Connection Between Regular Expressions and Regular Languages

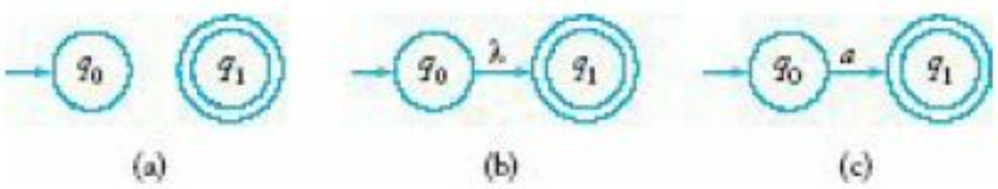
Section 3.2 [Textbook-1]

Regular Expressions and Regular Languages

- for every regular language there is a regular expression
- for every regular expression there is a regular language.

Regular Expressions and Regular Languages

- Regular Expressions denote Regular Languages

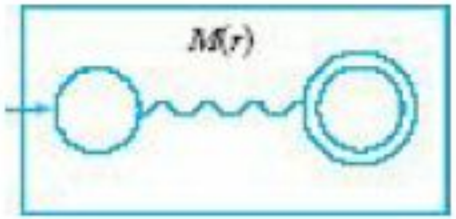


(a) nfa accepts $\emptyset$.

(b) nfa accepts $\{\lambda\}$.

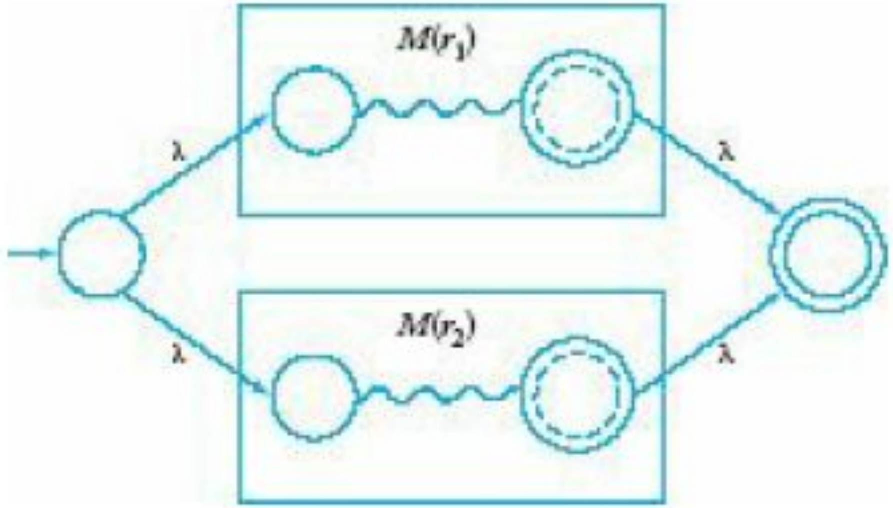
(c) nfa accepts $\{a\}$.

Regular Expressions and Regular Languages



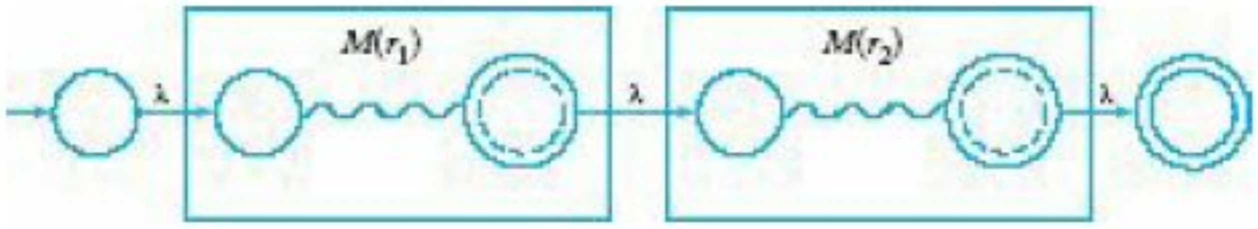
Schematic representation of an nfa accepting $L(r)$.

