Theory of Computation

RegExp /DFA_Properties

Lecture 10

Gaurav Raj

Symbols Reminder

- + → Union / OR
- . → Concatenation
- ε → Empty string
- Ø → Empty language
- R, S, T → Regular expressions

Properties of + (OR / Union)

Identity
Identity element does **not change** the expression
R+Ø=R

Explanation:
Union with an empty language gives the same language.

Commutative
Order does **not matter**
R+S=S+R

Example:
a + b = b + a

Associative
Grouping does **not matter**
(R+S)+T=R+(S+T)

Example:
(a + b) + c = a + (b + c)

Idempotent
Repeating does not change result
R+R=R

Example:
a + a = a

Dominator (Absorbing Element)
Σ^* dominates union
$R+\Sigma^*=\Sigma^*$
Because Σ^* already contains all strings.

Annihilator
NO annihilator exists for +

Because nothing makes union empty except both being empty.

Properties of . (Concatenation)

Identity
Identity element keeps expression same
$R.\varepsilon=\varepsilon.R=R$

Explanation:
Concatenating with empty string does nothing.

Associative
Grouping does **not matter**
$(R.S).T=R.(S.T)$

Example:
(ab)c = a(bc)

Commutative (NOT TRUE)

Order **matters**
$R.S \neq S.R$

Example:
$ab \neq ba$

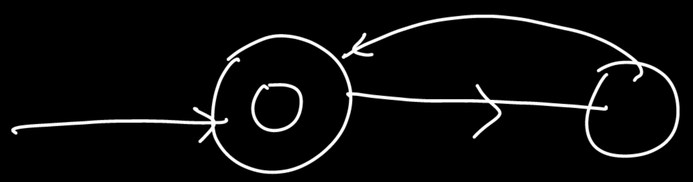
Annihilator
Element that destroys everything
$R.\emptyset = \emptyset.R = \emptyset$
Explanation:
No strings available to concatenate.

Dominator
No dominator for concatenation

Find All The languages of given regular expression

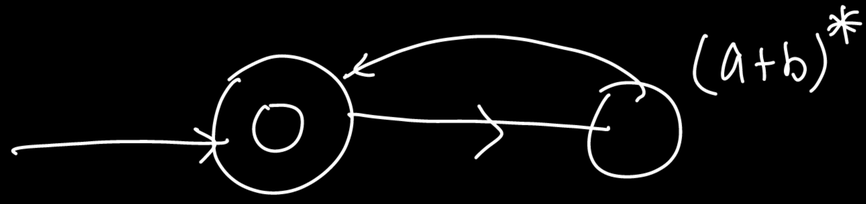
$\Sigma = \{a, b\}$

1. aa + ab + ba + bb
2. $(a+b)^+$ $\{a, b, \ldots\}$
3. $(\varepsilon + a + b)^2$ $L(\omega) \leq 2$ $(\varepsilon + a + b).(\varepsilon + a + b) = \{\varepsilon, a, b, aa, ab\}$
4. $(a + b)^2.(a + b)^*$ $L(\omega) \geq 2$ $(\varepsilon, a, b..)$
5. $(\Sigma^2)^*$
6. $(\Sigma\Sigma)^*$ $\Sigma'\Sigma'$ $[(a+b)(a+b)]^*$ $L(\omega) =$ even
7. $(\Sigma\Sigma\Sigma)^*\,\Sigma$ $(aa, ab, ba, bb)^*$
8. $(\Sigma\Sigma\Sigma\Sigma\Sigma)^*$



Find All The languages of given regular expression

- ~~aa + ab + ba + bb~~ $\Sigma = \{a, b\}$
- $(a+b)^+$
- $(\varepsilon + a + b)^2$ $\{a, b, \dots\}$
- $(a + b)^2.(a + b)^*$ $L(\omega) \leq 2$
- $(\Sigma^2)^*$
- $(\Sigma\Sigma)^*$
- $(\Sigma\Sigma\Sigma)^* \Sigma$
- $(\Sigma\Sigma\Sigma\Sigma\Sigma)^*$



