# Theory of Computation Slides
# based on Michael Sipser's Textbook

Moses A. Boudourides[1]

Visiting Associate Professor of Computer Science
Haverford College

[1] Moses.Boudourides@cs.haverford.edu

**Section 1.3**

*Regular Expressions*

February 1, 2022

# Regular Languages and Regular Expressions, I

## Definition of Regular Languages over Alphabet $\Sigma$

The family $\mathcal{R}$ of **regular languages** (or **regular sets**) over alphabet $\Sigma$ is defined recursively as follows:

1. The language $\varnothing$ is an element of $\mathcal{R}$ and, for every $\sigma \in \Sigma$, the language $\{\sigma\}$ is in $\mathcal{R}$.
2. For any languages $L_1, L_2$ in $\mathcal{R}$, the three languages $L_1 \cup L_2, L_1 \circ L_2$ and $L_1{}^*$ are elements of $\mathcal{R}$.

## Notation of Regular Expressions

Given a regular language $L$, the **regular expression** of $L$ is a notational representation of this language, in which

- set delimiters $\{\}$ in regular languages are replaced by parentheses $()$ in regular expressions and they are omitted whenever the rules of precedence allow it, and
- the union symbol $\cup$ in regular languages is replaced by the $+$ symbol in regular expressions.

# Regular Languages and Regular Expressions, II

## Examples of Regular Expressions of Regular Sets

| **Regular Set** | **Regular Expression** |
|---|---|
| $\varnothing$ | $\varnothing$ |
| $\{\varepsilon\}$ | $\varepsilon$ |
| $\{a\}$ or $\{aba\}$ etc. | $a$ or $aba$ etc. |
| $\{a, b\}^*$ | $(a+b)^*$ |
| $\{aa, bb\} \cup \{ab, ba\}$ | $aa + bb + ab + ba$ |

## Examples of Regular Expressions of Described Regular Languages

| **Regular Language** | **Regular Expression** |
|---|---|
| Strings beginning with an $a$ and followed only by $b$'s | $ab^*$ |
| Strings $x$ with $n_a(x) = 2$ | $b^*ab^*ab^*$ |
| Strings $x$ with $n_{aa}(x) \geq 1$ or $n_{bb}(x) \geq 1$ | $(a+b)^*(aa+bb)(a+b)^*$ |
| Strings $x$ ending in $b$ with $n_{aa}(x) = 0$ | $(b+ab)^+$ |

# Regular Languages and Regular Expressions, III

## Definition of Equality among Regular Expressions

Two **regular expressions are equal** if the languages (regular sets) they denote (describe) are equal.

## Proposition: Properties of Regular Expressions

Let $R, S$ and $T$ be regular expressions over $\Sigma$. Then:

1. $R + S = S + R, R + \varnothing = \varnothing + R, R + R = R,$ $(R + S) + T = R + (T + S),$
2. $R\varepsilon = \varepsilon R = R, R\varnothing = \varnothing R = \varnothing, (RS)T = R(ST)$ (note that generally $RS \neq SR$),
3. $R(S + T) = RS + RT, (S + T)R = SR + TR,$
4. $R^* = R^* R^* = (R^*)^* = (\varepsilon + R)^*, \varnothing^* = \varepsilon^* = \varepsilon,$
5. $R^* = \varepsilon + \sum_{j=1}^{k} R^j + R^{k+1} R^*$, for all $k \geq 1$ (special case: $R^* = \varepsilon + RR^*$),

# Regular Languages and Regular Expressions, IV

**Proposition: Properties of Regular Expressions (cont.)**

8. $(R+S)^* = (R^*+S^*)^* = (R^*S^*)^* = (R^*S)^*R^* = R^*(SR^*)^*$ (note that generally $(R+S)^* \neq R^*+S^*$),
9. $R^*R = RR^*, R(SR)^* = (RS)^*R$,
10. $(R^*S)^* = \varepsilon + (R+S)^*S, (RS^*)^* = \varepsilon + R(R+S)^*$,
11. **Arden Rule**: If $\varepsilon \notin S$, then

$$R = SR + T \text{ if and only if } R = S^*T,$$
$$R = RS + T \text{ if and only if } R = TS^*.$$

