Lecture 9

Proof of $q \Rightarrow p$ cont. (RE-to-NFA conversion)

Example: $\alpha = (ab)^* b + \varepsilon$

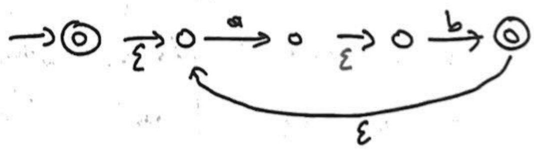
For a: $\rightarrow \circ \xrightarrow{a} \circledcirc$

For b: $\rightarrow \circ \xrightarrow{b} \circledcirc$

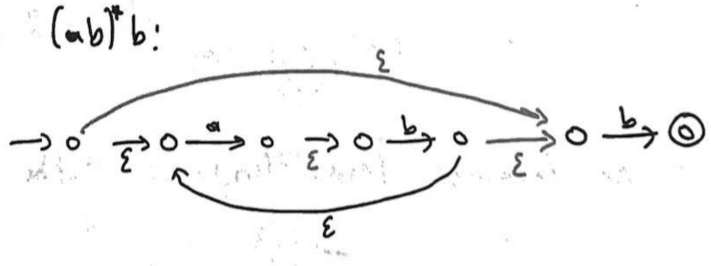
For ε: $\rightarrow \circledcirc$

For ab: $\rightarrow \circ \xrightarrow{a} \circ \xrightarrow[\varepsilon]{} \circ \xrightarrow{b} \circledcirc$

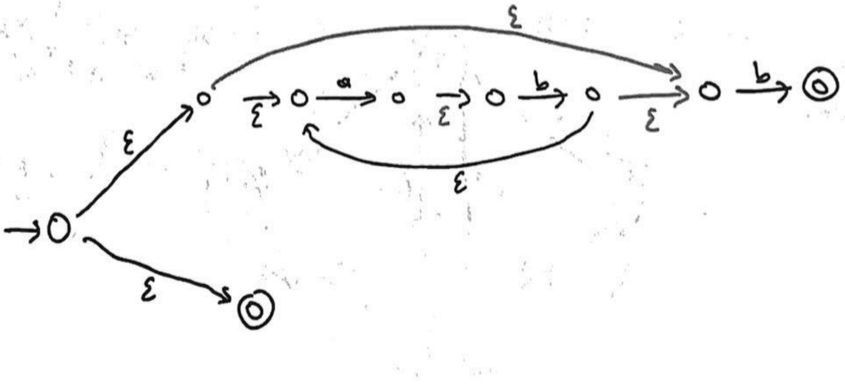
For $(ab)^*$:



For $(ab)^* b$:



For $(ab)^* b + \varepsilon$:



Exercise: NFA for $a^*(b+a)b + \varepsilon$?

<u>Proof of $p \Rightarrow q$:</u>

(i.e. $L = L(N)$ for some NFA N $\Rightarrow$ $L = L(\alpha)$ for some RE α.)

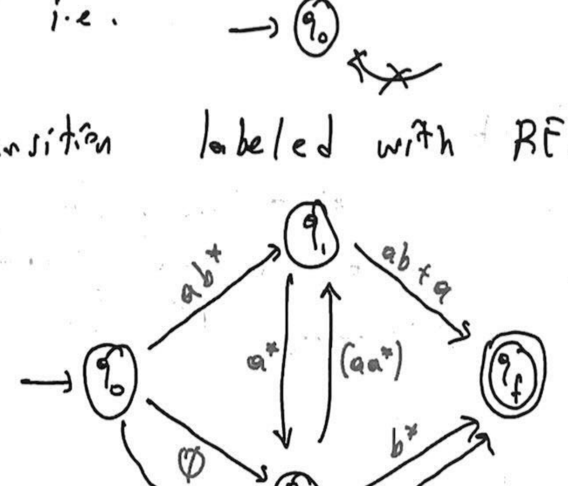
NFA-to-RE conversion.

To prove this, we introduce a new computational model called "generalized NFA" (GNFA).

A GNFA has

- a unique initial state q_0
- a unique final state. q_f
- no outgoing transitions from the final state. i.e. $(q_f) \not\to$
- no incoming transitions to the initial state. i.e. $\to (q_0)$
- transition labeled with REs.