$(\varepsilon + a + b).(\varepsilon + a + b) = \{\varepsilon, a, b, aa, ab\}$

$(\Sigma\Sigma\Sigma)^* \cdot [(a+b)(a+b)(a+b)]^*$

$[aaa, aab, aba, abb, \dots]^* \cdot (a+b)^1$

$1 \% 3 = 1$

$4 \% 3 = 1$

$7 \% 3 = 1$

$\Sigma = \{a, b\}$ $\{0, 1\}$ $\Sigma = \{\phi, 1, 0, a, b, \varepsilon\}$

$\phi = \{\ \}$

$\Sigma = \{01, 10\}$

Set $\leftarrow \{0\}$

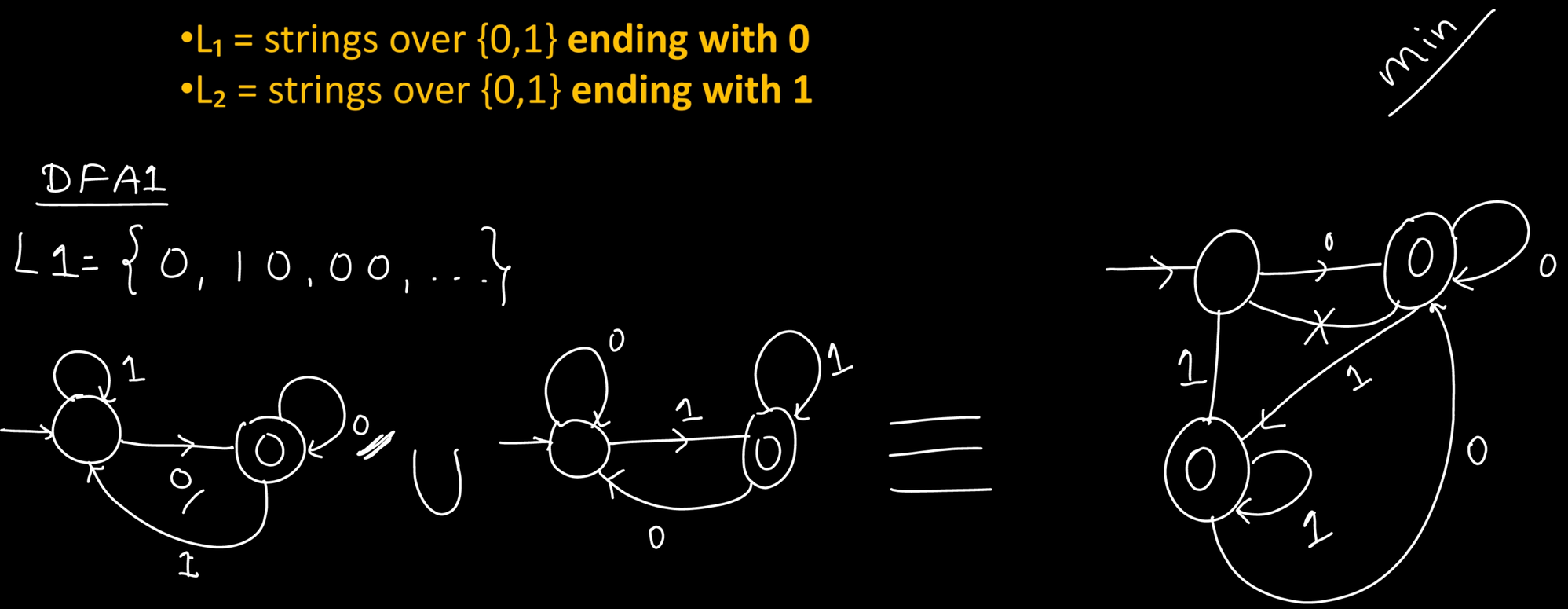
$\{\ \}$

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

	Length	set	String	Symbol	RegExp
ε	0	X	✓	X	yes
0	1	X	✓	✓	yes
1	1	X	✓	✓	yes
Ø	0	✓	X	X	yes
01	2	X	✓	X	yes
{01,10}	2	✓	X	✓	No

DFA : Union

- L_1 = strings over {0,1} **ending with 0**
- L_2 = strings over {0,1} **ending with 1**



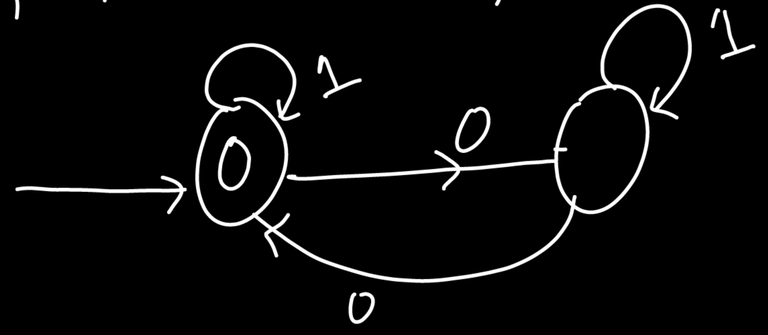
DFA : Concatenation

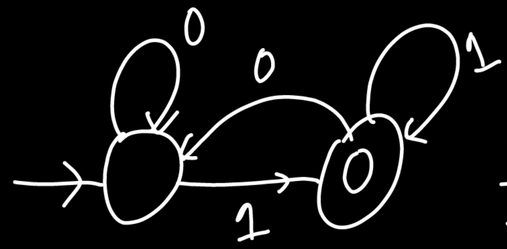
L_1 = { strings with **even number of 0s** }
L_2 = { strings ending with **1** }

