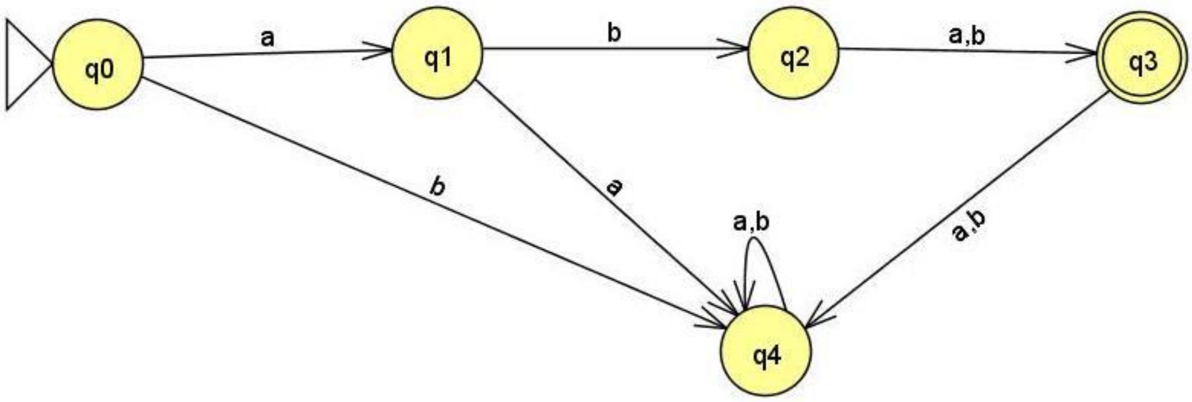
Closure Properties of Regular Languages:

- The union of two regular languages is regular.
- The concatenation of regular languages is regular.
- The closure (star) of a regular language is regular.
- The complement of a regular language is regular.
- The intersection of two regular languages is regular.
- The difference of two regular languages is regular.
- The reversal of a regular language is regular.
- The closure (star) of a regular language is regular.
- A homomorphism (substitution of strings for symbols) of a regular language is regular.
- The inverse homomorphism of a regular language is regular.

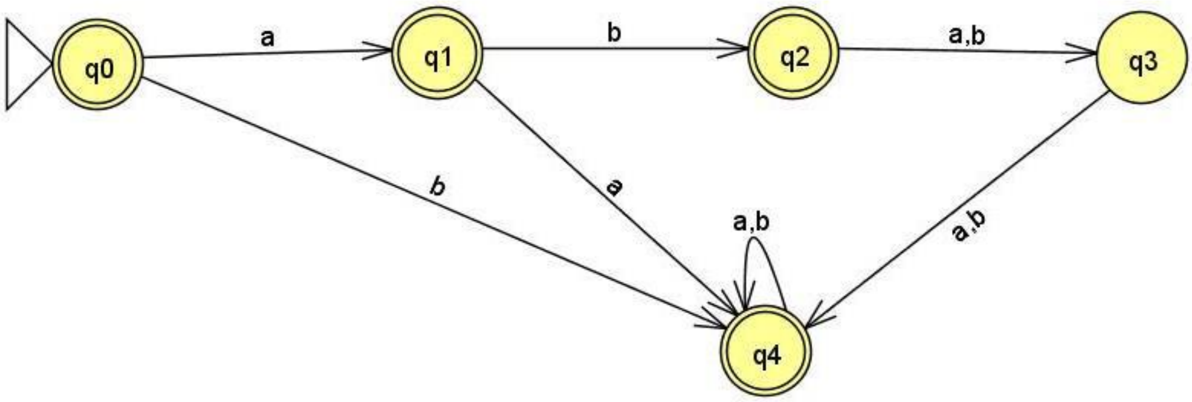
The complement of a regular language is regular:

Let us consider Language L with the following DFA:



What will be its complement L'?

- Change all original final states to non-final.
- Change all original non-final states to final.



Task: What is the RE of this DFA?

