Formleg mál og reiknanleiki

Pétur

October 1, 2018

1.

a)

$A = \{0^{n+m}1^m | n \geq 0, m \geq 0\}$

- Assume A is regular only if pumping lemma holds for some p.
- Let $s = 0^p1^p$, so that $|s| = 2p > p$. According to pumping lemma, s can be split into 3 pieces, $s = xyz$, such that $xy^iz \epsilon A$ for all $i \geq 0$.
- 3 cases
  1. y consist only of zeros: xy^2z has more zeros than ones and that is not in the language A.
  2. y consist solely on ones: xy^2z has more ones than zeros and that is not in the language A.
  3. y consist of "10": xy^2z has then "1010" which is not in the language A.
- as all three cases lead to contradiction, the initial assumption, A being regular must be false.

b)

$B = \{0^n1^m0^n | n \geq 0, m \geq 0\}$

- Assume B is regular only if pumping lemma holds for some p.
- Let $s = 0^p10^p$, so that $|s| = 2p > p$. According to pumping lemma, s can be split into 3 pieces, $s = xyz$, such that $xy^iz \epsilon B$ for all $i \geq 0$.
- 3 cases
  1. y consist only of zeros left hand side of the one: xy^2z has more zeros left hand side than right, which is not in the language B.
  2. y consist only of zeros right hand side of the one: xy^2z has more zeros right hand side than left, which is not in the language B.
  3. y consist of "010": xy^2z the string contains then "010010" which is not in the language B.
- as all three cases lead to contradiction, the initial assumption, B being regular must be false.

c)

$C = \{www | w \epsilon \{0,1\}*\}$

- Assume C is regular only if pumping lemma holds for some p.
- Let $s = 0^p10^p10^p1$, so that $|s| = 3p > p$. According to pumping lemma, s can be split into 3 pieces, $s = xyz$, such that $xy^iz\epsilon C$ for all $i \geq 0$.
- 3 cases
  1. y consist only of zeros left hand side: xy^2z has more zeros left hand side than right and middle, which is not in the language C.
  2. y consist only of zeros right hand side : xy^2z has more zeros right hand side than left and middle, which is not in the language C.
  3. y consist of zero in the middle: xy^2z the string contains then more zeros in the middle than right and left hand site, which means that it is not in the language C.
- as all three cases lead to contradiction, the initial assumption, C being regular must be false.

2.

$A = \{x = y + z | x, y, z$ are binary integers and x is the sum of y and z $\}$

- Assume A is regular only if pumping lemma holds for some p
- x,y,z are not pump able. If they are chosen as y^2 that means it will change the outcome of the calculation.
- as all three cases lead to contradiction, the initial assumption, A being regular must be false.

3.

a)

Let $F = \{a^ib^jc^k | i, j, k \geq 0$, and if $i = 1$ then $j = k\}$

- Assume F is regular only if pumping lemma holds for some p
- let $s = a^pb^pc^p$, so that $|s| = 3p > p$. According to pumping lemma s = xyz, such that xy^iz ϵ C for all $i \geq 0$.
- 3 cases
  1. y consist of "ab": xy^2z which generates "abab" which is not in the language F.
  2. y consist of "abc": xy^2z which generates "abcabc" which is not in the language F.
  3. y consist of "bc": xy^2z which generates "bcbc" which is not in the language F.
- as all three cases lead to contradiction, the initial assumption, A being regular must be false.

b)

- 3 cases
    1. y consist of "a": xy^2z which generates "aa" which is in the language F.
    2. y consist of "b": xy^2z which generates "bb" which is in the language F.
    3. y consist of "c": xy^2z which generates "cc" which is in the language F.
- as all three cases lead to no contradiction, the initial assumption, F being regular must be true.

c)

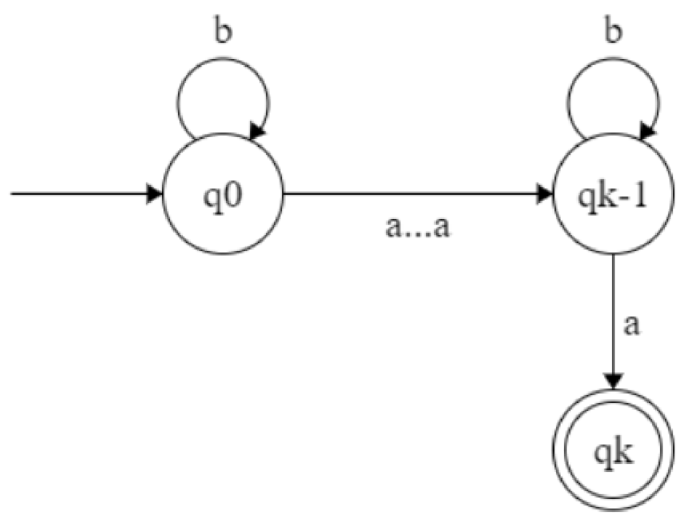
They do not contradict the pumping lemma because the pumping lemma says that if you can pump any y and it is in the language then the language must be regular. So it depends on what y you chose. One can show irregularity and other can show regularity but both are able to pump. While you are proving that language is not regular than you chose the y that you are pumping to show the irregularity.

4.

a)

The NFA that will recognize will start with

- k number of a "a" or "b" for each k.
- when k-1 has bin reach then it ends with a.
- That means that you need infinite amount of states to manage all the "a" transition. So that the state machine knows when it should quit and end with k-1 "a".



b)