$L_1 \cdot L_2$

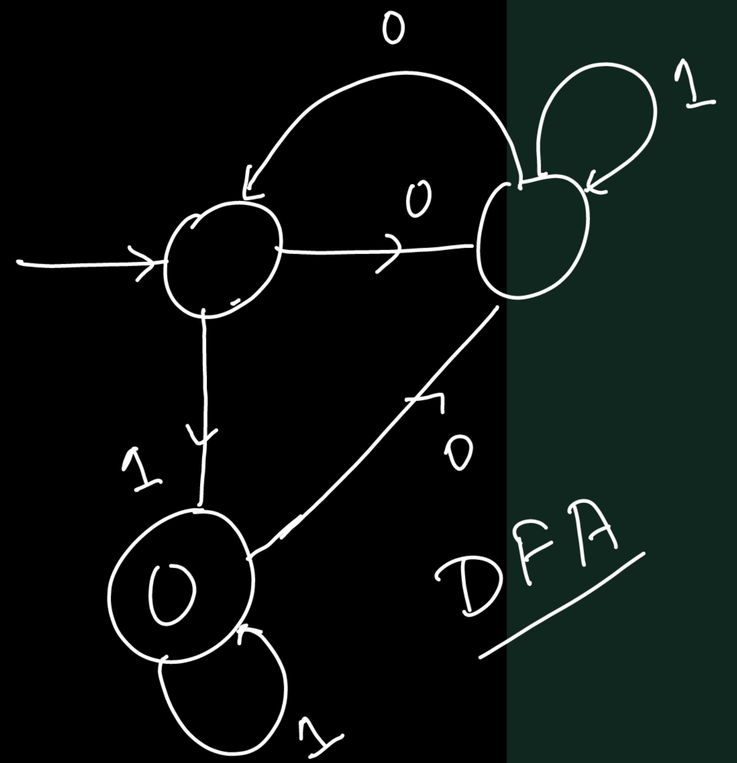
$L_3 = L_1 \cdot L_2$

~~$L_4 = L_2 \cdot L_1$~~





≡



DFA

DFA : Cross-Product

L_1 = strings with **even number of 0s**
L_2 = strings with **even number of 1s**

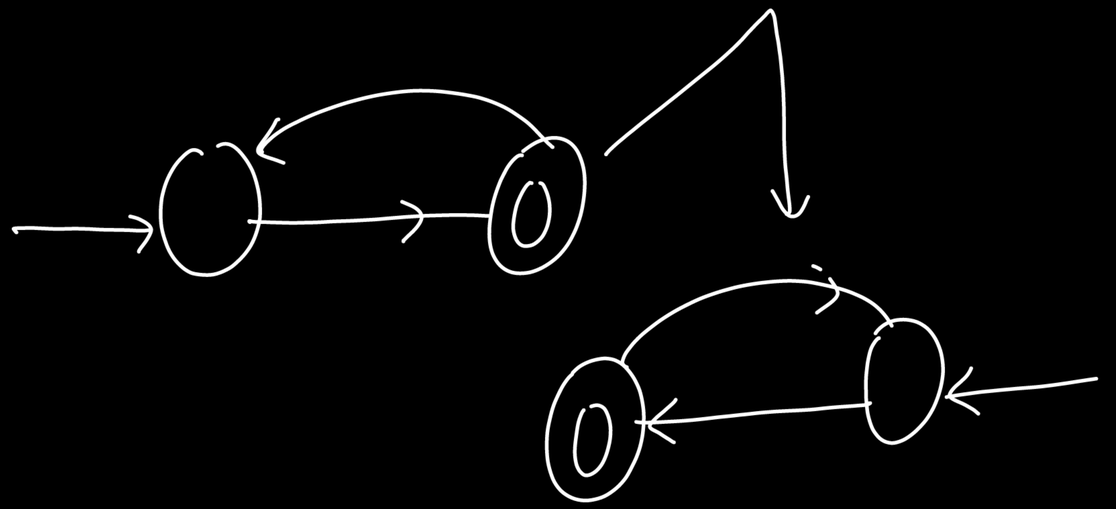
DFA : Complement

L = strings over {0,1} starts with 1 **ending with 0**

DFA : Reversal

If L = strings ending with 01
then L^R = strings starting with 10

If a language **L** contains strings:
L={w1,w2,w3,… }
Then the **reverse language** is:
L^R={$w1^R$,$w2^R$,$w3^R$,… }



Thank You