<u>Example:</u>



$w = \overbrace{ab^5}^{w_1}\overbrace{a^2}^{w_2}\overbrace{b^3}^{w_3}$ ✓

$w = \underline{b^3}$ ✗

Formally, a GNFA N is defined as
$N = (Q, \Sigma, \delta, q_0, q_f)$ where

$$\delta : (Q - \{q_f\}) \times (Q - \{q_0\}) \longrightarrow RE$$

$$q \xrightarrow{RE} q' \ .$$

<u>Definition:</u> An input string $w \in \Sigma^*$ is accepted by a GNFA N if w can be written as a concatenation of substrings (i.e. $w = w_1 \cdots w_n$) and there exists a sequence of states $r_0, r_1, \ldots, r_n$ with the following properties

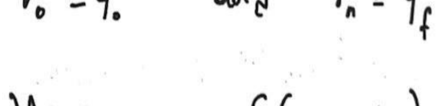
1) $r_0 = q_0$ and $r_n = q_f$

2) $\forall i = 1, \ldots, n \quad \delta(r_i, r_{i+1}) = R_i \ \wedge \ w_i \in L(R_i)$.

e.g. $r_0 \xrightarrow{w_1} r_1 \rightarrow r_2 \ \cdots \ \rightarrow r_n$
$\uparrow w_1 \in L(R_1)$

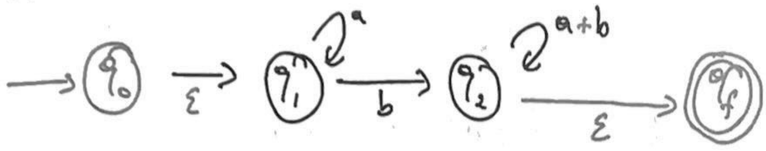
(Our proof strategy : NFA $\rightarrow$ GNFA $\rightarrow$ RE)

Example: NFA $\rightarrow$ GNFA transformation

N :  The equiv. GNFA?

GNFA

N':



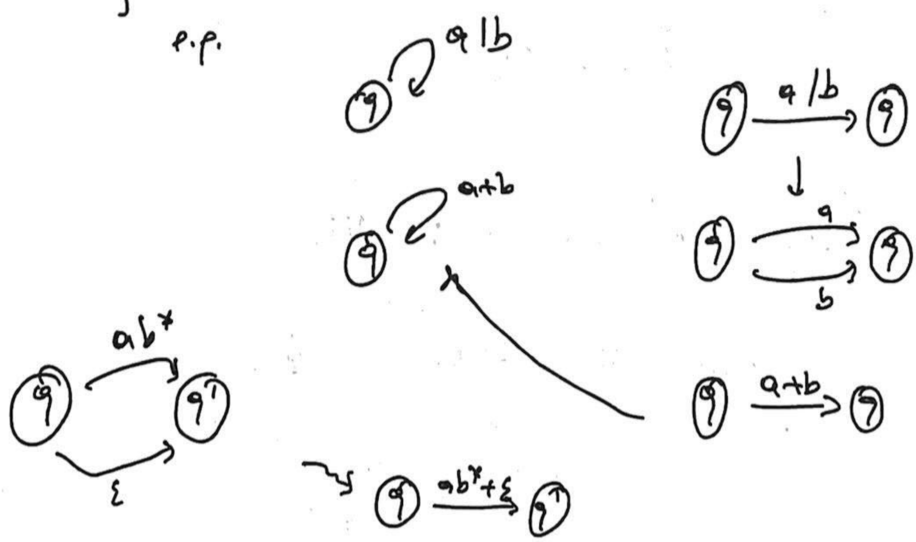
<u>In general:</u>

Input: NFA $N = (Q, \Sigma, \delta, q_1, F)$

Output: GNFA $N' = (Q \cup \{q_0, q_f\}, \Sigma, \delta', q_0, q_f)$

- Add q_0 and an ε-transition to q_0 to q_1.
- Add q_f and ε-transitions from the states in F to q_f
- Replace transitions between each pair of states of N by a single transition whose label is a "+" of labels of original transitions.

e.g.



- Eliminating states from GNFA:
  We want to obtain an equivalent GNFA with only two states, namely q_0 and q_f.

  Then the label of the transition between q_0 and q_f will be the desired RE.