Regular Expressions and Regular Languages

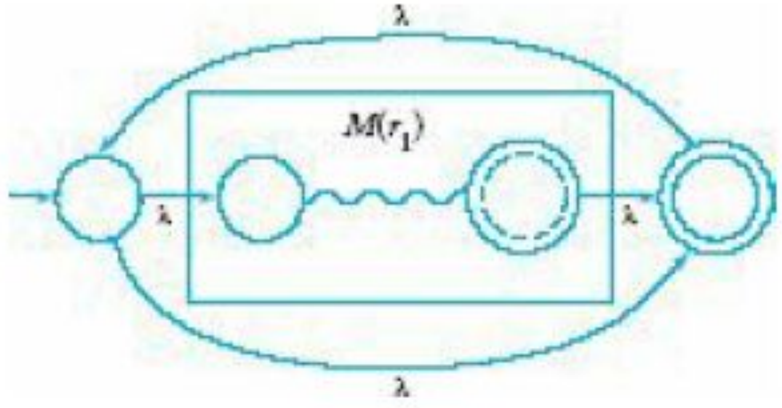


Automaton for $L(r_1 + r_2)$.

Regular Expressions and Regular Languages



Automaton for $L(r_1 r_2)$.



Automaton for $L(r_1^*)$.

Exercise

Find an nfa that accepts $L(r)$, where

$$r=(a + bb)^* \, (ba^* + \lambda)$$

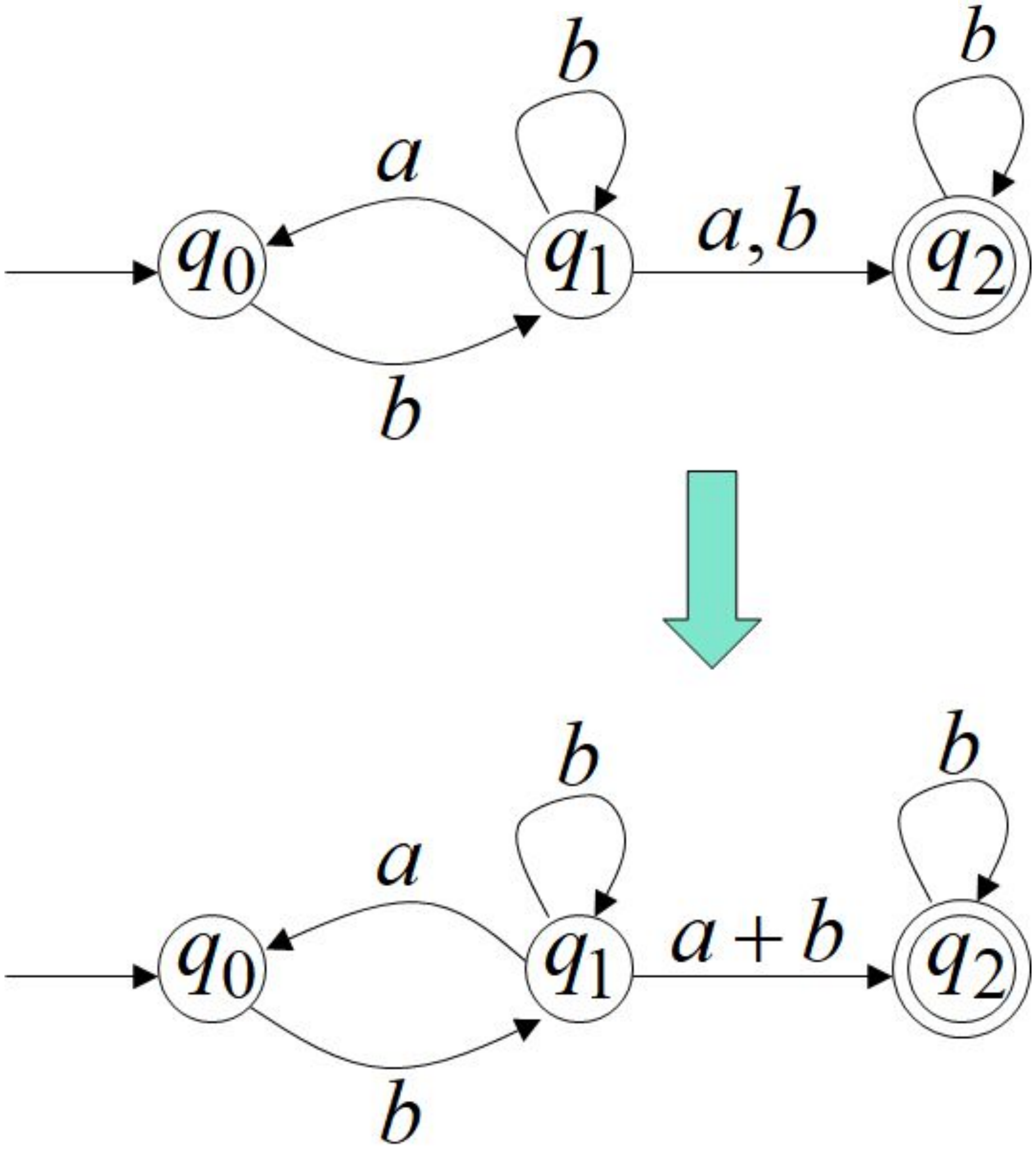
Regular Expressions for Regular Languages

