Closure Properties of DFAs

CSE 2233

Regular Language

Language:

A *set of strings* all of which are chosen from some Σ^*, where Σ is a particular alphabet, is called a language.

Regular Language:

Regular languages are the languages recognized by *DFA's*, by *NFA's*, and ε-*NFA's*. They are also the languages defined by *regular expressions*.

Ex.

If alphabet $\Sigma = \{ 0, 1 \}$
then Σ^* = set of all strings over $\Sigma = \{ \varepsilon, 0, 1, 00, 01, 10, 11, 000, \ldots \ldots \} = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \ldots \ldots \ldots$

Language, $L = \{ w \mid w \text{ ends with } 101 \} = \{ 101, 0101, 1101, \ldots \ldots \} \subseteq \Sigma^*$

If we can design a *DFA* or *NFA* or ε-*NFA* that can recognize L or, if we can define a *regular expression* then L will be a *Regular Language*.

Regular Operations

– Arithmetic operations create a new value from 1 or 2 existing values (eg. 2+3)

– Similarly Regular operations create a new language from 1 or 2 existing languages.

We define 3 regular operations (using languages A and B):

- **Union:** $A \cup B = \{ x \mid x \in A \text{ or } x \in B \}$
- **Concatenation:** $A \circ B = AB = \{ xy \mid x \in A \text{ and } x \in B \}$
- **Star:** $A^* = \{ x_1 x_2 x_3 \ldots \ldots x_k \mid k >= 0 \text{ and each } x_i \in A \}$

Here, languages A and B don't have to be regular always.

Regular Operations - Example

Let,

Alphabet $\Sigma = \{ a, b, c, \dots \dots \dots, z \}$

Language $A = \{ good, bad \} \subseteq \Sigma^*$

Language $B = \{ boy, girl \} \subseteq \Sigma^*$

Then,

$A \cup B = \{ good, bad, boy, girl \}$

$AB = \{ goodboy, goodgirl, badboy, badgirl \}$

$A^* = \{ \varepsilon, good, bad, goodgood, goodbad, badgood, badbad, goodgoodgood, goodgoodbad, \dots \dots \}$

Other Operations

Let A and B be languages, then

- **Intersection:**

  $$A \cap B = \{ x \mid x \in A \text{ and } x \in B \} = \overline{\bar{A} \cup \bar{B}}$$

- **Complementation:**

  $$\bar{A} = \{ x \mid x \notin A \}$$

- **Subtraction:**

  $$A - B = \{ x \mid x \in A \text{ and } x \notin B \} = A \cap \bar{B}$$

- **Reverse:**

  $$A^R = \{ w_1 \ldots w_k \mid w_k \ldots w_1 \in A \}$$

  ex. if $A = \{ 001, 10, 111 \}$ then $A^R = \{ 100, 01, 111 \}$

Closure Properties of DFAs

Languages captured by DFA's are closed under

- Union
- Concatenation
- Star
- Intersection
- Complement
- Subtraction
- Reverse

Closure Property:

if language L_1 and L_2 are recognized by a DFA then there exists another DFA that will also recognize the new language L_3 generated by performing any of the above mentioned operation.

Operation 1 - Union

Let, Machine M_1 recognizes A_1, where $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$
and Machine M_2 recognizes A_2, where $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$

To construct M that can recognize $A_1 \cup A_2$, here $M = (Q, \Sigma, \delta, q_0, F)$

- $Q = \{ (r_1, r_2) \mid r_1 \in Q_1 \text{ and } r_2 \in Q_2 \}$
- Σ will remain same
- Transition function: $\delta ((r_1 , r_2) , a) = (\delta_1 (r_1, a), \delta_2 (r_2 , a))$
- $q_0 = (q_1 , q_2)$
- $F = \{ (r_1 , r_2) \mid r_1 \in F_1 \text{ or } r_2 \in F_2 \}$