  <u>Algorithm:</u>

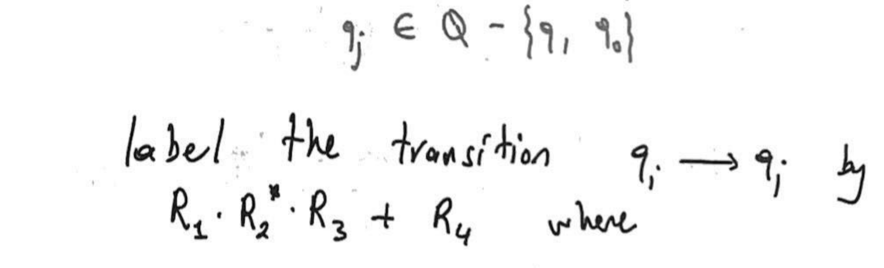
  <u>Repeat</u>

  - Select a state $q \in Q - \{q_0, q_f\}$
  - For all $q_i \in Q - \{q, q_f\}$ and
    $q_j \in Q - \{q, q_0\}$

  label the transition $q_i \longrightarrow q_j$ by $R_1 \cdot R_2^* \cdot R_3 + R_4$ where

  R_1: label of $q_i \rightarrow q$
  R_2: " " $q \rightarrow q$
  R_3: " " $q \rightarrow q_j$
  R_4: " " $q_i \rightarrow q_j$

  <u>Until</u> there are two states left.

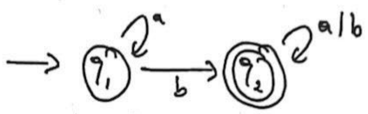


Overall, the algorithm for NFA-to-RE conversion:

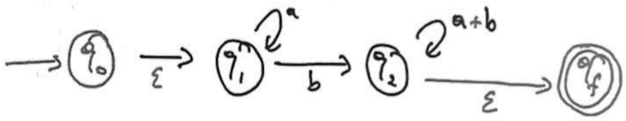
- Transform the NFA N into an equiv. GNFA N'
- Transform GNFA N' into an equiv. GNFA N'' with only two states
- Return the label of the transition $q_0 \longrightarrow q_f$ in N''.

Example:

N :

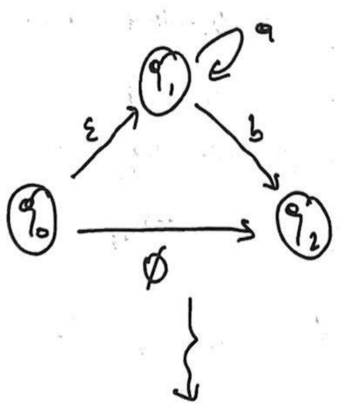


GNFA N' :



We need to eliminate q_1 and q_2.

Eliminate q_1:



$R_1 = \varepsilon$

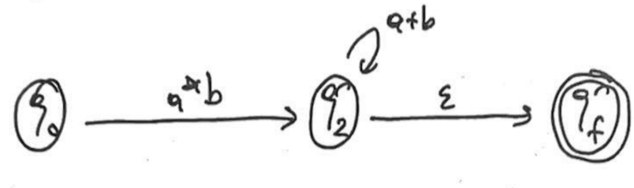
$R_2 = a$

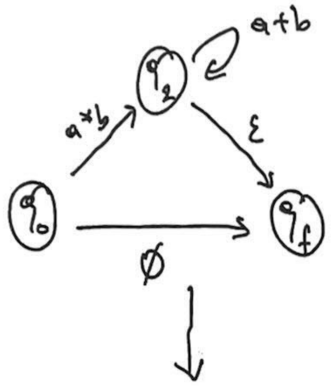
$R_3 = b$

$R_4 = \emptyset$

$$q_0 \xrightarrow{\varepsilon \cdot a^* b + \emptyset} q_2 \equiv q_0 \xrightarrow{a^* b} q_2$$

