CS 4410

Automata, Computability, and Formal Language

Dr. Xuejun Liang

Spring 2019

Chapter 3

Regular Languages and Regular Grammars

1. Regular Expressions
   - Formal Definition of a regular Expression
   - Languages Associated with Regular Expressions
2. Connection Between Regular Expressions and Regular Languages
   - Regular Expressions Denote Regular Languages
   - Regular Expressions for Regular Languages
   - Regular Expressions for Describing Simple Patterns
3. Regular Grammars
   - Right- and Left-Linear Grammars
   - Right-Linear Grammars Generate Regular Languages
   - Right-Linear Grammars for Regular Languages
   - Equivalence Between Regular Languages and Regular grammars

Learning Objectives

At the conclusion of the chapter, the student will be able to:

- Identify the language associated with a regular expression
- Find a regular expression to describe a given language
- Construct a nondeterministic finite automaton to accept the language denoted by a regular expression
- Use generalized transition graphs to construct a regular expression that denotes the language accepted by a given finite automaton
- Identify whether a particular grammar is regular
- Construct regular grammars for simple languages
- Construct a nfa that accepts the language generated by a regular grammar
- Construct a regular grammar that generates the language accepted by a finite automaton

Regular Expression

Definition 3.1

Let Σ be a given alphabet. Then

1. $\varnothing$, λ, and $a \in \Sigma$ are all regular expressions. These are called primitive regular expressions.
2. If r_1, r_2 and r are regular expressions, so are $r_1 + r_2$, $r_1 \bullet r_2$, r^*, and (r).
3. A string is a regular expression if and only if it can be derived from the primitive regular expressions by a finite number of applications of the rules in (2).

Example 3.1

For $\Sigma = \{a, b, c\}$,
the string $(a+b \bullet c)^* \bullet (c + \varnothing)$ is a regular expression,
but, the string $(a+b+)$ is not.

Languages Associated with Regular Expressions

Definition 3.2

The language L(r) denoted by any regular expression r is defined by the following rules.

1. $\varnothing$ is a regular expression denoting the empty set,
2. λ is a regular expression denoting $\{\lambda\}$,
3. For every $a \in \Sigma$, a is a regular expression denoting $\{a\}$.

If r_1, r_2 and r are regular expressions, then

4. $L(r_1 + r_2) = L(r_1) \cup L(r_2)$
5. $L(r_1 \bullet r_2) = L(r_1)\, L(r_2)$
6. $L((r)) = L(r)$
7. $L(r^*) = (L(r))^*$

Precedence rule

Star-closure: *
Concatenation: •
Union: +
Note: • can be omitted.

Sample Regular Expressions and Associated Languages

Regular Expression	Language
(ab)*	{ $(ab)^n$, $n \geq 0$ }
a + b	{ a, b }
(a + b)*	{ a, b }* (in other words, any string formed with a and b)
a(bb)*	{ a, abb, abbbb, abbbbbb, … }
a*(a + b)	{ a, aa, aaa, …, b, ab, aab, … } (Example 3.2)
(aa)*(bb)*b	{ b, aab, aaaab, …, bbb, aabbb, … } (Example 3.4)
(0 + 1)*00(0 + 1)*	Binary strings containing at least one pair of consecutive zeros

Example 3.2

L(a* • (a+b)) = ?

Languages and Regular Expressions

Example 3.3 Let r = (a+b)* (a + bb). L(r) = ?

Example 3.4 Let r = (aa)* (bb)* b. L(r) = ?

Example 3.5 For $\Sigma = \{0, 1\}$, give a regular expression r such that

L(r) = { w∈{0,1}*: w has at least one pair of consecutive zeros}

Example 3.6 Find a regular expression for the language

L(r) = { w∈{0,1}*: w has no pair of consecutive zeros}

We say the two regular expressions are equivalent if they denote the same language.