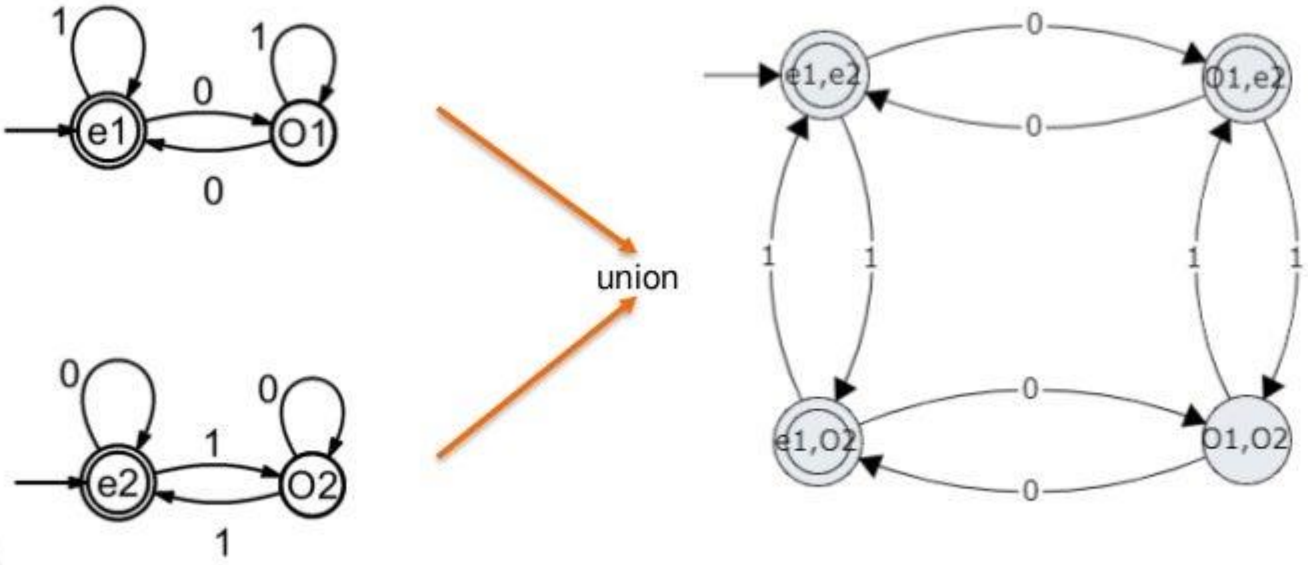
Union Operation – Example 1

Problem:

The set of strings containing even number of 0’s or even number of 1’s over $\Sigma = \{ 0, 1 \}$

