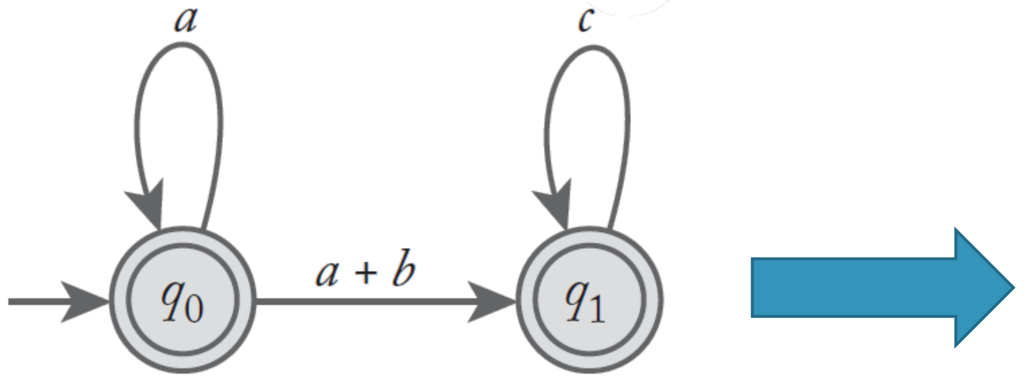
Theory of Machines and Languages

Fatemeh Deldar

1403-1404

Connection Between Regular Expressions and Regular Languages

- **Regular Expressions for Regular Languages**
  - **For every regular language, there should exist a corresponding regular expression**
- **Generalized transition graph (CTG)**
  - **A generalized transition graph is a transition graph whose edges are labeled with regular expressions**
- **Example**

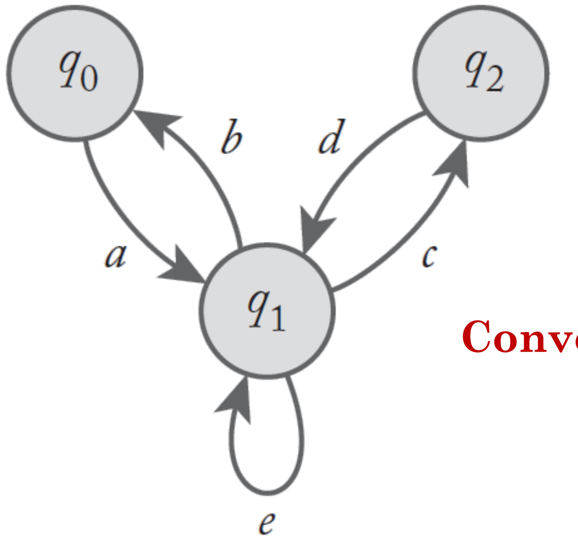


$$L\left(a^* + a^*\left(a+b\right)c^*\right)$$

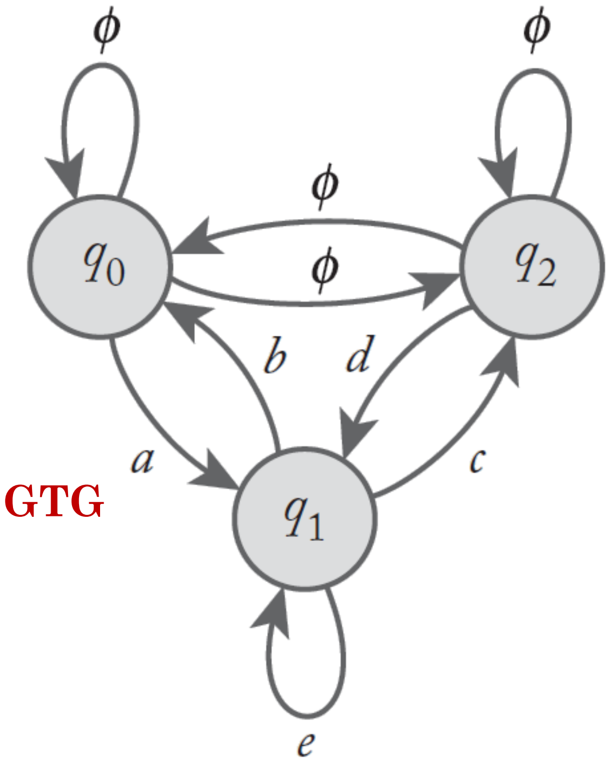
Connection Between Regular Expressions and Regular Languages

