Theorem Let Σ be a finite alphabet.
The class of languages over Σ that are regular is equal to the class of languages that are described by regular expressions

Proof has 2 directions:

(i) L has a regular expression $\rightarrow$ L has an NFA
(proof uses closure properties!)

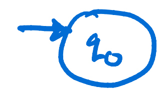
(ii) L has a DFA (or NFA) $\rightarrow$ L has a regular expression
(harder)

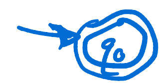
(i) L has a regexp $\longrightarrow$ L is regular

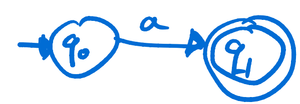
Proof : By induction on length (number of $*, \cup, \cdot$) of regexp

<u>Base Case</u> R contains NO operations : $*, \cdot, \cup$

<u>Language $L(R)$:</u> | <u>NFA :</u>

$R = \phi \longrightarrow L(\phi) = \phi \longrightarrow$ 

$R = \varepsilon \longrightarrow L(\varepsilon) = \{\varepsilon\} \longrightarrow$ 

$R = a,\ a \in \Sigma \longrightarrow L(a) = \{a\} \longrightarrow$ 

For all cases, there is an NFA for $L(R)$, so $L(R)$ is regular.

(i) L has a regexp $\rightarrow$ L is regular

Inductive Step: Assume (inductive hypothesis) that for every regexp R with $\leq K$ operations $*, \cup, \circ$, $L(R)$ is regular

Show: For any regexp R with $K+1$ operations, $L(R)$ is regular

Case 1: $R = R_1 \cup R_2$, so $L(R) = L(R_1) \cup L(R_2)$

- By assumption, $L(R_1)$ and $L(R_2)$ are both regular
- By closure property of Union:

  $L(R_1), L(R_2)$ regular $\rightarrow L(R_1 \cup R_2) = L(R)$ is regular

Case 2: $R = R_1 \circ R_2$. Same idea as case 1, Now use closure property of concatenation

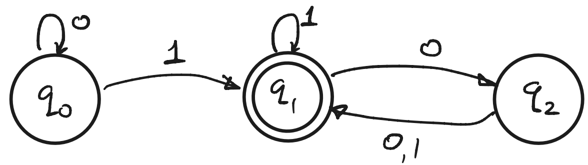
Case 3: $R = (R)^*$ Same idea, use closure of star

(ii) L has a DFA (or NFA) $\rightarrow$ L has a regular expression

To prove this direction we will give an <u>algorithm</u> that takes as input a DFA or NFA M produces a regular expression such that the language accepted by M corresponds to the regular expression

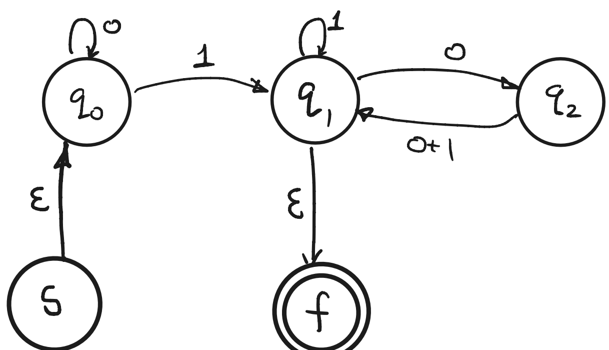
* Our algorithm different (and easier I think) than Sipser.

Example 1 : constructing Regular Expression from an NFA

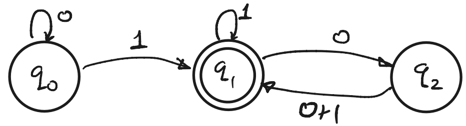


step 1

- New start state s with ε-transition to original start state
- New single accept state f with ε-transitions from old accept states to f

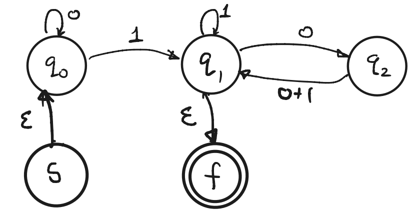


Step 0

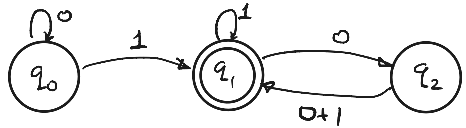


step 1

- New start state s with ε-transition to original start state
- New single accept state f with ε-transitions from old accept states to f