for every regular language, there should exist a corresponding regular expression.

- For every regular language L there should be an NFA M that accepts it.
- From the NFA, we can extract the respective regular expression using **generalized transition graphs (GTG)**

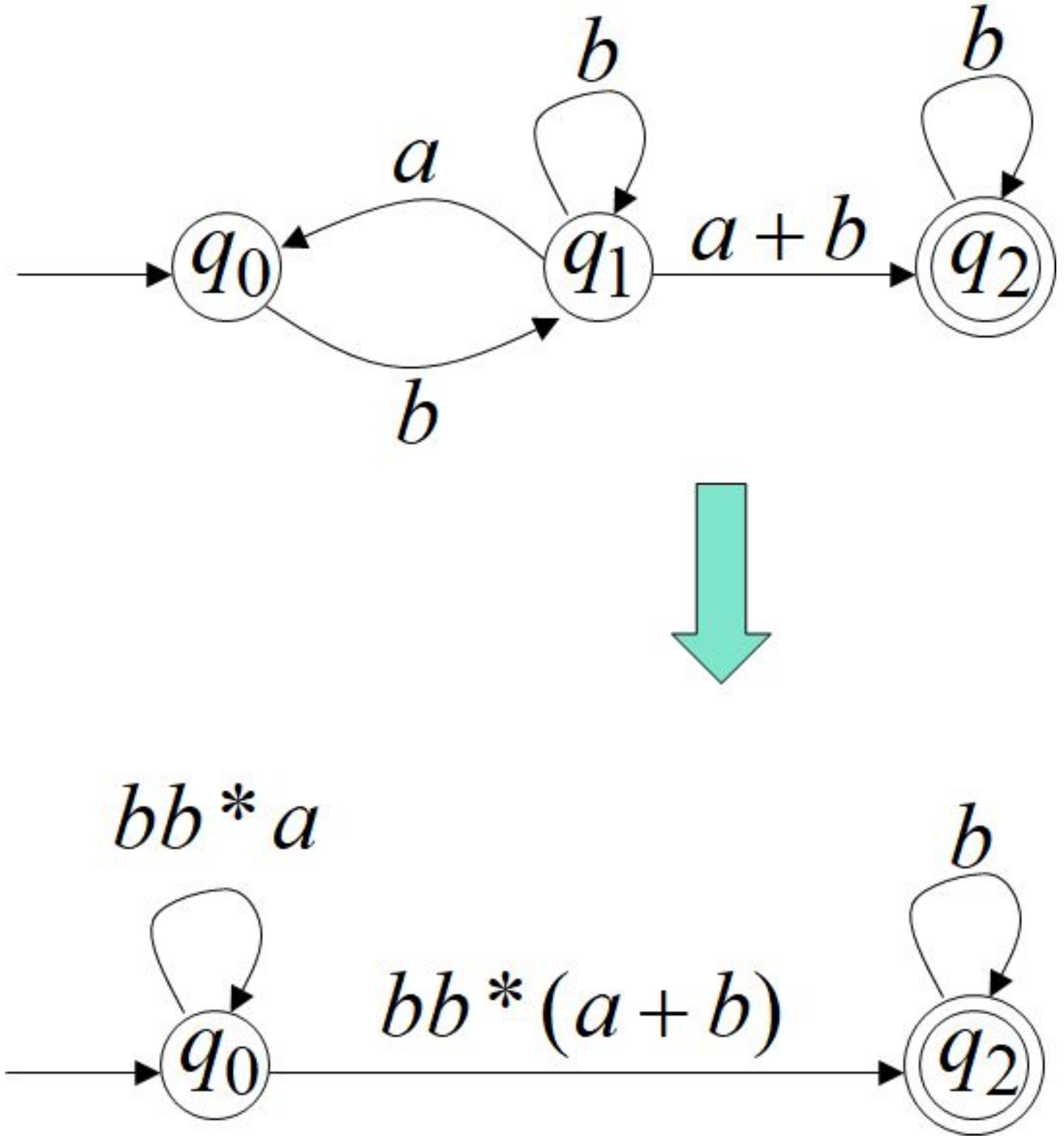
From M construct the equivalent Generalized Transition Graph in which transition labels are regular expressions.

