LING 570 HW3

Dhwani Serai (dserai)

20th October 2022

Answer 1

According to the definition of regular languages
$\forall\, a\; \epsilon \sum \cup [\varepsilon], [a]$ is a regular language
Here $\sum$ is a finite set of symbols known as an alphabet
The language in question is a finite set of strings where each individual $string_i$ can be represented as a Language L_i over an alphabet $\sum$

- Here every L_i is a regular language as defined from the original definition
- Let L be a set of union of all individual L_i where each L_i is regular (by definition)
- $L \equiv L_1 \cup L_2 \cup ... \cup L_n$
- We know that a regular language is closed under union
- Hence L is a Regular language because it is defined over a finite set of strings

Answer 2

If Language L is a regular language and there is an FSA which represents Language L:

- Then for $\bar{L}$ i.e. L complement;
- We turn every final-state in the FSA to a non-final state and every non-final state to final state
- If the FSA has only one state then we turn that to a non-final state and add another state which will be a final state corresponding to every other symbol in $\sum$ not mentioned earlier in L i.e. the production will be S1=¿S2 and S2=¿S2 for $\sum$ - initial production symbols
- So we can say that for every regular L $\bar{L}$ is also regular [since it can be represented by a FSA]
- in the case of a* we will add another state in the FSA to represent the final-state where State S1=¿S2 for b and B=¿B for b

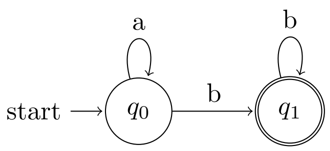
Answer 3

To prove that language is closed under intersection; given that it is closed under union and complement
By set theory using De Morgan's law we know that for two sets A and B:

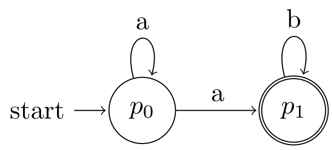
- $\overline{\overline{A} \cup \overline{B}} \equiv A \cap B$
  if A and B are regular the, $\overline{A}\, and\, \overline{B}$ are also regular
  Then $\overline{A} \cup \overline{B}$ and $\overline{\overline{A} \cup \overline{B}}$ are regular. Reason: Regular languages are closed under union and complement
- So, if LHS is regular then RHS is also regular because they are equivalent
  Therefore, for two regular languages L1 and L2 we can say that they are also closed under intersection

Answer 4

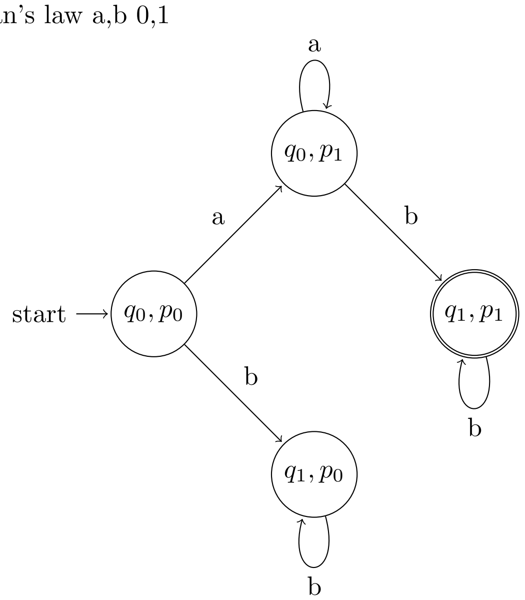
FSA1 a^*b^+



FSA2 a^+b^*



Using DeMorgan's law a,b 0,1



$FSA1 \cap FSA2$

Answer 5

Done with bonus content

Answer 6

If L is regular, then **there exists** a $p \geq 1$, such that **for all** $w \in L$ with $|w| \geq p$, **there exists** a "splitting" $w = xyz$, such that **for all** $i \geq 0$ xy^$(i)z \in L$

a) L1 = wv | w and v are strings of the form (a | b)*, and v is the reverse of w
According to the definition above let us split L1 into x,y,z
using the example string "aabbaa"

Case1: x=ϵ y=aab z=baa
initial conditions are satisfied i.e. $|y| \geq 1$ and $|xz| \geq n$
if we pump y i.e. increase the value of i in y^i we get "aabaabbaa" but "aabaabbaa" is not accepted by language L_1
Similarly for other splits(cases) we can prove that after pumping the resulting string will not be accepted by the language
Hence we can prove that language L_1 is not regular

b)L2 = waaaw | w is a string of the form (a | b)*
using example string "aabaaaaab"
Case 1: Let us assume x="aab" y="aaa" z="aab"
in this also the initial conditions are satisfied. On pumping y i.e. y=0 the string will be "aabaab" but this is not accepted by L2 because in L2 there should be 3a's in the middle
So, language L2 is not regular

Answer 7

Readme for the code:
handled two types of productions:

- A t B
- A t

For final state I have defined my own final state i.e. `_F_` In the given input if there is any terminal production i.e. A t or A=¿t I have considered it as a final state
and appended a new production going from A *e* goes to `_F_` which is my program's final state