Rajshahi University of Engineering & Technology
Department of Computer Science and Engineering

Compiler
Lexical Analysis

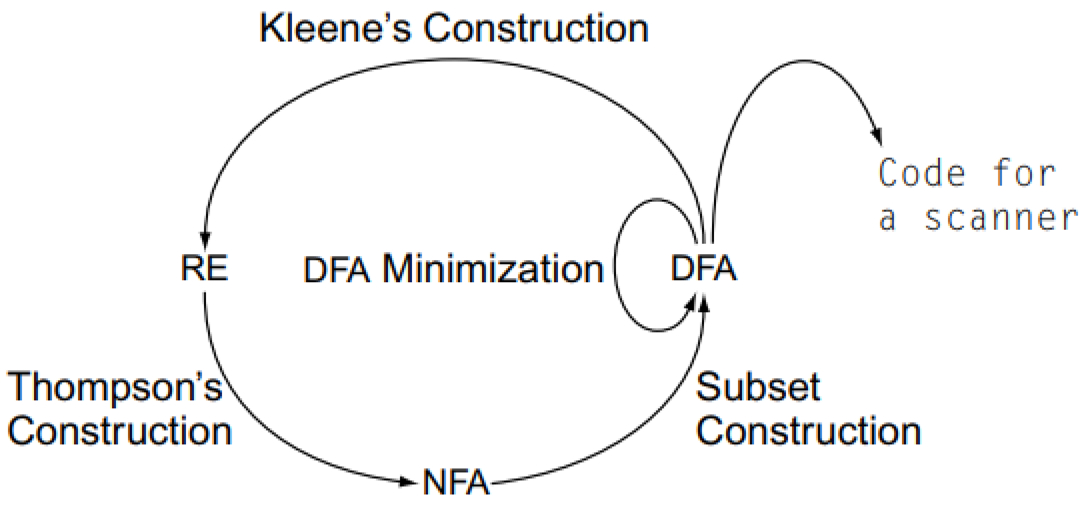
Md. Sozib Hossain
Lecturer, CSE
sozib.hossain@cse.ruet.ac.bd

❖ Introduction

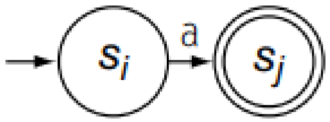
Regular Expression(RE) is a notation for designing Recognizer that accepts a specific pattern. It is relatively easier to design.

Finite Automata(FA) is the actual machine that recognize pattern. There are two types of FA: NFA and DFA. In that case, NFA is more easier to design than DFA. But NFA can not be implemented in practical.

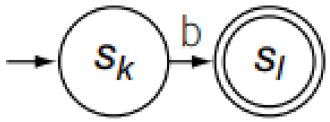
So at the end we have to convert all RE and NFA to DFA.



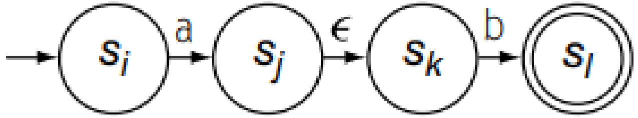
❖ Thompson’s Construction(RE to NFA)



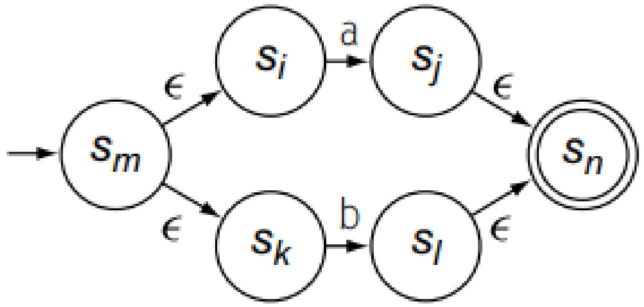
(a) NFA for “*a*”



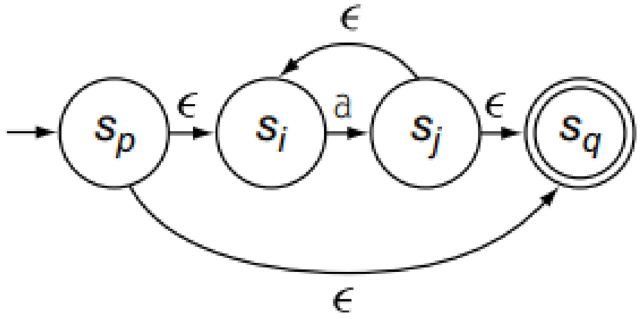
(b) NFA for “*b*”



(c) NFA for “*ab*”

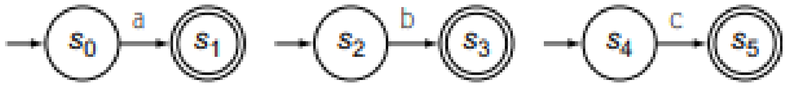


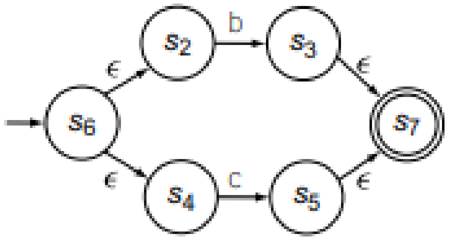
(d) NFA for “*a* | *b*”

