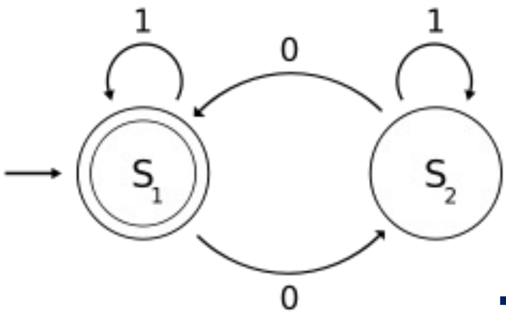
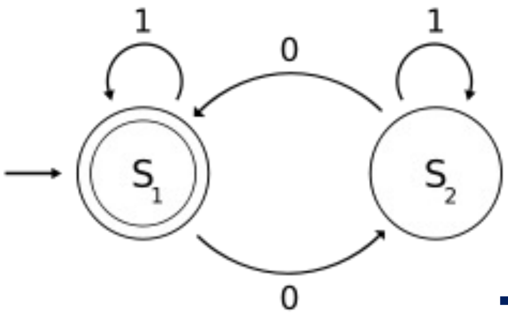


Regular Expressions

Sipser: Section 1.3 pages 63 - 69



NFA Exercise

Consider the following NFA $N = (Q, \Sigma, \delta, q_1, F_1)$ where

$Q = \{q_1, q_2, q_3\}$

$\Sigma = \{a, b\}$

q_1 is the start state

$F = \{q_3\}$

$\delta(q_1, a) = \{q_2, q_3\}$

$\delta(q_1, b) = \varnothing$

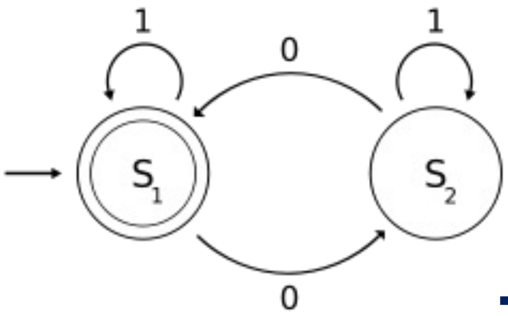
$\delta(q_2, a) = \varnothing$