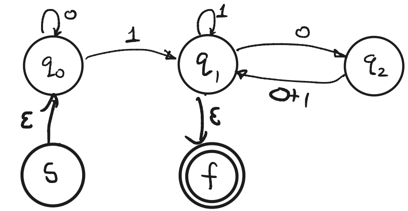


Step 0



Step 1

- New start state s with ε-transition to original start state
- New single accept state f with ε-transitions from old accept states to f



Step 2 (remove q_1)

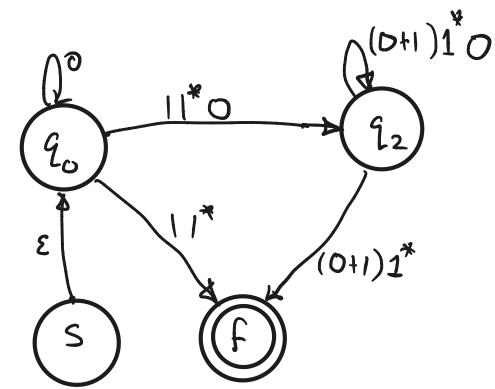
- Consider all pairs of edges $(q \to q_1 \quad q_1 \to q')$, $q, q' \neq q_1$

$q_0 \to q_1 \quad q_1 \to q_2$: $1\,1^*\,0$

$q_0 \to q_1 \quad q_1 \to f$: $1\,1^*$

$q_2 \to q_1 \quad q_1 \to f$: $(0+1)1^*$

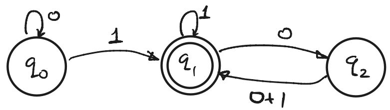
$q_2 \to q_1 \quad q_1 \to q_2$: $(0+1)\,1^*\,0$



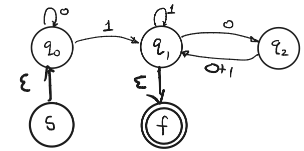
- Remove q_1.

- For all pairs $q \to q_1 \quad q_1 \to q'$ add the corresponding regular expression to edge $q \to q'$

Step 0

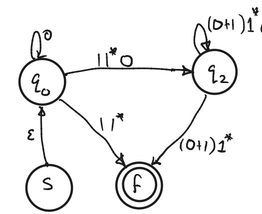


Step 1



- New start state s with ε-transition to original start state
- New single accept state f with ε-transitions from old accept states to f

Step 2 (remove q_1)



- Consider all pairs of edges $(q \to q_1 \quad q_1 \to q')$, $q, q' \neq q_1$

$q_0 \to q_1 \quad q_1 \to q_2 : 11^*0$

$q_0 \to q_1 \quad q_1 \to f : 11^*$

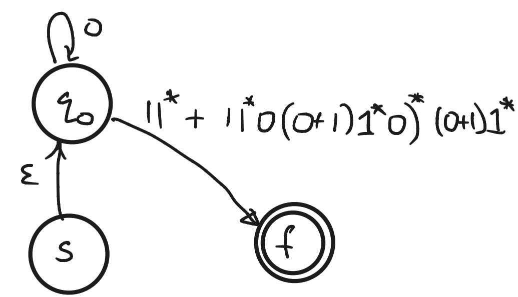
$q_2 \to q_1 \quad q_1 \to f : (0+1)1^*$

$q_2 \to q_1 \quad q_1 \to q_2 : (0+1)1^*0$

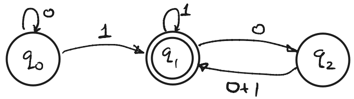
- Remove q_1.
- For all pairs $q \to q_1 \quad q_1 \to q'$ add the corresponding regular expression to edge $q \to q'$

Step 3 (remove q_2)