- A complete GTG is a graph in which all edges are present
  - We put missing edges and label them with ∅

- **Example**

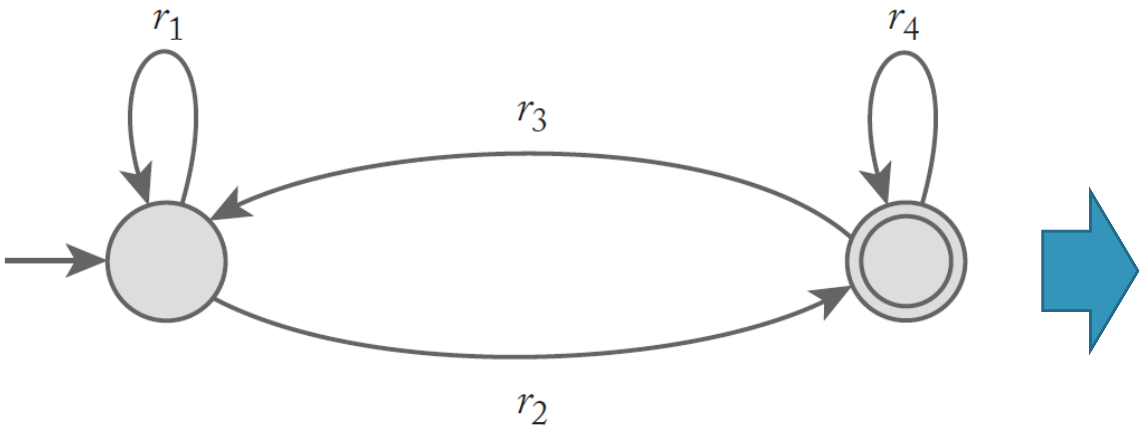


Convert to the complete GTG



Connection Between Regular Expressions and Regular Languages

- A two-state complete GTG

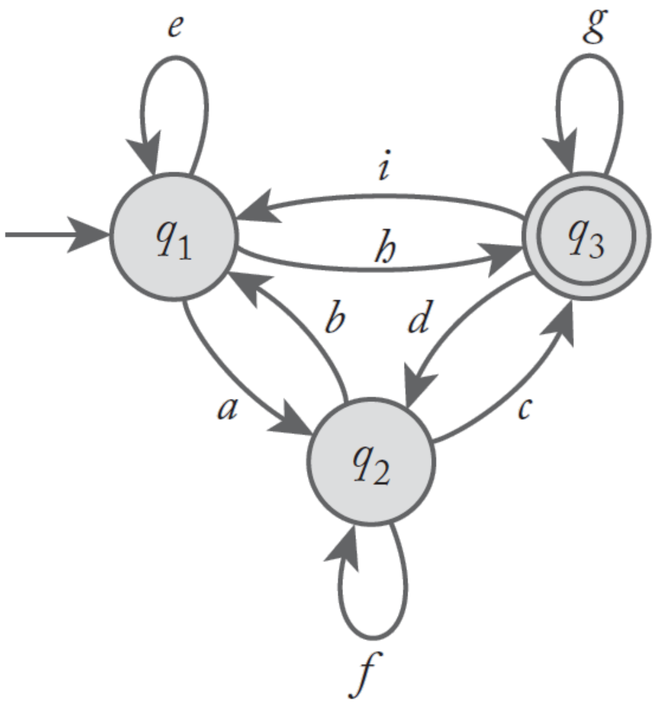


$$r = r_1^* r_2 (r_4 + r_3 r_1^* r_2)^*$$

- When a GTG has more than two states, we can ***remove one state at a time*** until only two states are left