# Direct Derivation of Regular Expressions from Descriptions of Regular Languages

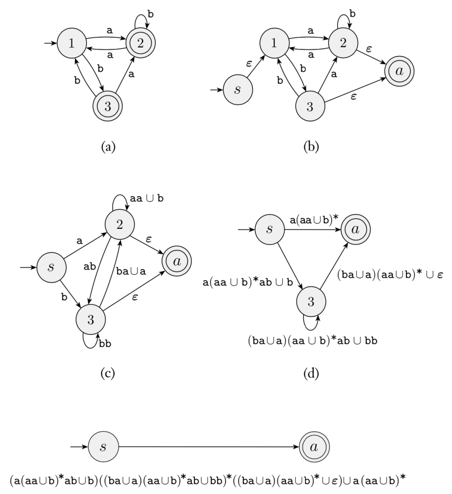
## Example

Find the regular expression for the language over $\{a, b\}$ of strings with an odd number of $a$'s.

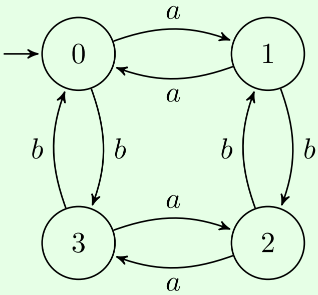
- Apparently, a string with an odd number of $a$'s should contain at least one $a$ and the additional $a$'s grouped into pairs.
- Thus, taking the single $a$ in the beginning of the string, the regular expression is:
$$b^*ab^*(ab^*ab^*)^*.$$
- Another way to group pairs of $a$'s is considering them produced by the star closure of $ab^*a$ and $b$'s, i.e., another correct regular expression is:
$$b^*a(b + ab^*a)^*.$$
- Furthermore, by considering the single $a$ to be placed at the end of these strings, we get two more correct regular expressions:
$$(b^*ab^*a)^*b^*ab^*, (b + ab^*a)^*ab^*.$$

# Derivation of Regular Expressions given a DFA: The Method of Removal of States and Replacement of Transitions

## Example



## Example




EVEN–EVEN: $F = \{0\}$ accepted are $n_a = 0 \bmod 2$ and $n_b = 0 \bmod 2$.

ODD–EVEN: $F = \{1\}$ accepted are $n_a = 1 \bmod 2$ and $n_b = 0 \bmod 2$.

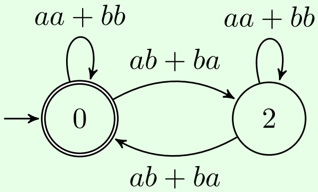
ODD–ODD: $F = \{2\}$ accepted are $n_a = 1 \bmod 2$ and $n_b = 1 \bmod 2$.

EVEN–ODD: $F = \{3\}$ accepted are $n_a = 0 \bmod 2$ and $n_b = 1 \bmod 2$.

Below, we are going to derive the regular expressions of these languages through the **method of removal of states and replacement of transitions by strings**, i.e., through the construction of an equivalent **transition graph**, which is a FA with strings as labels of transitions.

## EVEN–EVEN

By removing & replacing nodes 1 and 3 (**both together**), the following transition graph is obtained:

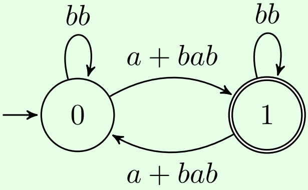



Thus, the regular expression describing the language EVEN–EVEN is:

$$R = (aa + bb + (ab + ba)(aa + bb)^*(ab + ba))^*.$$

## ODD–EVEN

By removing & replacing nodes 2 and 3 (**both together**), the following transition graph is obtained:

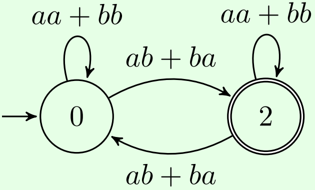



Thus, the regular expression describing the language ODD–EVEN is:

$$R = (bb)^*(a + bab)(bb + (a + bab)(bb)^*(a + bab))^*.$$

## ODD–ODD

By removing & replacing nodes 1 and 3 (**both together**), the following transition graph is obtained:

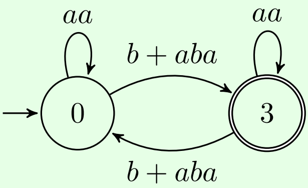



Thus, the regular expression describing the language ODD–ODD is:

$$R = (aa+bb)^*(ab+ba)(aa+bb+(ab+ba)(aa+bb)^*(ab+ba))^*.$$

## EVEN–ODD

By removing & replacing nodes 1 and 2 (**both together**), the following transition graph is obtained:




Thus, the regular expression describing the language EVEN–ODD is:

$$R = (aa)^*(b + aba)(aa + (b + aba)(aa)^*(b + aba))^*.$$

# Derivation of Regular Expressions given a DFA: Kleene's Algorithm on Regular Expressions of Paths

## Kleene's Algorithm on Regular Expressions of Paths

Let $M = (Q, \Sigma, q_0, F, \delta)$ be a DFA, assuming that $Q = \{1, 2, \ldots, n\}$ and $q_0 = 1$. For any positive integer $k \leq n$, denote by $R(i, j, k)$ a regular expression for the set of strings that $M$ accepts when starting at state $i$ and terminating at state $j$, only using states in the set $\{1, 2, \ldots, k\}$, i.e., without passing from any state $l > k$. Then, if $k < n$,

$$R(i,j,k+1) = R(i,j,k) + R(i,k+1,k)R(k+1,k+1,k)^* R(k+1,j,k),$$

where

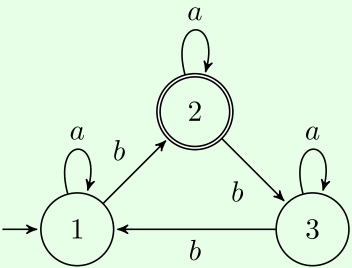
$$R(i,j,0) = \begin{cases} \{\sigma \in \Sigma \mid \delta(i,\sigma) = j\}, & \text{if } i \neq j, \\ \{\varepsilon\} \cup \{\sigma \in \Sigma \mid \delta(i,\sigma) = j\}, & \text{if } i = j, \end{cases}$$

and, moreover, the regular expression of the language accepted by $M$ is:

$$R = \bigcup_{f \in F} R(1, f, n).$$

## Example 1

Let $M$ be the following FA:




Calculating regular expressions from the to down, we get:

$$
\begin{aligned}
R &= R(1,2,3) \\
R(1,2,3) &= R(1,2,2) + R(1,3,2)R(3,3,2)^* R(3,2,2) \\
R(1,2,2) &= R(1,2,1) + R(1,2,1)R(2,2,1)^* R(2,2,1) \\
R(1,3,2) &= R(1,3,1) + R(1,2,1)R(2,2,1)^* R(2,3,1) \\
R(3,3,2) &= R(3,3,1) + R(3,2,1)R(2,2,1)^* R(2,3,1) \\
R(3,2,2) &= R(3,2,1) + R(3,2,1)R(2,2,1)^* R(2,2,1)
\end{aligned}
$$

## Example 1 (cont.)

$$
\begin{aligned}
R(1,2,1) &= R(1,2,0) + R(1,1,0)R(1,1,0)^*R(1,2,0)\\
R(2,2,1) &= R(2,2,0) + R(2,1,0)R(1,1,0)^*R(1,2,0)\\
R(1,3,1) &= R(1,3,0) + R(1,1,0)R(1,1,0)^*R(1,3,0)\\
R(2,3,1) &= R(2,3,0) + R(2,1,0)R(1,1,0)^*R(1,3,0)\\
R(3,3,1) &= R(3,3,0) + R(3,1,0)R(1,1,0)^*R(1,3,0)\\
R(3,2,1) &= R(3,2,0) + R(3,1,0)R(1,1,0)^*R(1,2,0)
\end{aligned}
$$