$\delta(q_2, b) = \{q_2\}$

$\delta(q_3, a) = \{q_1, q_2\}$

$\delta(q_3, b) = \{q_2, q_3\}$

$\delta(q, \varepsilon) = \varnothing$ for all $q \in Q$



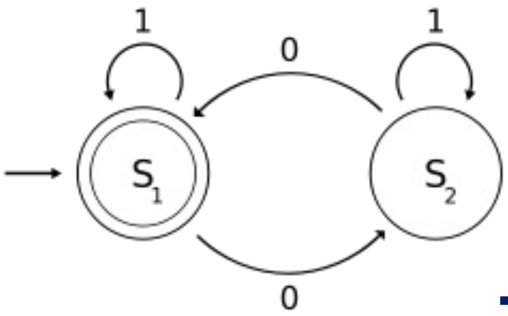
Regular Expressions

$(0 \cup 1)0^*$

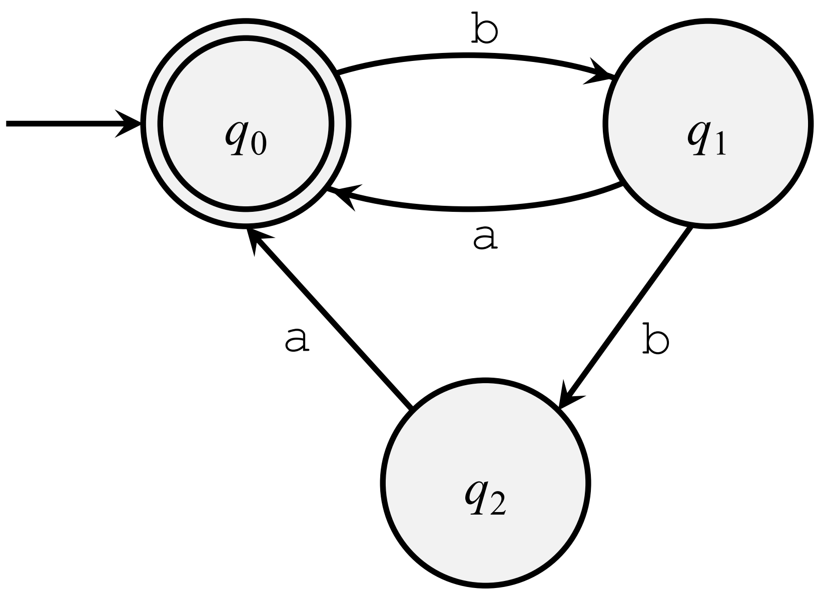
$(0 \cup 1)^*$

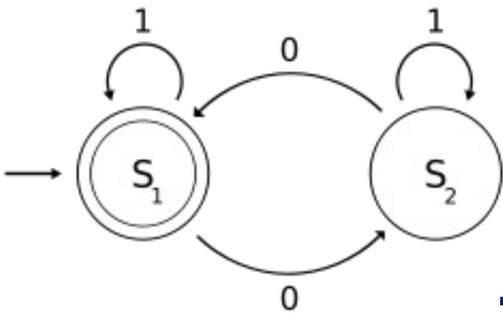
Σ^*1

R^+

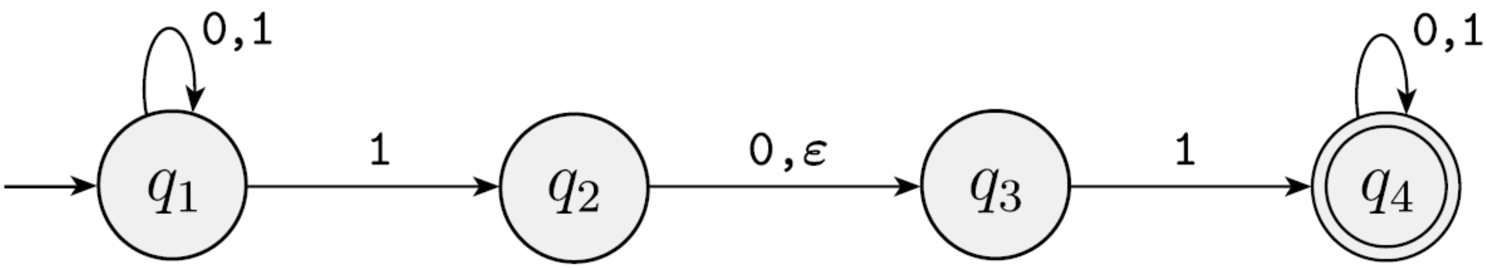


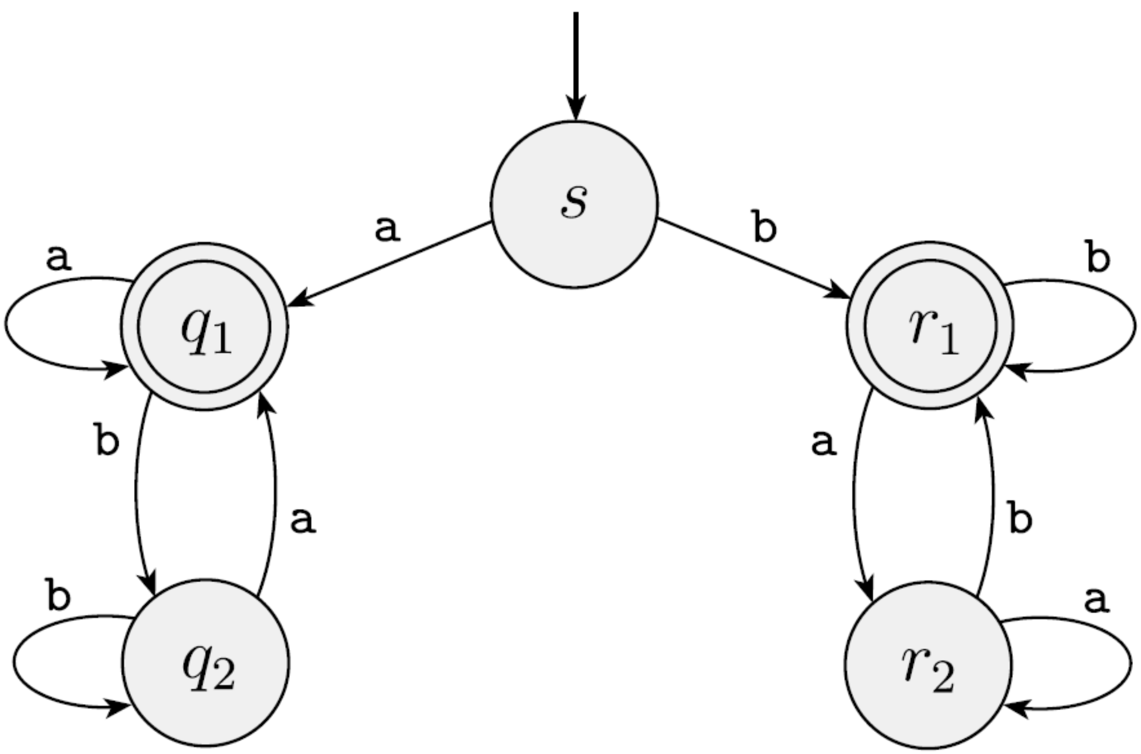
Long Ago in a Place Not Far Away