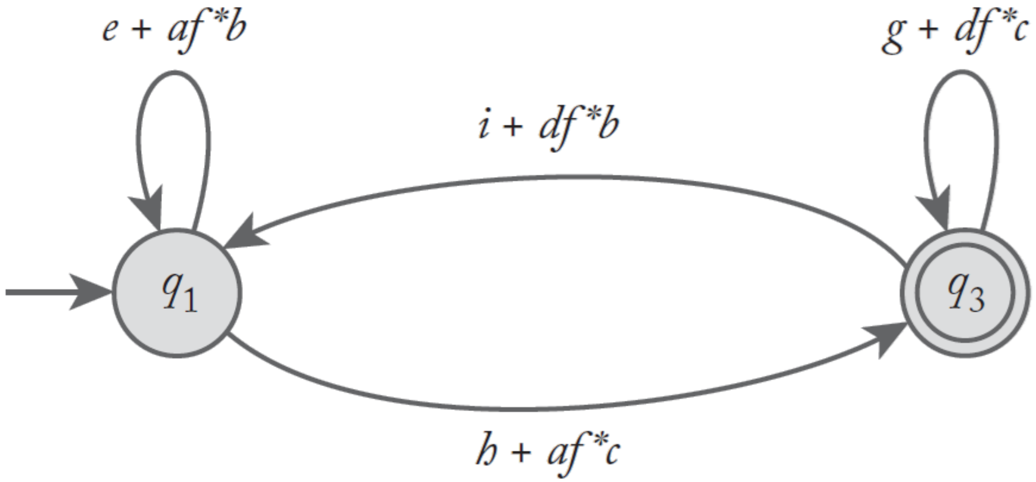
Connection Between Regular Expressions and Regular Languages

- A three-state complete GTG



To remove q_2:

- Create an edge from q_1 to q_1 and label it $e + af^*b$
- Create an edge from q_1 to q_3 and label it $h + af^*c$
- Create an edge from q_3 to q_1 and label it $i + df^*b$
- Create an edge from q_3 to q_3 and label it $g + df^*c$