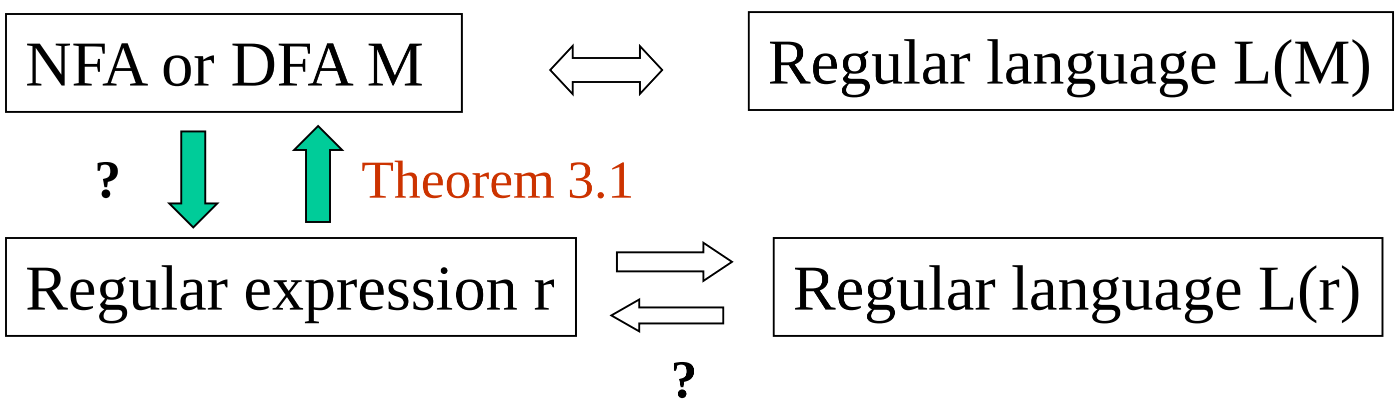
Regular Expressions Denote Regular Languages

Theorem 3.1: Let *r* be a regular expression. Then there exists some nondeterministic finite accepter that accepts L(*r*). Consequently, L(*r*) is a regular language.

Example 3.7

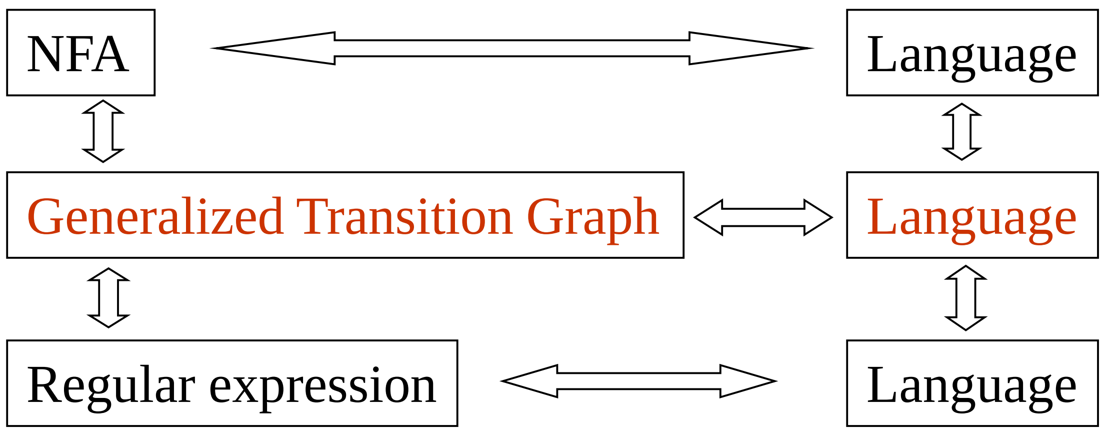
Find an nfa which accepts L(r), where

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



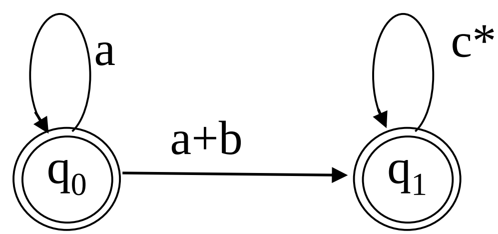
Generalized Transition Graph

In generalized transition graph, edges are regular expressions



Example 3.8

Find the language accepted by the generalized transition graph

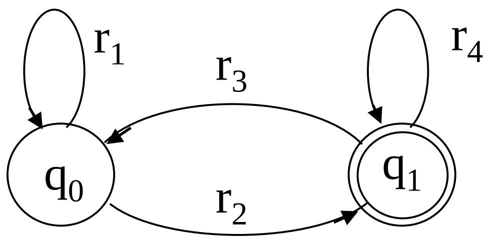


Regular Expressions for Regular Languages

Theorem 3.2: Let L be a regular language. Then there exists a regular expression r such that $L(r) = L$.