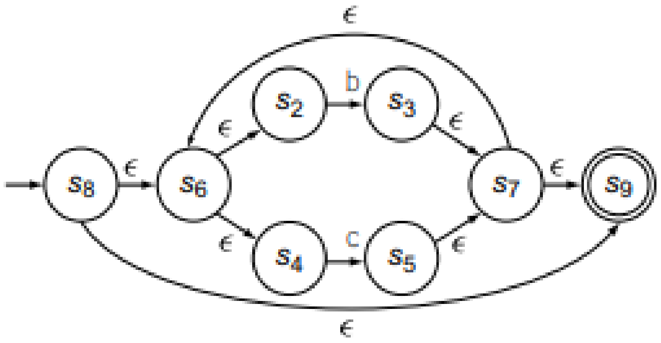


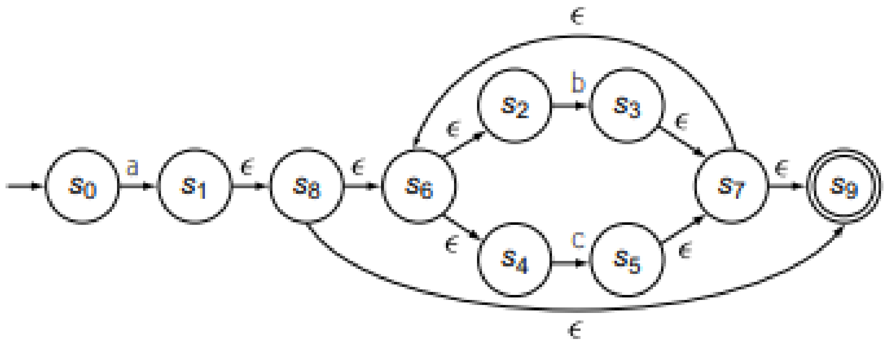
(e) NFA for “*a**”

❖ Example


(a) NFAs for "*a*", "*b*", and "*c*"


(b) NFA for "*b* | *c*"

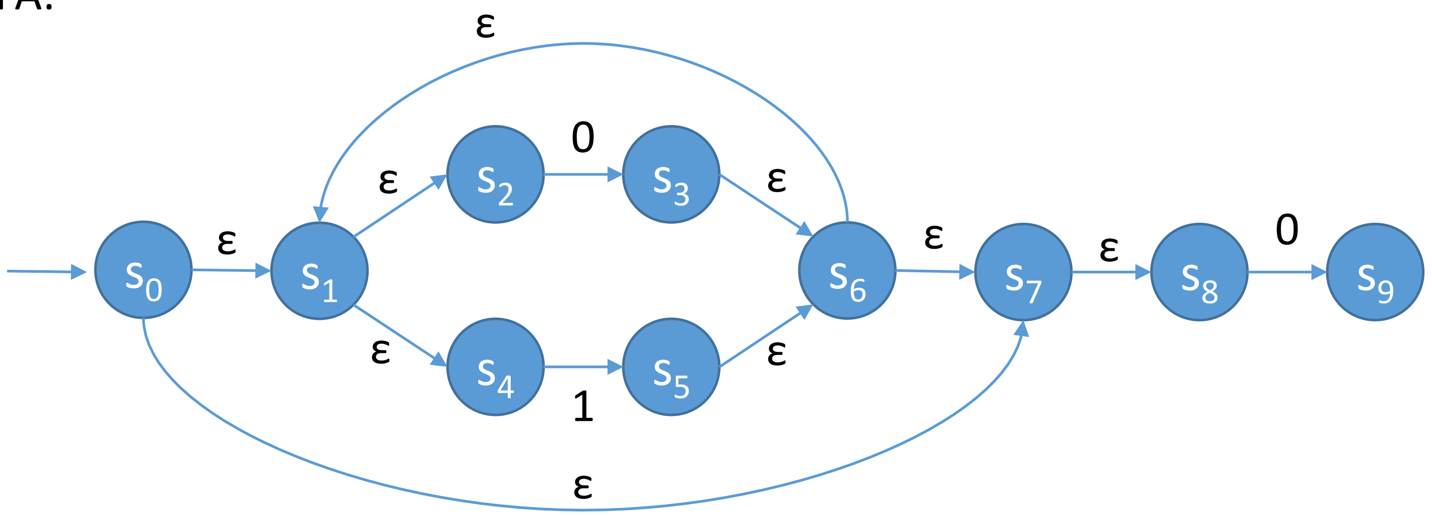

(c) NFA for "(*b* | *c*)*"


(d) NFA for "*a*(*b* | *c*)*"

sozib.hossain@cse.ruet.ac.bd

❖ Example

- Language: All binary even number
- Regular Expression: (0|1)*0
- NFA:



s_0
s_1
s_2
s_3
s_4
s_5
s_6
s_7
s_8
s_9
ε
0
1