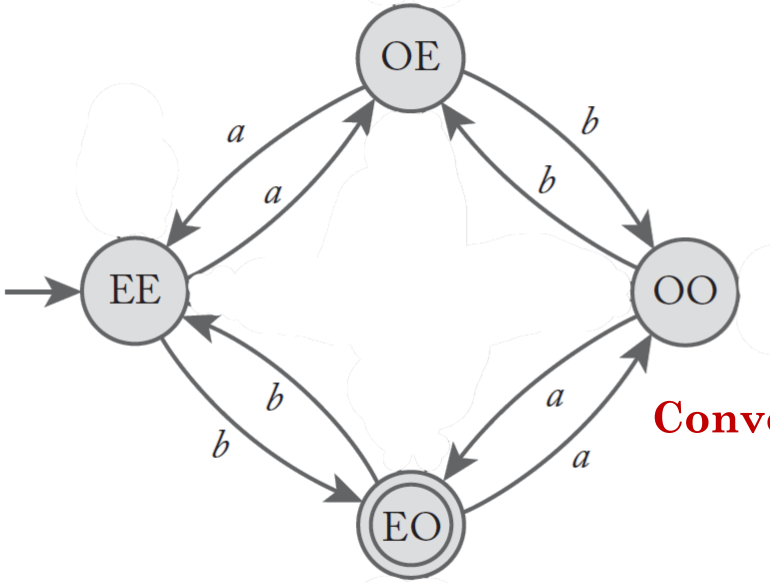


Connection Between Regular Expressions and Regular Languages

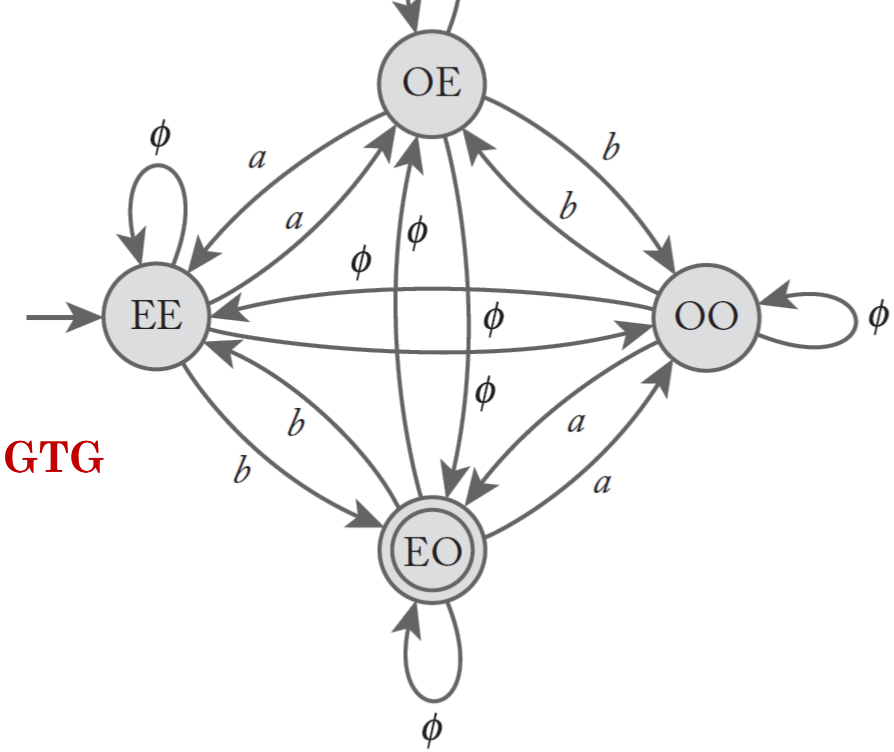
- **Example**

Find a regular expression for the language

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

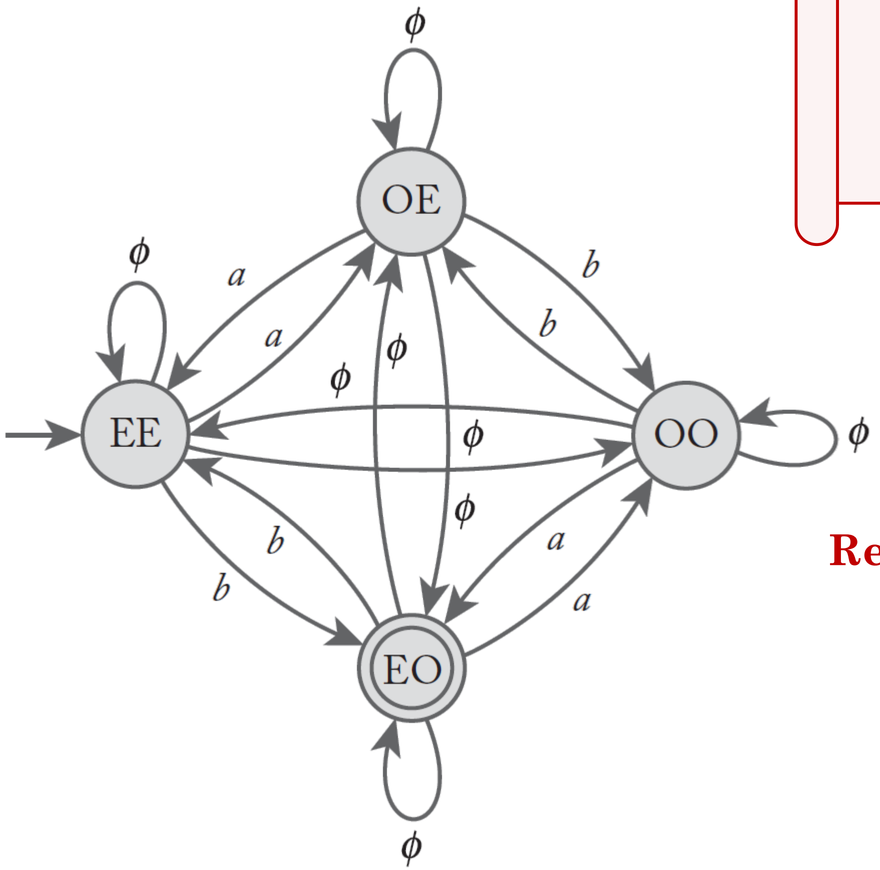