Proof Ideals

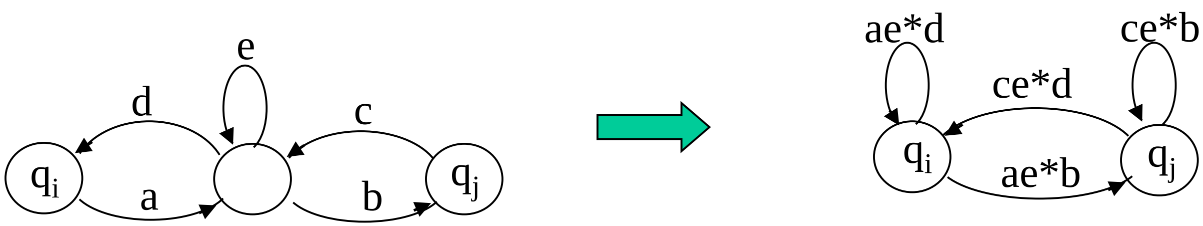
1. Let an NFA M accept L. Assume M has only one final state that is different with the initial state.
2. Convert M to an equivalent generalized transition graph by removing all states except the initial state and the final state.



3. The regular expression is

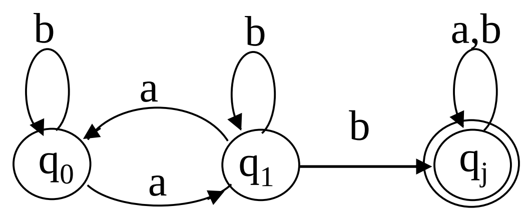
$$r = r_1{}^*r_2(r_4+r_3r_1{}^*r_2)^*$$

Transition Graph → Generalized Transition Graph



Transition Graph Generalized Transition Graph

Example 3.9:
Convert the nfa to generalized transition graph



Example: Find a regular expression for the language

$$L(r) = \{ w \in \{0,1\}^*: \text{w has no pair of consecutive zeros}\}$$

Describing Simple Patterns by Regular Expressions /*aba*c*/

Regular Grammar

Definition 3.3: A grammar G=(V, T, S, P) is said to be right-linear if all productions are of the form

$A \rightarrow xB$
$A \rightarrow x$

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

$A \rightarrow Bx$
$A \rightarrow x$

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

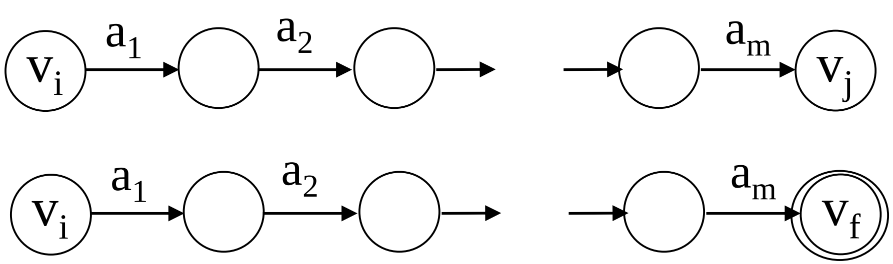
Example 3.13: $G_1=(\{S\}, \{a,b\}, S, P_1)$, and $S \rightarrow abS \mid a$. $L(G_1)=?$
$G_2=(\{S, S_1, S_2\}, \{a,b\}, S, P_2)$, and $S \rightarrow S_1ab$, $S_1 \rightarrow S_1ab \mid S_2$, $S_2 \rightarrow a$. $L(G_2)=?$

Example 3.14: $G=(\{S, A, B\}, \{a,b\}, S, P)$, and $S \rightarrow A$, $A \rightarrow aB|\lambda$, $B \rightarrow Ab$

A linear grammar is a grammar in which at most one variable can occur on the right side of any production.

Regular Grammar and Regular Language

Theorem 3.3: Let G=(V, T, S, P) be a right-linear grammar. Then G(L) is a regular language.



⟺ $V_i \rightarrow a_1a_2\ldots a_mV_j$

⟺ $V_i \rightarrow a_1a_2\ldots a_m$

Example 3.15: Construct a finite automaton that accepts the language generated by the grammar

$$V_0 \rightarrow aV_1$$
$$V_1 \rightarrow abV_0 \mid b$$

Theorem 3.4: If L is a regular language on alphabet Σ. Then there exists a right-linear grammar G=(V, T, S, P) such that L=L(G).

Regular Grammar and Regular Language

Example 3.16: Construct a right-linear grammar for L(aab*a).