Example



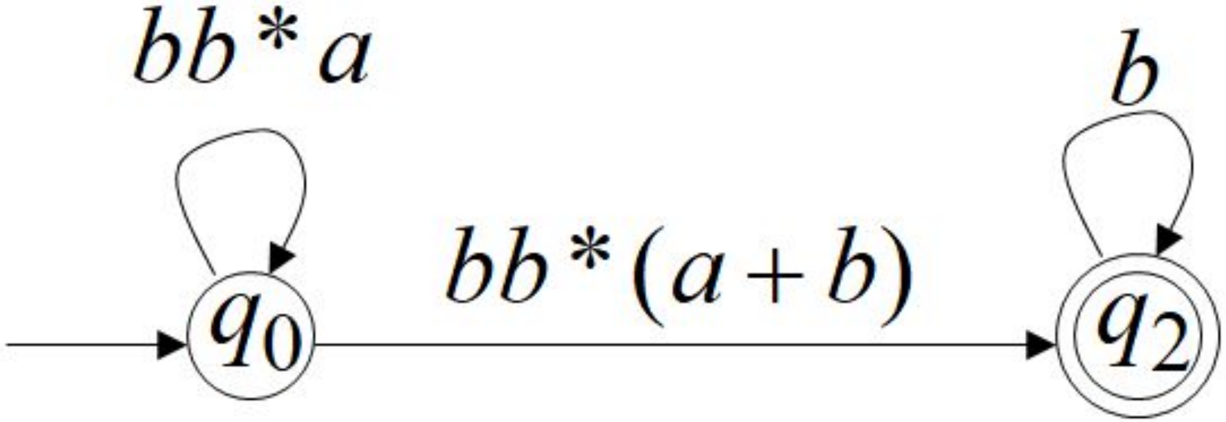
Reducing the states

Reducing the states



Example (cont.)

Resulting Regular Expression:

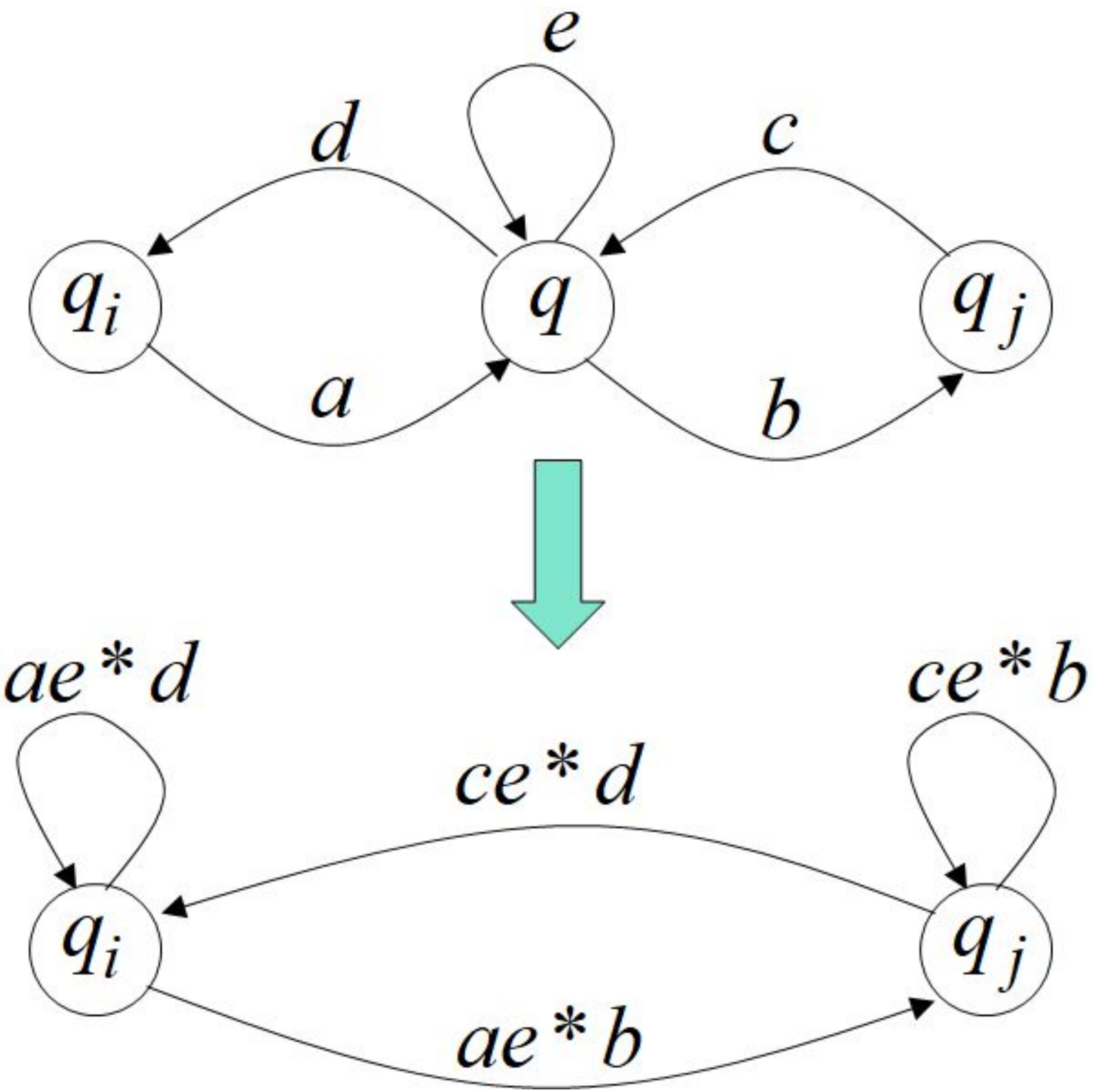


$$r = (bb^*a)^*bb^*(a+b)b^*$$

$$L(r) = L(M) = L$$

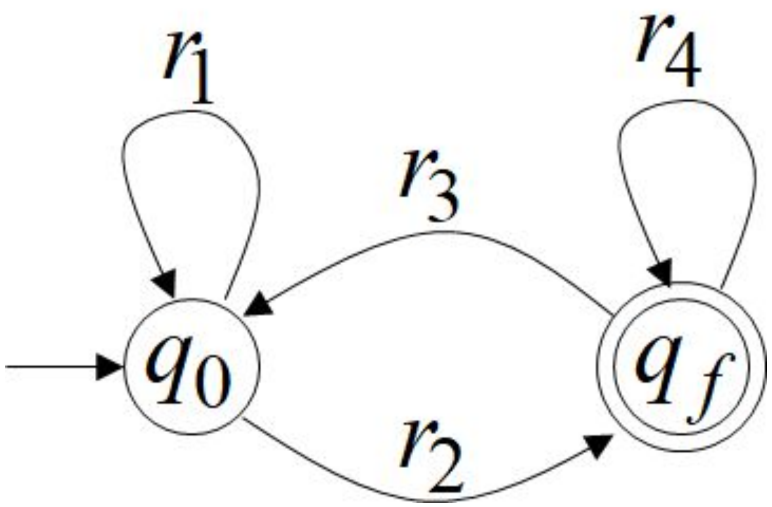
In General

Removing states:



In General

The final transition graph:



The resulting regular expression:

$$r = r_1 * r_2 (r_4 + r_3 r_1 * r_2) *$$

$$L(r) = L(M) = L$$

Example

Find a regular expression for the language

$$L = \{w \in \{a, b\}^* : n_a(w) \text{ is even and } n_b(w) \text{ is odd}\}.$$

Steps:

- Build NFA that accepts this language
- Generate the complete GTG by adding edges between each pair of states in the NFA
- Use the described state reduction procedure to reach the final transition graph, then apply the equation in the previous slide to find the regular expression.

- [Note: in Section 2.3, students can find a detailed procedure on this **nfa-to-rex** process]