Now we have



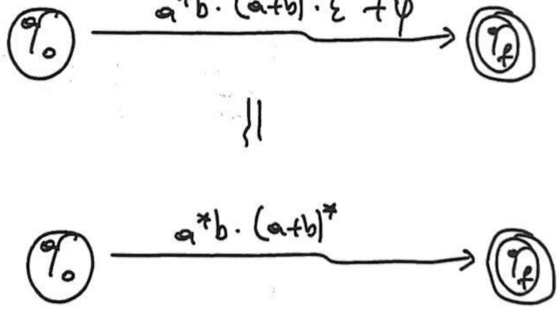


$R_1 = a^*b$

$R_2 = a+b$

$R_3 = \varepsilon$

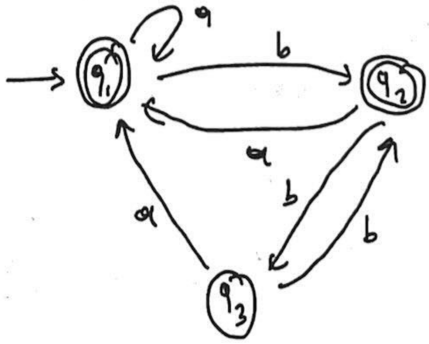
$R_4 = \emptyset$



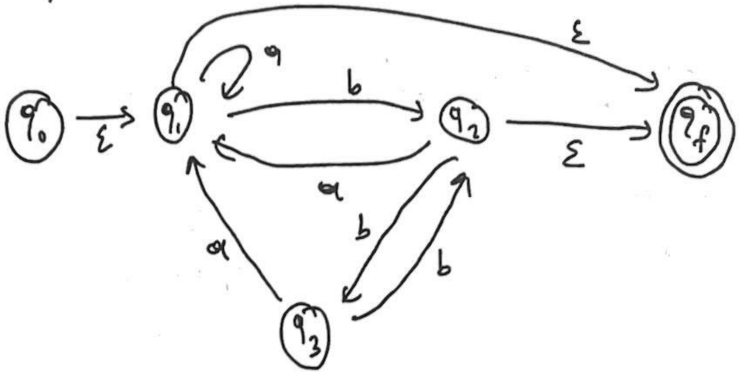
The equiv. RE is $a^* b \cdot (a+b)^*$

Example:

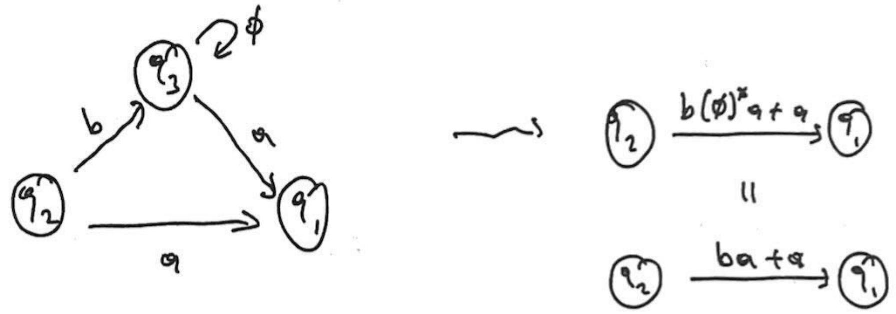
N :

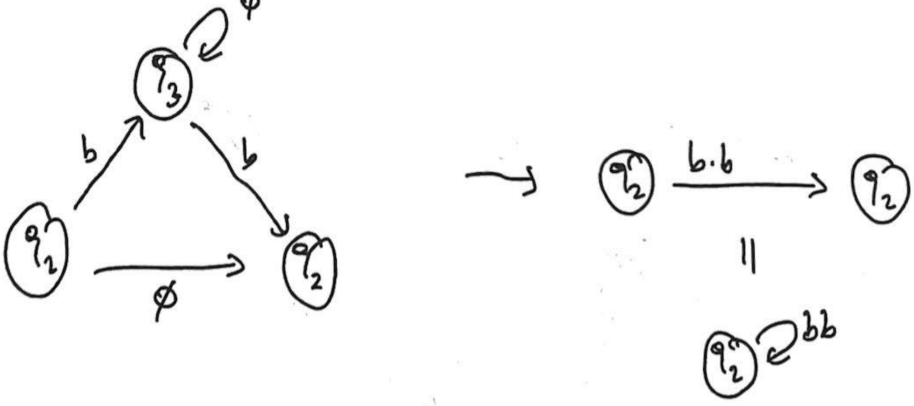


The equiv. GNFA N':