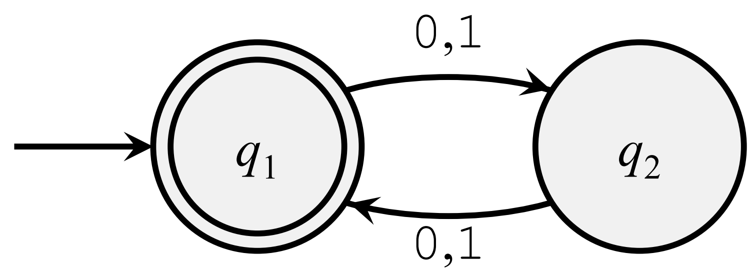


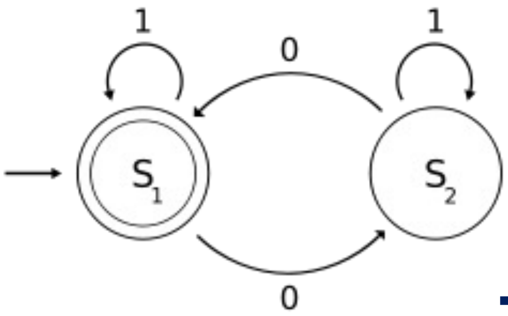


Old Home Week





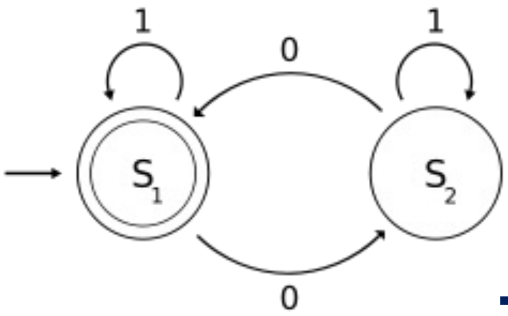




Regular Expressions

Definition. Say that R is a ***regular expression*** if R is

1. a for some a in the alphabet Σ,
2. ε,
3. $\emptyset$,
4. $(R_1 \cup R_2)$, where R_1 and R_2 are regular expressions,
5. $(R_1 \circ R_2)$, where R_1 and R_2 are regular expressions,
6. $(R_1)^*$, where R_1 is a regular expression.



