1

a

T

Let L is a finite language, and Let |L| = n, for some $n \geq 0$

For n = 0, $L = \phi$ and it is regular.

For $n > 0$,

lets , $L = \{a_1, a_2, a_3, ...a_n\}$ where $a_k \ for \ k \geq 1$ is the individual string in L.

Now, for each string $a_k \in L$ we generate a new Language, $L_k = a_k$ for $k \geq 1$.

We can draw a DFA for a language that define a single string. therefore all L_k for $k \geq 1$ is regular.

Since, $L = L_1 \cup L_2 \cup ..... \cup L_k$ and set of regular language is closed under Union operation. therefore L is a regular language.

b

F, proof this by showing an example.

For proving false when we have Every in the statement, giving one example contradicting the statement should suffice to proof that the satement is false.

We will disprove this satement with an example,

Let, $L = \{W \in \{0, 1\}^*$| W contains an even number of 0's$\}$

This is a regular language as we have constructed this DFA in Lecture 7, but it is a infinite language.

c

T

If L is regular then there is a Machine M that outputs yes for all string $x \in L$.

Now since L

For input alphabets a and b, a* b* is regular. A DFA can be drawn for a* b* but $a^n b^n$ for n≥0 which is a subset of *ab* is not regular as we cannot define a DFA for it.

Language of all binary string is regular, we can draw a DFA. with one state.

But $L_m = \{W \in \{0, 1\} * |W = 0^k 1^k$,for $k \geq 0\}$ is not regular even thous it is a subset of all binary string.

d

F
Q1:Prove that Regular Sets are NOT closed under infinite union. (A counterexample suffices).

Ans1: Consider the sets {0}, {01}, {0011}, etc. Each one is regular because it only contains one string. But the infinite union is the set {0i1i | i>=0} which we know is not regular. So the infinite union cannot be closed for regular languages.

Q2: What about infinite intersection?

Ans2: We know that

{0i1i | i>=0} = {0} U {01} U {0011} U ...,

Taking complements and applying DeMorgan's law gives us

{0i1i | i>=0}c = {0}c ^ {01}c ^ {0011}c ^ ...,

e

Regular language are closed under union.

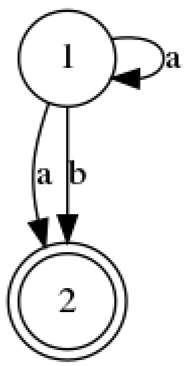
f

NFA complement

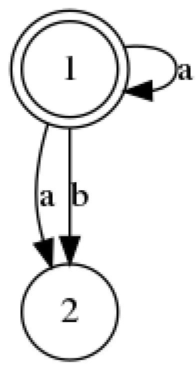
The normal way to take the complement of a regular language L, assuming you have a DFA recognising L is the following:

> Take all accept states and change them into non-accepting states, and vice versa.

But you cannot do the same for NFAs:



M which recognises at least {a, b}.



Complement of M which also recognises at least {a}.

So for correctness you cannot just make non-accepting states accepting and vice versa; you probably need to convert it into an equivalent DFA first and run the complement algorithm on that.

```
#states
A
B
C
#initial
A
#accepting
A
B
#alphabet
0
1
#transitions
A:0>A,B
A:1>A
B:1>C
```

Try 2 strings ending with 01 and see the difference in this DFA
Give counter example.

g

F, proof this by showing an example
L = {00, 1, 10, 11}
A = 00
B = 1
X = 0
Accordign to definition, $\forall$,X if $AX \in L$ if and only if $BX \in L$,
but in our example, $AX \notin L$ but $BX \in L$

2

a

$\mathscr{L}(M) = \{W \in 0, 1^* | W \ has \ no \ consecutive \ 0's\}$

b

Invariants

To prove:
After running the DFA on input W, the following three Invariants are true.
Invariant 1: If the current state is A, then W has no consecutive 0's.
Invariant 2: If the current state is B, then W has no consecutive 0's and last symbol of W is 0.
Invariant 2: If the current state is C, then W has at least one instance of consecutive 0's.

Proof:

We proceed by induction on the length of W.

Base case

For the base case, $W = \epsilon$, notice that we are in $A$and W has no 2 consecutive 0's. hence out invariant 1 is true.

When $W = \epsilon$ we are not in state B and C, so invariant 2 and 3 are vacuously true.

Induction hypothesis

Assume that, after running the DFA on any string Y of length $k \geq 0$, the following statements are true:

1. If the current state A, then Y has no consecutive 0's. and if $|Y| \geq 1$ then, the last symbol of Y is 1.
2. If the current state B, then Y has no consecutive 0's. Last symbol of Y is 0.
3. If the current state C, then Y has at least one consecutive 2 0's.

Inductive step

Let's consider a string W of length $k+1$, such that, $W = Y \cdot z$, where z is the final symbol of W.

Invariant 1

Direct Proof:
Suppose that after reading $W = Y \cdot z$, the current state is A,
Then by induction Hypothesis, Y has no consecutive 0.
And since $z = 1$ being the last symbol of W. therefore $W = Y \cdot z$ also has no consecutive 0's.