Part 1 :
Set of strings containing even number of 0’s and any number of 1’s

Part 2:
Set of strings containing even number of 1’s and any number of 0’s



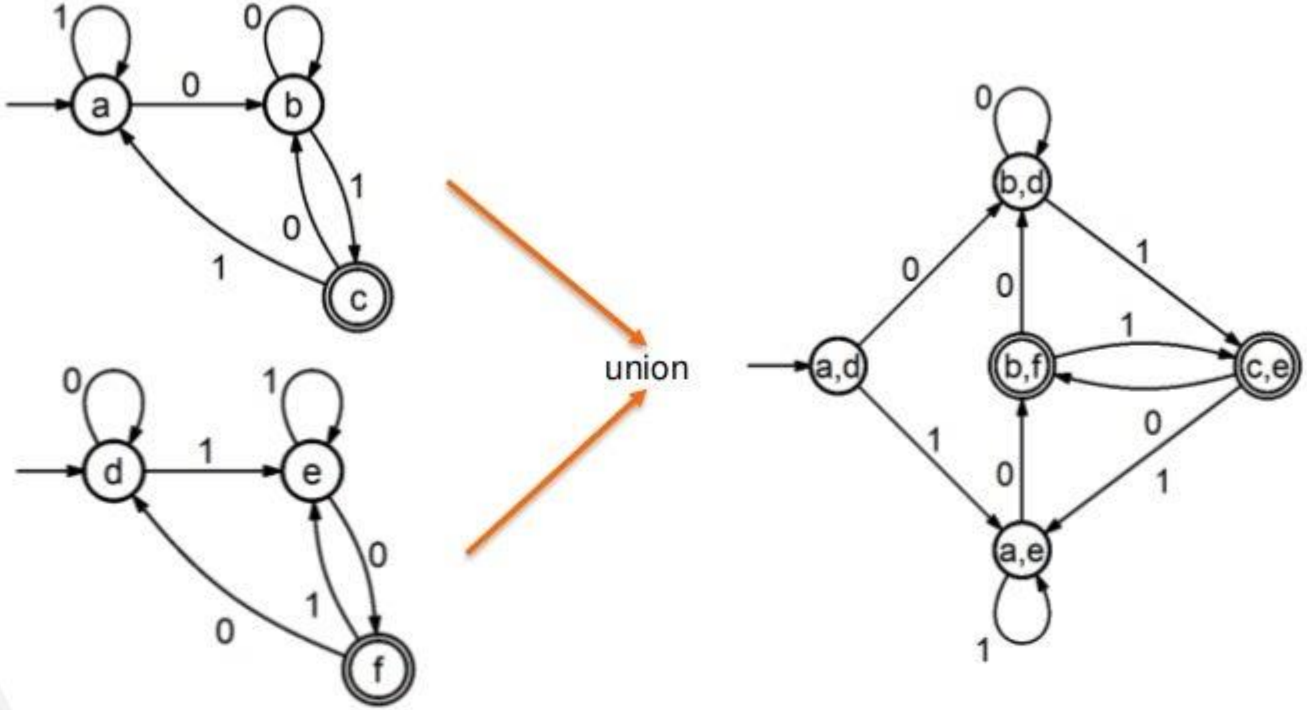
Union Operation – Example 2

Problem:

The set of strings either ending with 01 or 10 over $\Sigma = \{ 0, 1 \}$

Part 1 :
Set of strings ending with 01

Part 2:
Set of strings ending with 10



Union Operation – Practices

1) The set of strings either begin or end with 01
2) The set of strings with 01 or 10 as substring
3) The set of strings starts and ends with the same symbol over $\Sigma = \{ 0 ,1 \}$

Operation 2 - Intersection

Let, M_1 recognizes A_1, where $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$

and M2 recognizes A_2 where $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$

To construct M that can recognize $A_1 \cap A_2$, here $M = (Q, \Sigma, \delta, q_0, F)$

- $Q = \{ (r_1, r_2) \mid r_1 \in Q_1 \text{ and } r_2 \in Q_2 \}$
- Σ will remain same
- Transition function: $\delta ((r_1 , r_2) , a) = (\delta_1 (r_1, a), \delta_2 (r_2 , a))$
- $q_0 = (q_1 , q_2)$
- $F = \{ (r_1 , r_2) \mid r_1 \in F_1 \text{ and } r_2 \in F_2 \}$

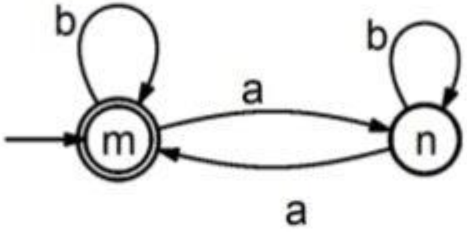
Intersection Operation – Example 1

Problem:

L(M) = { w | w has an even number of a's and each a is followed by at least one b }

