

AUTOMATA FORMAL LANGUAGES AND LOGIC



Lecture notes on Equivalence of Regular expressions

Prepared by

**Ms. Kavitha K N
Assistant Professor**

**Ms. Preet Kanwal
Associate Professor**

**Department of Computer Science & Engineering
PES UNIVERSITY**

(Established under Karnataka Act No.16 of 2013)

100-ft Ring Road, BSK III Stage, Bangalore - 560 085

Table of Contents:

Section	Topic	Page number
1.	Equivalence of two regular expression	3
2.	Methods to determine the equivalence of Regular Expressions	3
3.	Examples	4

Examples Solved:

Section	Problems on Equivalence of two regular expression	Page number
1.	$R1 = (0+1)^*(0+\lambda)$ and $R2 = (1+\lambda)(1+0)^*$	4
2.	$R1 = (0+\lambda)(11^*0)^*(1+\lambda)$ and $R2 = (1+\lambda)(011^*)(0+\lambda)$	5
3.	$R1 = (1+\lambda)(00^*1)^*0^*$ and $R2 = (0+\lambda)(11^*0)1^*$	5
4.	$R1 = 0^*(10^*)^*$ and $R2 = (1^*0)^*1^*$	5

1. Equivalence of two regular expression

Two regular expressions(R_1 and R_2) are equivalent($R_1 = R_2$) iff:

$$L(R_1) = L(R_2)$$

2. Methods to determine the equivalence of Regular Expressions:

We can determine the equivalence using two methods

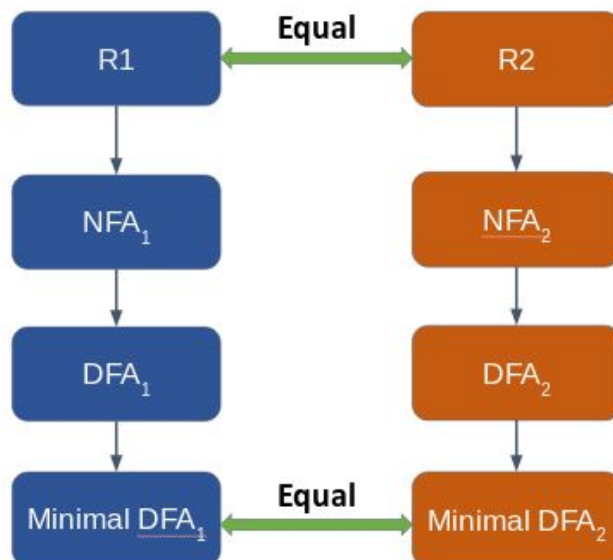
1) A Formal method

The classical approach to the problem of comparing two regular expressions α and β , i.e., deciding if $L(R_1) = L(R_2)$, typically consists of:

1. obtain non-deterministic finite automata, $N(R_1)$ and $N(R_2)$, which accept the same language as R_1 and R_2 , respectively using Thompson construction method.

2. convert the non-deterministic automata to equivalent deterministic ones, $D(R_1) \equiv N(R_1)$ and $D(R_2) \equiv N(R_2)$ using subset construction method

3. minimise both $D(R_1)$ and $D(R_2)$. Because, for a given regular language, the minimal automaton is unique

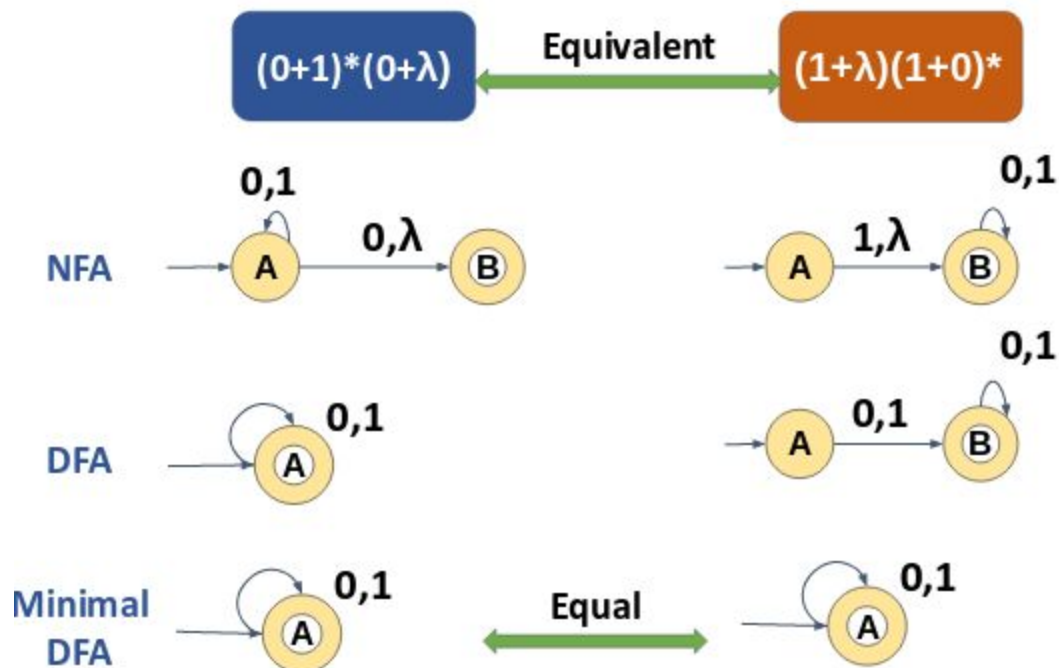


2) An Informal method

In Informal method, the aim is not to prove that the two regular expressions are equal. Instead we focus on proving that they are not equivalent. To show that two regular expressions are not equivalent we have to find a word that belongs to one expression and does not belong to the other.

3. Examples

- 1) Check the Equivalence of two regular expressions using formal method
 $R1 = (0+1)^*(0+\lambda)$ and $R2 = (1+\lambda)(1+0)^*$



Since both the minimal DFAs are equal We conclude that the given regex are equivalent.

- 2) Check the Equivalence of two regular expressions using informal method

$R1 = (0 + \lambda)(11^*0)^*(1 + \lambda)$ and $R2 = (1 + \lambda)(011^*)(0 + \lambda)$

Consider the String 011

$011 \in R2$

$011 \notin R1$

So,

$R1 \neq R2$

- 3) Check the Equivalence of two regular expression using informal method

$R1 = (1 + \lambda)(00^*1)^*0^*$ and $R2 = (0 + \lambda)(11^*0)1^*$

Consider the String 00

$00 \in R1$

$00 \notin R2$

So,

$R1 \neq R2$

- 4) Check the Equivalence of two regular expression using informal method

$R1 = 0^*(10^*)^*$ and $R2 = (1^*0)^*1^*$

$R1$ can generate all strings generated by $R2$ and vice versa , So they are equivalent.

Example string :

b, aba, bb, ...