Now, we know the expressions for $k = 0$:

$$
\begin{array}{lll}
R(1,1,0) = \varepsilon + a, & R(1,2,0) = b, & R(1,3,0) = \varnothing\\
R(2,1,0) = \varnothing, & R(2,2,0) = \varepsilon + a, & R(2,3,0) = b\\
R(3,1,0) = b, & R(3,2,0) = \varnothing, & R(3,3,0) = \varepsilon + a
\end{array}
$$

## Example 1 (cont.)

Thus, substituting the values in the expressions for $k = 1$, we get:

$$R(1,2,1) = a^*b$$
$$R(2,2,1) = \varepsilon + a$$
$$R(1,3,1) = \varnothing$$
$$R(2,3,1) = b$$
$$R(3,3,1) = \varepsilon + a$$
$$R(3,2,1) = ba^*b$$

which result:

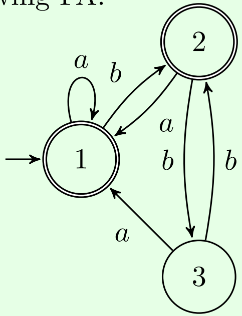
$$R(3,2,2) = ba^*ba^*$$
$$R(3,3,2) = \varepsilon + a + ba^*ba^*b$$
$$R(1,3,2) = a^*ba^*b$$
$$R(1,2,2) = a^*ba^*$$

Therefore, the wanted regular expression is:

$$R = R(1,2,3) = a^*ba^* + a^*ba^*b(a + ba^*ba^*b)^*ba^*ba^*.$$

## Example 2

Let $M$ be the following FA:




Calculating regular expressions from the to down, we get:

$$R = R(1,1,3) + R(1,2,3)$$
$$R(1,1,3) = R(1,1,2) + R(1,3,2)R(3,3,2)^*R(3,1,2)$$
$$R(1,2,3) = R(1,2,2) + R(1,3,2)R(3,3,2)^*R(3,2,2)$$
$$R(1,1,2) = R(1,1,1) + R(1,2,1)R(2,2,1)^*R(2,1,1)$$
$$R(1,3,2) = R(1,3,1) + R(1,2,1)R(2,2,1)^*R(2,3,1)$$
$$R(3,3,2) = R(3,3,1) + R(3,2,1)R(2,2,1)^*R(2,3,1)$$
$$R(3,1,2) = R(3,1,1) + R(3,2,1)R(2,2,1)^*R(2,1,1)$$
$$R(1,2,2) = R(1,2,1) + R(1,2,1)R(2,2,1)^*R(2,2,1)$$
$$R(3,2,2) = R(3,2,1) + R(3,2,1)R(2,2,1)^*R(2,2,1)$$

## Example 2 (cont.)

Thus, from the table of regular expressions for $n = 0$:

| $i$ | $R(i,1,0)$ | $R(i,2,0)$ | $R(i,3,0)$ |
|---|---|---|---|
| 1 | $\varepsilon + a$ | $b$ | $\varnothing$ |
| 2 | $a$ | $\varepsilon$ | $b$ |
| 3 | $a$ | $b$ | $\varepsilon$ |

we obtain the values in the following two tables:

| $i$ | $R(i,1,1)$ | $R(i,2,1)$ | $R(i,3,1)$ |
|---|---|---|---|
| 1 | $a^*$ | $a^*b$ | $\varnothing$ |
| 2 | $aa^*$ | $\varepsilon + aa^*b$ | $b$ |
| 3 | $aa^*$ | $a^*b$ | $\varepsilon$ |

| $i$ | $R(i,1,2)$ | $R(i,2,2)$ | $R(i,3,2)$ |
|---|---|---|---|
| 1 | $a^*(baa^*)^*$ | $a^*(baa^*)^*b$ | $a^*(baa^*)^*bb$ |
| 2 | $aa^*(baa^*)^*$ | $(aa^*b)^*$ | $(aa^*b)^*b$ |
| 3 | $aa^* + a^*baa^*(baa^*)^*$ | $a^*b(aa^*b)^*$ | $\varepsilon + a^*b(aa^*b)^*b$ |

Substituting the values in the expressions for $k = 3$ from the above tables, we find the final form of the regular expression for $M$. (Algebra omitted.)