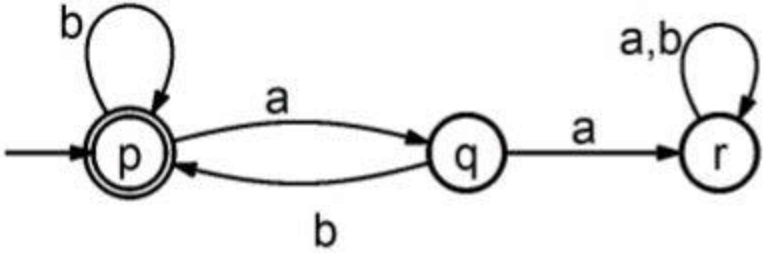
Part 1 :

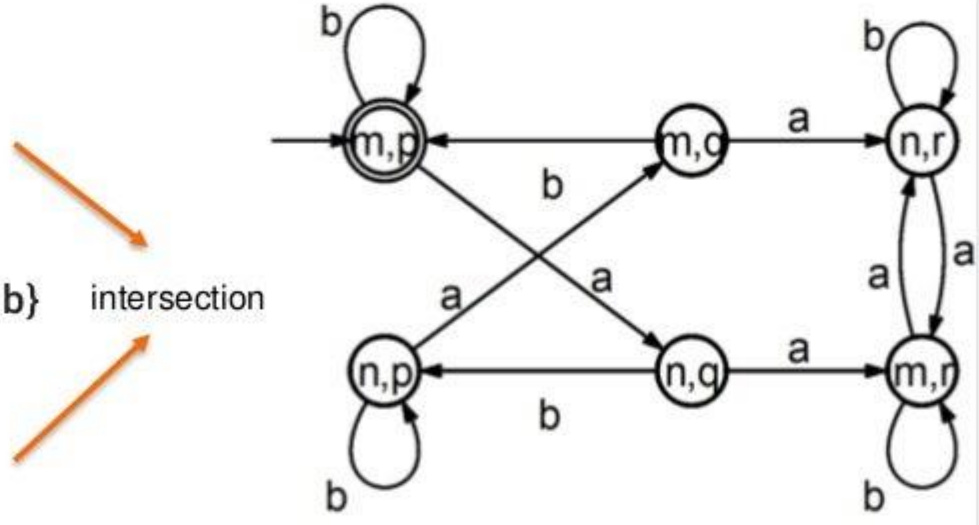
L(M1) = { w | w has an even number of a's}



Part 2:

L(M2) = { w | each a in w is followed by at least one b}





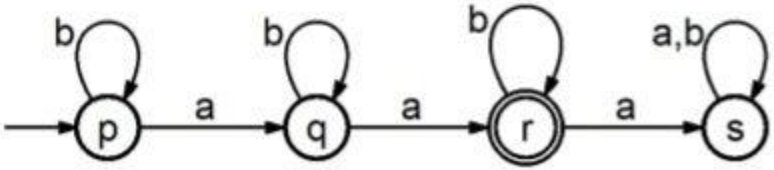
Intersection Operation – Example 2

Problem:

L(M) = { w | w has exactly two a's and at least two b's }

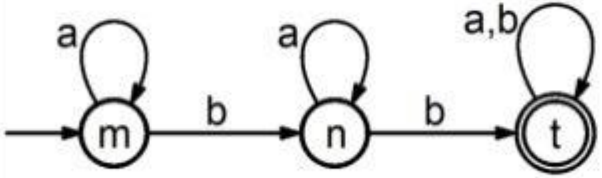
Part 1 :