The intersection of two regular languages is regular

Consider RE for L_1: $(a + b)^*aa(a + b)^*$
Consider RE for L_2: $b^*(ab^*ab^*)^*$

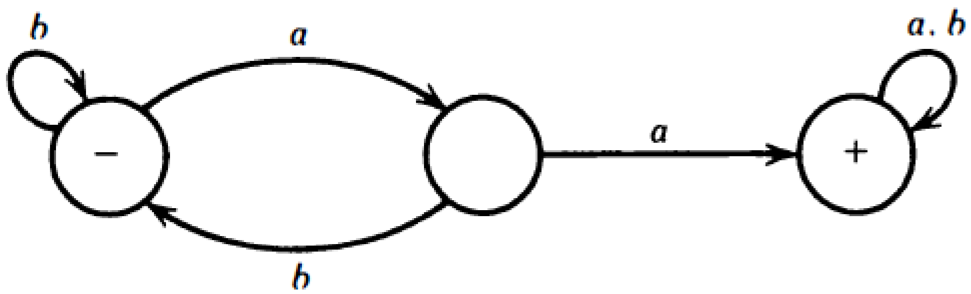
What will be the RE for $L_1 \cap L_2$?

$A \cap B = (A' \cup B')'$

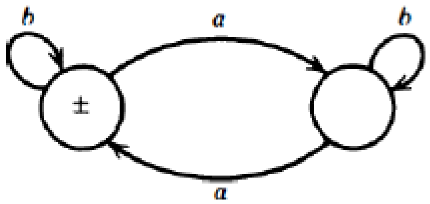
$(A' \cup B')' = (A')' \cap (B')' = A \cap B$

Step 1: A'
Step 2: B'
Step 3: A' U B'
Step 4: (A' U B')'

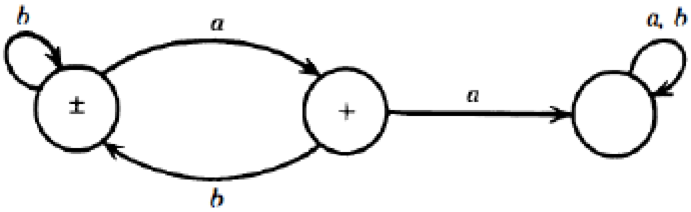
DFA1:



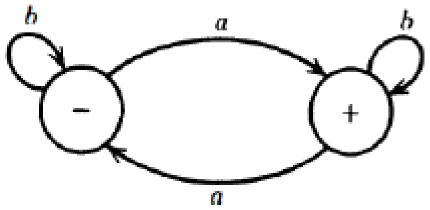
DFA2:



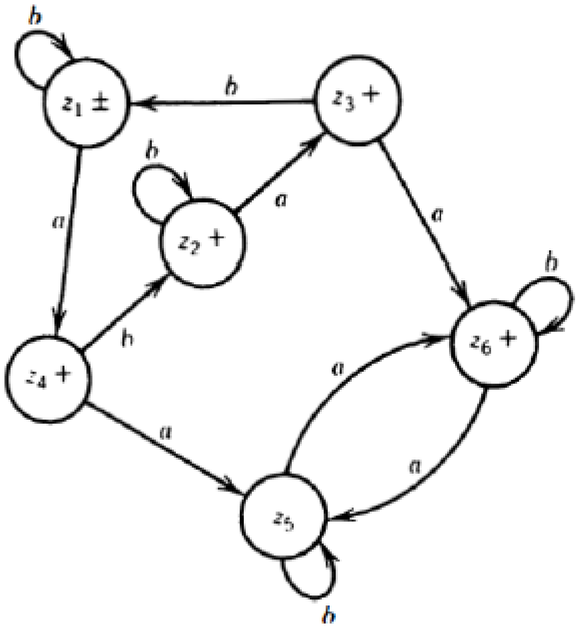
Step 1: DFA1':