Working with Regular Expressions

0^*10^* = { w | }

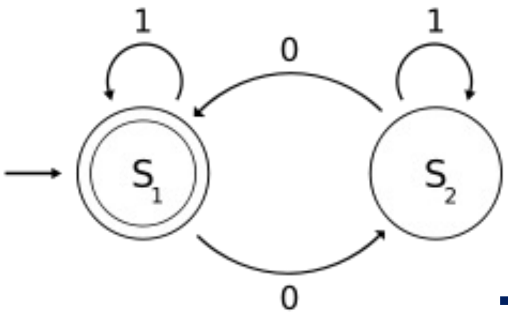
= { w | w is a string of odd length }

$(0 \cup \varepsilon)(1 \cup \varepsilon)$ =

$(01)^*\emptyset$ =

$(+ \cup - \cup \varepsilon)(DD^* \cup DD^*.D^* \cup D^*.DD^*)$ =

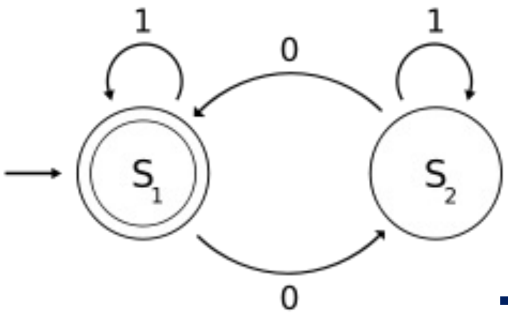
where $D = \{0,1,2,3,4,5,6,7,8,9\}$



Identities

Let R be a regular expression.

- $R \cup \emptyset =$
- $R \circ \varepsilon =$
- $R \cup \varepsilon =$
- $R \circ \emptyset =$

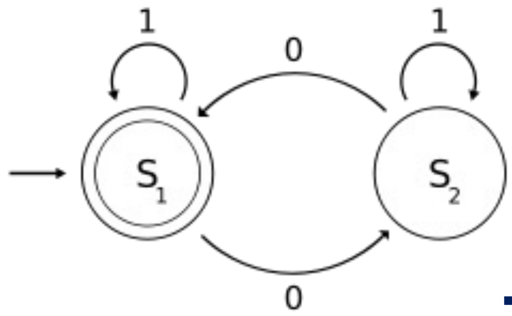


Regular Expressions and NFAs

Theorem. *A language is regular if and only if some regular expression describes it.*

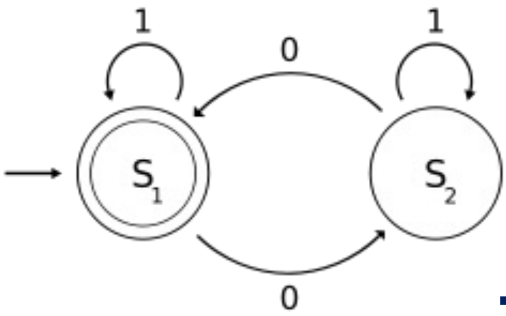
Proof. (⇐)

1. If $a \in \Sigma$, then a is regular.
2. ε is regular.
3. $\emptyset$ is regular.
4. If R_1 and R_2 are regular, then $(R_1 \cup R_2)$ is regular.
5. If R_1 and R_2 are regular, then $(R_1 \circ R_2)$ is regular.
6. If R_1 is regular, then $(R_1)^*$ is regular.



Proof in Action

Build an NFA that recognizes the regular expression: $(ab \cup a)^*$



Proof in Action

Build an NFA that recognizes the regular expression:

a(a ∪ b)*a