L(M1) = { w | w has exactly two a's }

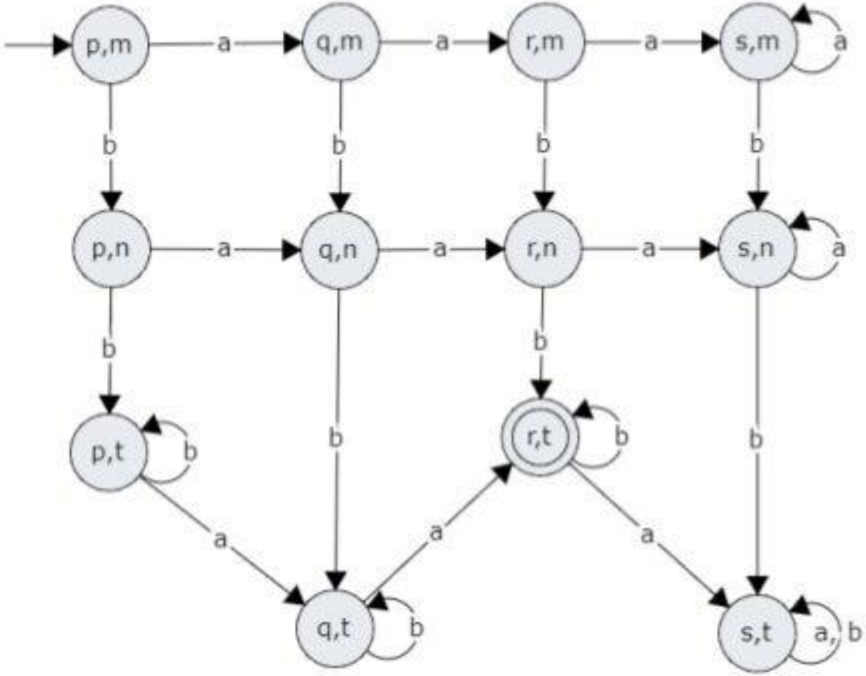


Part 2:

L(M2) = { w | w has at least two b's }



intersection



Intersection Operation – Practices

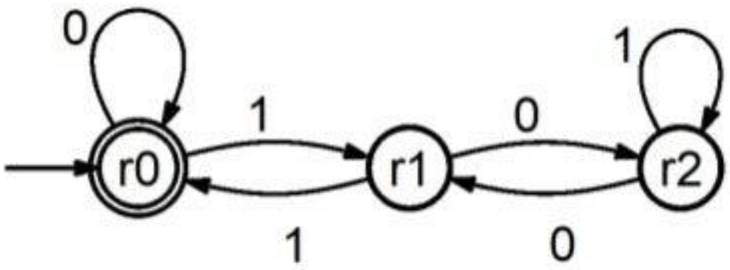
1) { w | w has at least three a's and at least two b's }
2) { w | w has exactly two a's and at least two b's }
3) { w | w has an even number of a's and at most two b's }
4) { w | w has an even number of a's and each a is followed by at least one b }
5) { w | w starts with an a and has at most one b }
6) { w | w has an odd number of a's and ends with a b }
7) { w | w has even length and an odd number of a's }

Operation 3 - Complement

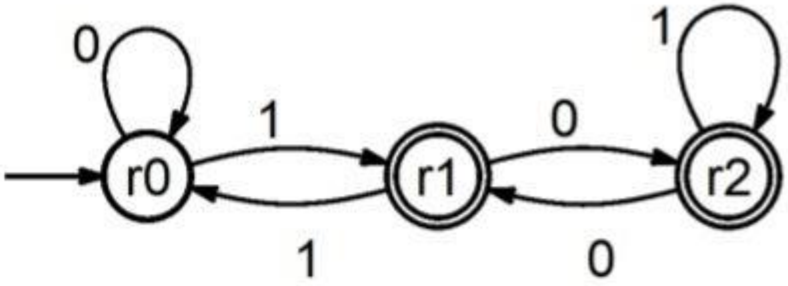
Technique:

Convert all the general states to final states and all the final states to general states.

L = Set of binary strings those are multiple of 3



$\bar{L}$ = Set of binary strings those are not a multiple of 3



Complement Operation – Practices

1) L(M) = { w | w does not contain the substring **ab**}
2) L(M) = { w | w does not contain the substring **baba** }
3) L(M) = { w | w contains neither the substrings **ab** nor **ba** }
4) L(M) = { w | w is any string that doesn't contain exactly two **a's** }