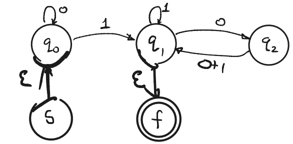
$q_0 \to q_2 \quad q_2 \to f : 11^*0((0+1)1^*0)^*(0+1)1^*$



Step 0

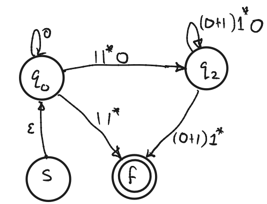


Step 1



- New start state s with ε-transition to original start state
- New single accept state f with ε-transitions from old accept states to f

Step 2 (remove q_1)



- Consider all pairs of edges ($q \to q_1$, $q_1 \to q'$), $q, q' \neq q_1$

$q_0 \to q_1 \quad q_1 \to q_2$: $1\,1^*\,0$

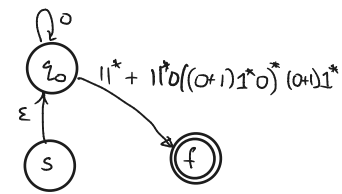
$q_0 \to q_1 \quad q_1 \to f$: $1\,1^*$

$q_2 \to q_1 \quad q_1 \to f$: $(0+1)1^*$

$q_2 \to q_1 \quad q_1 \to q_2$: $(0+1)\,1^*\,0$

- Remove q_1.
- For all pairs $q \to q_1$ $q_1 \to q'$ add the corresponding regular expression to edge $q \to q'$

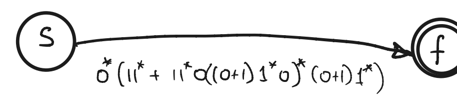
Step 3 (remove q_2)



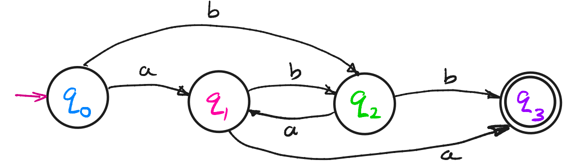
$q_0 \to q_2 \quad q_2 \to f$: $11^*0\,((0+1)1^*0))^*\,(0+1)1^*$

Step 4 (remove q_0)

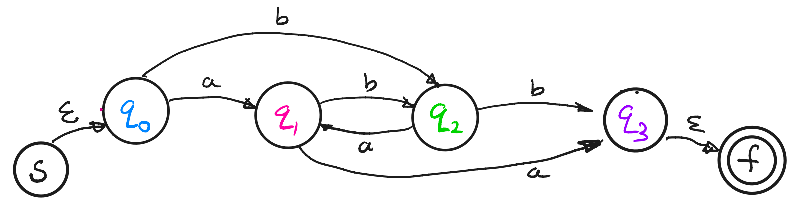
$s \to q_0 \to f$ $\quad 0^*(11^* + 11^*0((0+1)1^*0)^*(0+1)1^*)$



Example 2



Step 1



Step 2

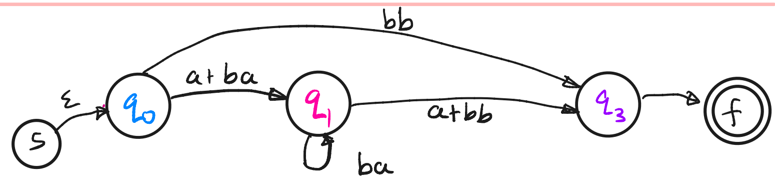
Remove q_2

$q_1 \to q_2 \to q_1$: ba

$q_1 \to q_2 \to q_3$: bb

$q_0 \to q_2 \to q_1$: ba

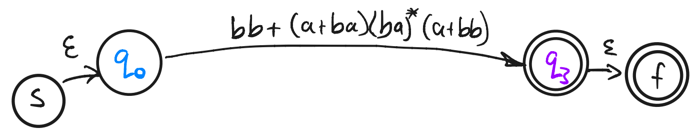
$q_0 \to q_2 \to q_3$: bb



Step 3

Remove q_1

$q_0 \to q_1 \to q_3$: $(a+ba)(ba)^*(a+bb)$



Steps 4-5

Remove q_0, q_3