

CD: COMPILER DESIGN
SPM: UNIT-1 18/09/2022

18 SEPTEMBER 2022 / IT-3rd year, Vth semester
FALL SEMESTER, YEAR (Vth, 3rd)
FALL SESSION (2021-23)
(CD)
MS. SHWETA TIWARI
Published: SEPTEMBER 18, 2022

PREPARED FOR

Engineering Students
All Engineering College

CD: COMPILER DESIGN

TOPIC On : UNIT-1 LEXICAL ANALYZER | Thompson method | RE TO NFA | NFA TO DFA

By SHWETA TIWARI

Under On: INTRODUCTION TO COMPILER

TOPIC On : UNIT-1 LEXICAL ANALYZER | Thompson method | RE TO NFA | NFA TO DFA

Regular expression to NFA using Thompson's rule

❑ Regular expression to NFA (Thompson's construction)

1. For ϵ , construct the NFA

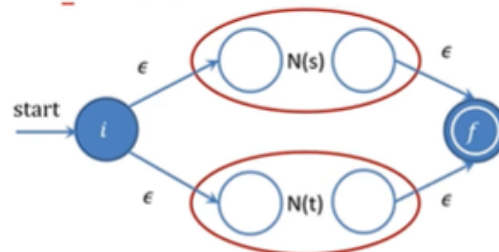


2. For a in Σ , construct the NFA

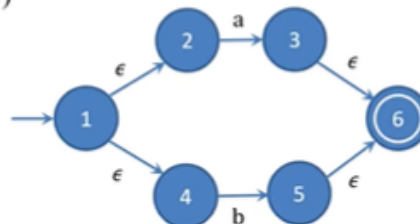


❑ Regular expression to NFA (Thompson's construction)

3. For regular expression $s|t$ or $(s+t)$



For Example: $(a|b)$ or $(a + b)$



❑ Regular expression to NFA (Thompson's construction)

4. For regular expression st

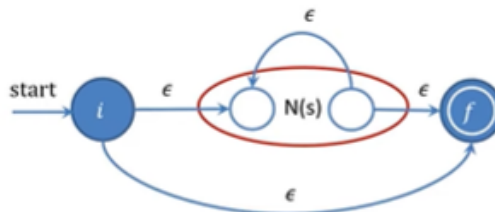


For Example: ab

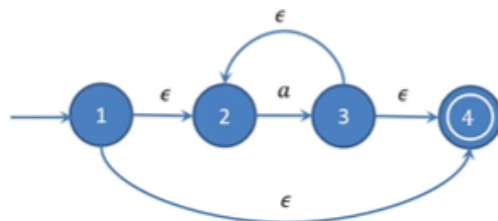


❑ Regular expression to NFA (Thompson's construction)

5. For regular expression s^*

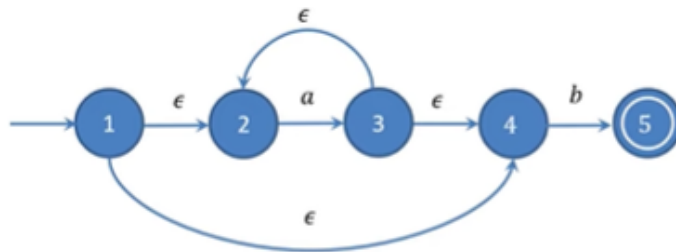


For Example : a^*

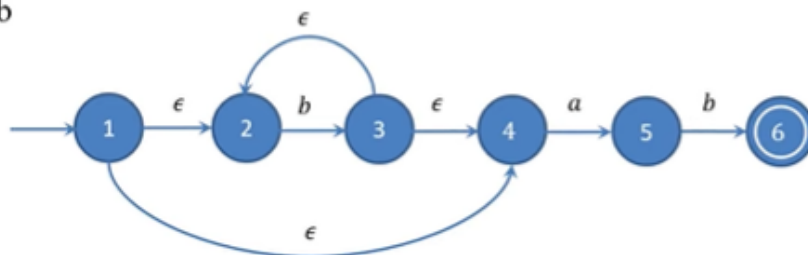


❑ Regular expression to NFA (Examples)