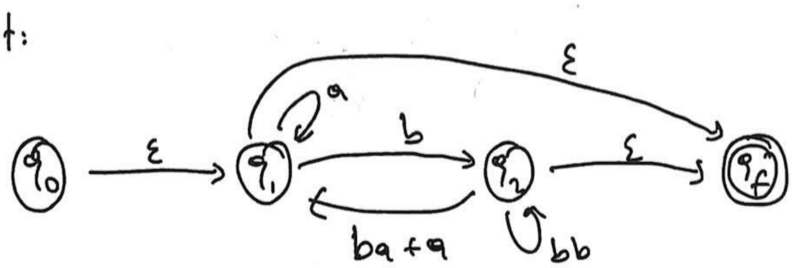


Eliminate q_3:

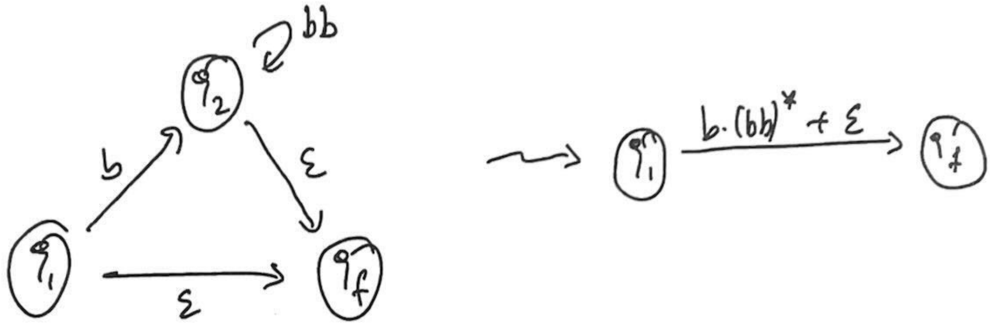


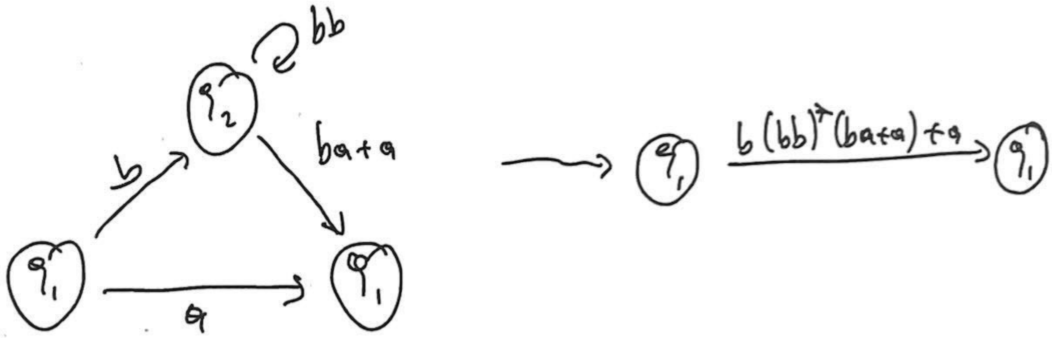


We get:

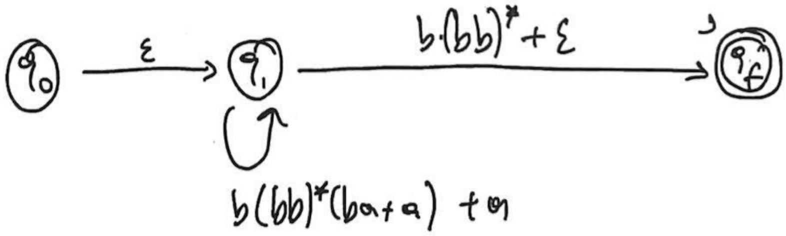


Eliminate q_2:

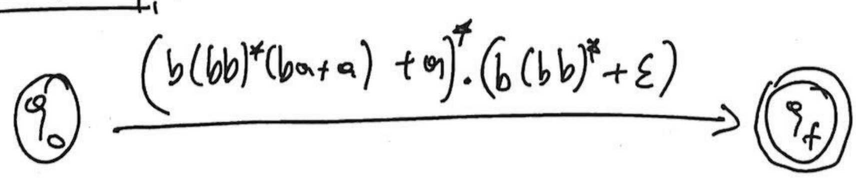




We get:



Eliminate q_1:



The equiv. RE : $(b(bb)^*(ba+a)+a)^* \cdot (b(bb)^*+\varepsilon)$