Convert to the complete GTG



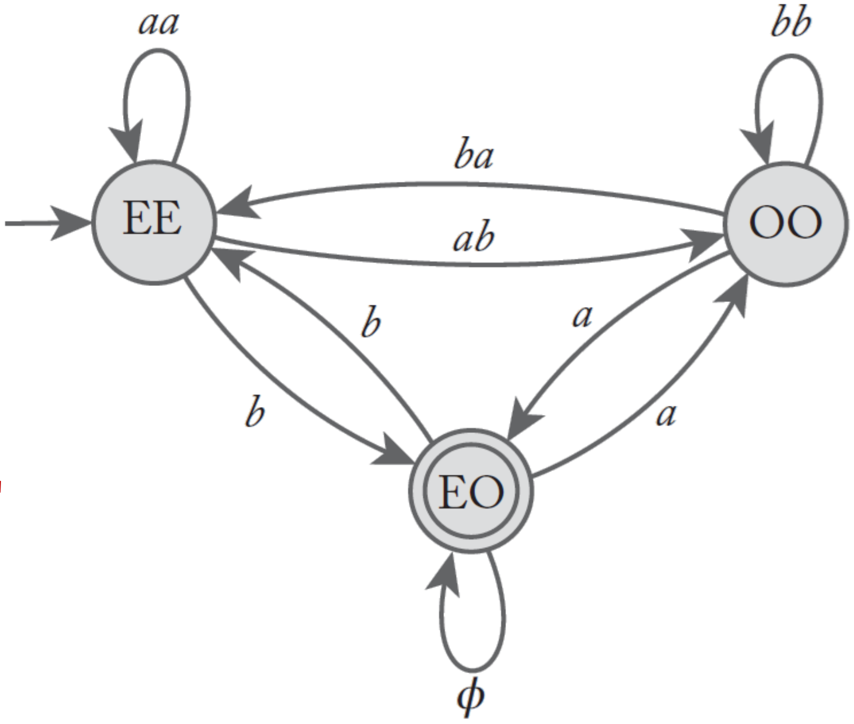
Connection Between Regular Expressions and Regular Languages

- **Example (Cont.)**



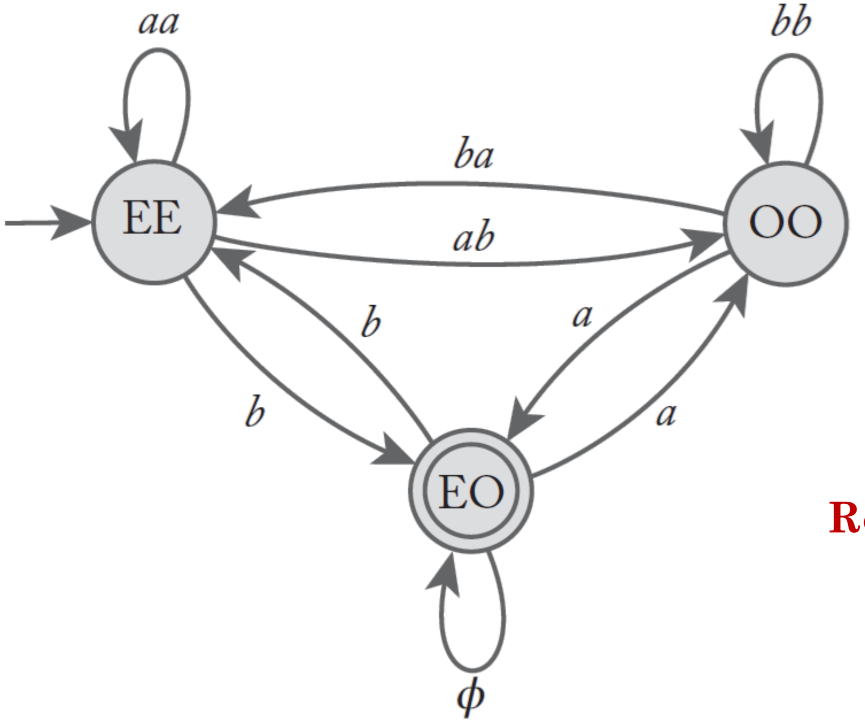
$$r + \varnothing = r,$$
$$r\varnothing = \varnothing,$$
$$\varnothing^* = \lambda,$$