➤ a^*b

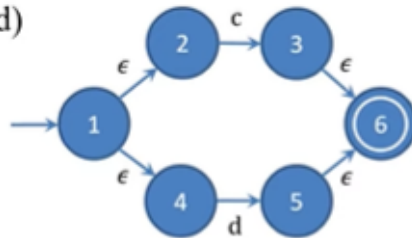


➤ b^*ab

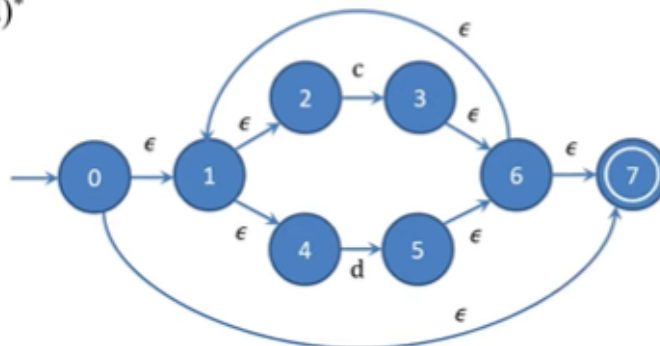


❑ Regular expression to NFA (Examples)

➤ $(c|d)$ **or** $(c + d)$



➤ $(c|d)^*$ **or** $(c + d)^*$



□ Exercise

Convert following regular expression to NFA:

1. abba
2. $bb(a)^*$
3. $(a|b)^*$
4. $a^* | b^*$
5. $a(a)^*ab$
6. $aa^* + bb^*$
7. $(a+b)^*abb$
8. $10(0+1)^*1$
9. $(a+b)^*a(a+b)$
10. $(0+1)^*010(0+1)^*$
11. $(010+00)^*(10)^*$
12. $100(1)^*00(0+1)^*$

Conversion from NFA to DFA using subset construction method

❑ Subset construction algorithm

Input: An NFA N .

Output: A DFA D accepting the same language.

Method: Algorithm construct a transition table D_{tran} for D . We use the following operation:

OPERATION	DESCRIPTION
$\epsilon - closure(s)$	Set of NFA states reachable from NFA state s on ϵ -transition alone.
$\epsilon - closure$	Set of NFA states reachable from some NFA state s in T on ϵ -transition alone.
$\delta(T, a)$	Set of NFA states to which there is a transition on input symbol a from some NFA state s in T .

❑ Exercise

❖ Convert following regular expression to DFA using subset construction method.

OR

❖ Construct a NFA for following regular expression using Thompson's notation and then convert it into DFA.

1. $(010+00)^*(10)^*$
2. $(a+b)^*a(a+b)$
3. $(a|b)^*a(a|b)^{\#}$

DFA Optimization

□ DFA Optimization Algorithm

1. Construct an initial partition Π of the set of states with two groups: the accepting states F and the non-accepting states $S - F$.
2. Apply the repartition procedure to Π to construct a new partition Π_{new} .
3. If $\Pi_{new} = \Pi$, let $\Pi_{final} = \Pi$ and continue with step (4). Otherwise, repeat step (2) with $\Pi = \Pi_{new}$.

for each group G of Π **do begin**

 partition G into subgroups such that two states s and t
 of G are in the same subgroup if and only if for all
 input symbols a , states s and t have transitions on a
 to states in the same group of Π .

 replace G in Π_{new} by the set of all subgroups formed.

end

□ DFA Optimization Algorithm

4. Choose one state in each group of the partition Π_{final} as the representative for that group. The representatives will be the states of M' . Let s be a representative state, and suppose on input a there is a transition of M from s to t . Let r be the representative of t 's group. Then M' has a transition from s to r on a . Let the start state of M' be the representative of the group containing start state s_0 of M , and let the accepting states of M' be the representatives that are in F .
5. If M' has a dead state d , then remove d from M' . Also remove any state not reachable from the start state.