Invariant 2

Direct Proof:
Suppose that after reading $W = Y \cdot z$, the current state is B,

Now, z = 0, since from the DFA, the only transition going into state B is by seeing symbol 0.

Invariant 3

Direct Proof: Suppose that after reading $W = Y \cdot z$, the current state is C,

Case 1

z = 0
For given M, we conclude that the machine was in state B or C after reading Y

- Case 1.1 (machine was in state B)
  - By induction hypothesis, Y has no 2 consecutive 0 and the last symbol of the string is 0.
  - Since z = 0, $W = Y \cdot z$ has at least one consecutive 2 0's.
- Case 1.2 (machine was in state C)
  - By induction hypothesis, Y has at least one consecutive 2 0's.
  - therefore W also has at least one consecutive 2 0's.

Case 2

z = 1
For given M, we conclude that the machine was in state C after reading Y
By induction hypothesis Y has at least one consecutive 2 0's.
therfore W has at least one consecutive 2 0's.

Hence by induction principle all the invariants are proved.

- From the machine description we see that the accepting states are $A,\ B$
- We proved invariant 1: if the machine is in state A, then the string has no consecutive 2 0's.
- We proved invariant 2: if the machine is in state B, then the string has no consecutive 2 0's.
- We proved invariant 3: if the machine is in state C, then the string has at least 1 consecutive 2 0's.
- In other words M accepts $W \implies W \in L$, and, M rejects $W \implies W \notin L$, as rquired.

http://ivanzuzak.info/noam/webapps/fsm2regex/

3

a

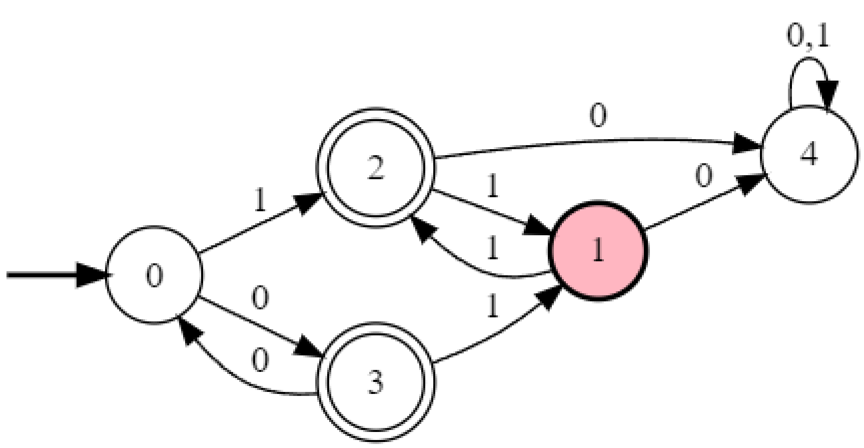
0(0+1)*1+1(0+1)*0

0(0+1)*1+1(0+1)*0

b

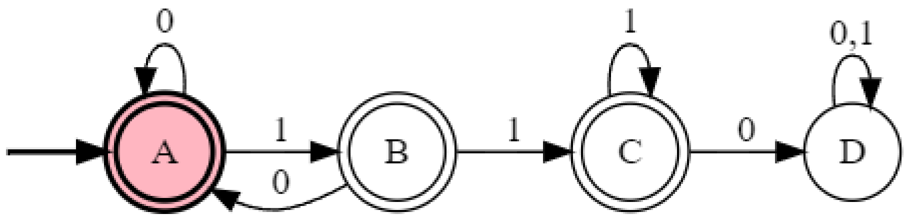
((00)*0)(11)*+(00)*((11)*1)

(00)*(0(11)*+(11)*1)



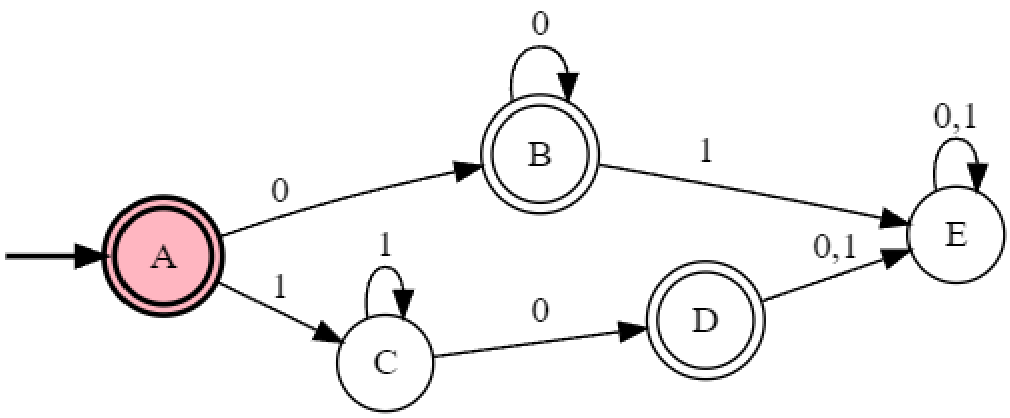
c

(0+10)*11(1)*0*(1+0)*



4

a



Equivalence class:

1. $1 \cdot 1^*$
2. ϵ
3. $0 \cdot 0^*$
4. $1 \cdot 1^*0$

Class	Regular Expression	Set builder notation
C_1	$1 \cdot 1^*$	$\{W \in \{0,1\}^* \| W = 1^k \; where \; k \geq 1\}$
C_2	ϵ	$\{W \in \{0,1\}^* \| W = \epsilon\}$
C_3	$0 \cdot 0^*$	$\{W \in \{0,1\}^* \| W = 0^k \; where \; k \geq 1\}$
C_4	$1 \cdot 1^*0$	$\{W \in \{0,1\}^* \| W = 1^k0 \; where \; k \geq 1\}$

Class	Regular Expression	Set builder notation
C_5	$1 \cdot 1^* \cdot 0(0+1) + 0 \cdot 0^* \cdot 1$	$\{W \in \{0,1\}^* \| W = 1^k 0 \; where \; k \geq 1\}$

b

C1

Let A and B be 2 arbritary string in C_1,
Consider an arbritary string X,
case 1 X = $1^k 0$ where $k \geq 0$
then, AX and BX in L
case 2 $X \neq 1^k 0$ where $k \geq 0$
then, AX, BX not in L

C2

Let A and B be 2 arbritary string in C_2,
Consider an arbritary string X,
Since C_2 has only one element therefore it is vacuaously true

C3

Let A and B be 2 arbritary string in C_3,
Consider an arbritary string X,
case 1 $X \in \{W \in \{0,1\}^* | W = 0^n \; for \; n \geq 0\}$
the both AX and BX are of the form 0^m where $m \geq 0$ and thus $AX, BX \in L$
❓ Is this a complement? Is this complement valid?
case 2 $X \notin \{W \in \{0,1\}^* | W = 0^n \; for \; n \geq 0\}$

> case 2 $X \notin \{W \in \{0,1\}^* | W \; has \; 1 \; as substring\}$

the both AX and BX are not in L, as for a string starting with 0 can only contain 0, but X has at least 1 and thus both AX and BX at least one or more 1.

C4

Let A and B be 2 arbritary string in C_4,
Consider an arbritary string X,
case 1 X = ϵ
the both AX and $BX \in L$

case 2 $X \neq \epsilon$
the both AX and $BX \notin L$

C5

Let A and B be 2 arbritary string in C_5,
Consider an arbritary string X,
for all X AX and $BX \notin L$

To prove C_1, C_2

We chose 11 from C_1 and ϵ from C_2,
for X = ϵ
$11 \cdot X \notin L$
$\epsilon \cdot X \in L$
therfore, C_1, C_2 forms seperate equivalence class

To prove C_1, C_3

We chose 11 from C_1 and 00 from C_3,
for X = 00
$11 \cdot X \notin L$
$00 \cdot X \in L$

therfore, C_1, C_3 forms seperate equivalence class

To prove C_1, C_4

We chose 11 from C_1 and 110 from C_4,
for X = 0
$11 \cdot X \in L$
$110 \cdot X \notin L$
therfore, C_1, C_2 forms seperate equivalence class

To prove C_1, C_5

We chose 11 from C_1 and 101 from C_5,
for X = 0
$11 \cdot X \in L$
$101 \cdot X \notin L$

therfore, C_1, C_5 forms seperate equivalence class

To prove C_2, C_3

We chose ϵ from C_2 and 0 from C_3,
for X = 10
$\epsilon \cdot X \in L$
$0 \cdot X \in L$
therfore, C_2, C_3 forms seperate equivalence class

To prove C_2, C_4

We chose ϵ from C_2 and 10 from C_4,
for X = 0
$\epsilon \cdot X \in L$
$10 \cdot X \notin L$
therfore, C_2, C_4 forms seperate equivalence class

To prove C_2, C_5

We chose ϵ from C_2 and 101 from C_4,
for X = 0
$\epsilon \cdot X \in L$
$101 \cdot X \notin L$
therfore, C_2, C_5 forms seperate equivalence class

To prove C_3, C_4

We chose 00 from C_3 and 10 from C_4,
for X = 0
$00 \cdot X \in L$
$10 \cdot X \notin L$
therfore, C_3, C_4 forms seperate equivalence class

To prove C_3, C_5

We chose 11 from C_3 and 101 from C_5,
for X = 0
$00 \cdot X \in L$
$101 \cdot X \notin L$
therfore, C_3, C_5 forms seperate equivalence class

To prove C_4, C_5

We chose 10 from C_4 and 101 from C_5,
for X = ϵ
$10 \cdot X \in L$
$101 \cdot X \notin L$
therfore, C_4, C_5 forms seperate equivalence class