Remove the state OE

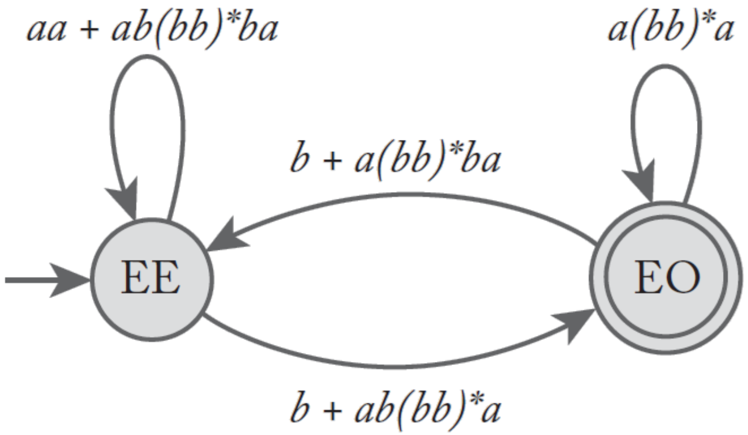


Connection Between Regular Expressions and Regular Languages

- **Example (Cont.)**



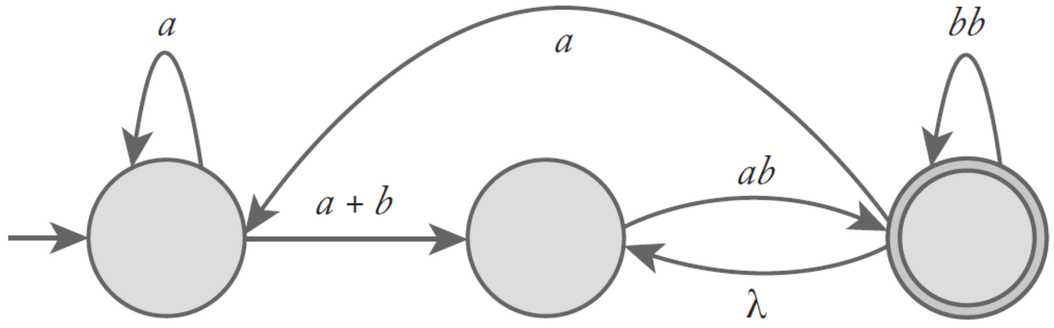
Remove the state OO



Connection Between Regular Expressions and Regular Languages

- **Example**

Consider the following generalized transition graph.



(a) Find an equivalent generalized transition graph with only two states.

(b) What is the language accepted by this graph?

Regular Grammars

- A grammar $G = (V, T, S, P)$ is said to be **right-linear** if all productions are of the form

$$A \to xB,$$
$$A \to x,$$

where $A,\ B \in V$, and $x \in T^*$. A grammar is said to be **left-linear** if all productions are of the form

$$A \to Bx,$$

or

$$A \to x.$$

A **regular grammar** is one that is either right-linear or left-linear.

Regular Grammars

- **Example**

The grammar $G_1 = (\{S\}, \{a, b\}, S, P_1)$, with P_1 given as

$$S \rightarrow abS|a$$

is $\boxed{\text{right-linear.}}$ The grammar $G_2 = (\{S, S_1, S_2\}, \{a, b\}, S, P_2)$, with productions

$$\begin{aligned} S &\rightarrow S_1ab, \\ S_1 &\rightarrow S_1ab|S_2, \\ S_2 &\rightarrow a, \end{aligned}$$

is $\boxed{\text{left-linear.}}$ $\boxed{\text{Both } G_1 \text{ and } G_2 \text{ are regular grammars.}}$

Regular Grammars

- **Example**

$$S \to A,$$
$$A \to aB|\lambda,$$
$$B \to Ab$$

- Although every production is either in right-linear or left-linear form, the grammar itself is neither right-linear nor left-linear, and therefore **is not regular**
- But this grammar is a **linear grammar**
  - A linear grammar is a grammar in which at most one variable can occur on the right side of any production, without restriction on the position of this variable
- ***A regular grammar is always linear, but not all linear grammars are regular***