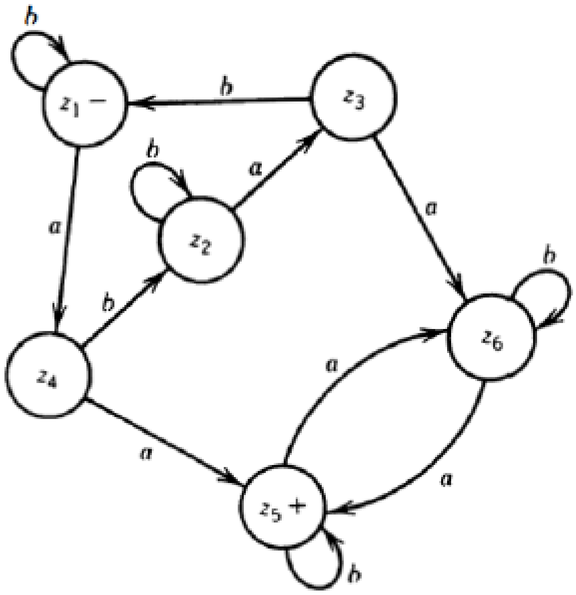
Step 2: DFA2':



Step 3: $DFA_1' \cup DFA_2'$:



Step 4 – $(DFA_1' \cup DFA_2')' = DFA_1 \cap DFA_2$:

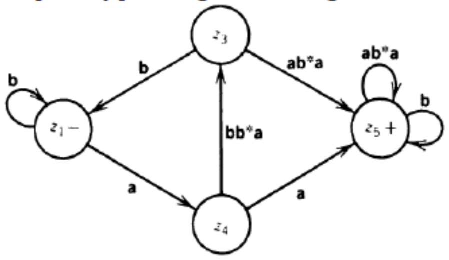


Finding out the RE for L1 ∩ L2

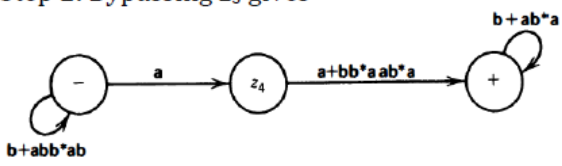
Step 1: Bypassing z2 and z6
Step 2: Bypassing z3
Step 3: Bypassing z4

Finding out the RE for $L_1 \cap L_2$

Step 1: Bypassing z_2 and z_6 gives



Step 2: Bypassing z_3 gives



Step 3: Bypassing z_4 gives: (b+abb*ab)*a(a+bb*aab*a)(b+ab*a)*

The difference of two regular languages is regular:

If L and M are regular languages then L-M is also a regular language.

Proof:

- We know that
  - If M is regular language then M' is also regular
  - If L and M are regular languages then L ∩ M is also regular
- Because L – M = L ∩ M', so L – M is also regular.

Task: Implement this on DFA for the language accepting strings containing aa.

Decidability:

1. Is this language empty?
2. Is the string w in the language?
3. Are the languages equivalent?

Q1. Is this language empty?

RE: (a+λ)(ab*+ba*)(λ+b*)*

Step 1: Remove all *
Step 2: Remove all + and its right side

Applying step 1 on the RE: (a+λ)(ab+ba)(λ+b)

Applying step 2 on the RE: (a)(ab)(λ)
= aabλ
= aab .... The language accepts aab at least !!!

So, The language is not empty.

Note:
- aλ = a
- λa = a
- aλb = ab
- bλa = ba

Q3. Are the languages equivalent?

L1 = {2,3,4,5}

L2 = {2,3,4,5,6}

> L_1 and L_2 are not equivalent because
> - although all members L_1 are in L_2,
> - but all members of L_2 are not in L_1.

L1 ∩ L2 = {} …… Does it mean that they are equivalent?

L1 = {1,2,3}

L2 = {6,7,8} …… L1 ∩ L2 = {} … They are NOT equivalent?

If $(L_1 \cap L_2') + (L_1' \cap L_2) = \{\}$ then it means both languages are equivalent !!!

U = {1,2,3,4,5}

L1 = {1,2,3}	L1' = {4,5}
L2 = {1,2,3}	L2' = {4,5}
L1 ∩ L2' = {}	L1' ∩ L2 = {}

$(L1 \cap L2') \cup (L1' \cap L2) = \{\}$ ……….. Both languages are equivalent