19 }
 Where delta is,
 {
        Delta(0,E) = 17
        Delta(1,a) = 2
        Delta(3,b) = 4
        Delta(2,E) = 3
        Delta(5,b) = 6
        Delta(7,E) = 5
        Delta(7,E) = 8
        Delta(6,E) = 5
        Delta(6,E) = 8
        Delta(9,c) = 10
        Delta(11,d) = 12
        Delta(10,E) = 11
        Delta(13,e) = 14
        Delta(12,E) = 13
        Delta(15,f) = 16
        Delta(14,E) = 15
        Delta(8,E) = 9
        Delta(17,E) = 1
        Delta(17,E) = 7
        Delta(4,E) = 18
        Delta(16,E) = 18
        Delta(18,E) = 19
 }
```

~Win>Automata.exe a(a+b)*

```
C:₩Users₩우정현₩Desktop₩[오토마타]20134220우정현₩Win>Automata.exe a(a+b)*
 Input RE : a(a+b)*

 Sigma          = { a b }
 Q              = { q0 q1 q2 q3 q4 q5 q6 q7 q8 q9 q10 q11 }
 Start-state    = q0
 Final-states   = { q11 }
 Where delta is,
 {
        Delta(0,E) = 1
        Delta(1,a) = 2
        Delta(3,a) = 4
        Delta(5,b) = 6
        Delta(7,E) = 3
        Delta(7,E) = 5
        Delta(4,E) = 8
        Delta(6,E) = 8
        Delta(9,E) = 7
        Delta(9,E) = 10
        Delta(8,E) = 7
        Delta(8,E) = 10
        Delta(2,E) = 9
        Delta(10,E) = 11
 }
```

In Linux,

~Unix$ make

~Unix$ ./Automata a\(a+b\)* (== a(a+b)*)

```
klokov@klokov-virtual-machine:/mnt/hgfs/win/Win$ ls
Automata  Makefile  define.c  define.h  define.o  main.c  main.o
klokov@klokov-virtual-machine:/mnt/hgfs/win/Win$ ./Automata a\(a+b\)*
 Input RE : a(a+b)*

 Sigma          = { a b }
 Q              = { q0 q1 q2 q3 q4 q5 q6 q7 q8 q9 q10 q11 }
 Start-state    = q0
 Final-states   = { q11 }
 Where delta is,
 {
        Delta(0,E) = 1
        Delta(1,a) = 2
        Delta(3,a) = 4
        Delta(5,b) = 6
        Delta(7,E) = 3
        Delta(7,E) = 5
        Delta(4,E) = 8
        Delta(6,E) = 8
        Delta(9,E) = 7
        Delta(9,E) = 10
        Delta(8,E) = 7
        Delta(8,E) = 10
        Delta(2,E) = 9
        Delta(10,E) = 11
 }
```

~Unix$ ./Automata (01)*110(0+1)*

```
klokov@klokov-virtual-machine:/mnt/hgfs/win/Win$ ./Automata \(01\)*110\(0+1\)*
 Input RE : (01)*110(0+1)*

 Sigma          = { 0 1 }
 Q              = { q0 q1 q2 q3 q4 q5 q6 q7 q8 q9 q10 q11 q12 q13 q14 q15 q16 q17 q18 q19 q20 q21 }
 Start-state    = q0
 Final-states   = { q21 }
 Where delta is,
 {
        Delta(0,E) = 5
        Delta(1,0) = 2
        Delta(3,1) = 4
        Delta(2,E) = 3
        Delta(5,E) = 1
        Delta(5,E) = 6
        Delta(4,E) = 1
        Delta(4,E) = 6
        Delta(7,1) = 8
        Delta(6,E) = 7
        Delta(9,1) = 10
        Delta(8,E) = 9
        Delta(11,0) = 12
        Delta(10,E) = 11
        Delta(13,0) = 14
        Delta(15,1) = 16
        Delta(17,E) = 13
        Delta(17,E) = 15
        Delta(14,E) = 18
        Delta(16,E) = 18
        Delta(19,E) = 17
        Delta(19,E) = 20
        Delta(18,E) = 17
        Delta(18,E) = 20
        Delta(12,E) = 19
        Delta(20,E) = 21
 }
```