Alexander Garcia
3 May 2017

1. Question 1

   (a) $S \rightarrow 00S1$
   $S \rightarrow 1S00$
   The first two are clear rules of the language. When adding a single 1, you must also add two 0's

   $S \rightarrow SS$
   Concatenating two existing strings in the grammar will also produce a vaild string. Both have the correct relationship between 1's and 0's, so the relationship does not change when they are combined

   $S \rightarrow 0S1S0$
   The forth rule allows the concatenation of two existing strings, and adding two 0's and a 1, allowing for "random" placement of 1's and 0's.

   $S \rightarrow \lambda$
   The fifth rule allows for the termination of a derivation

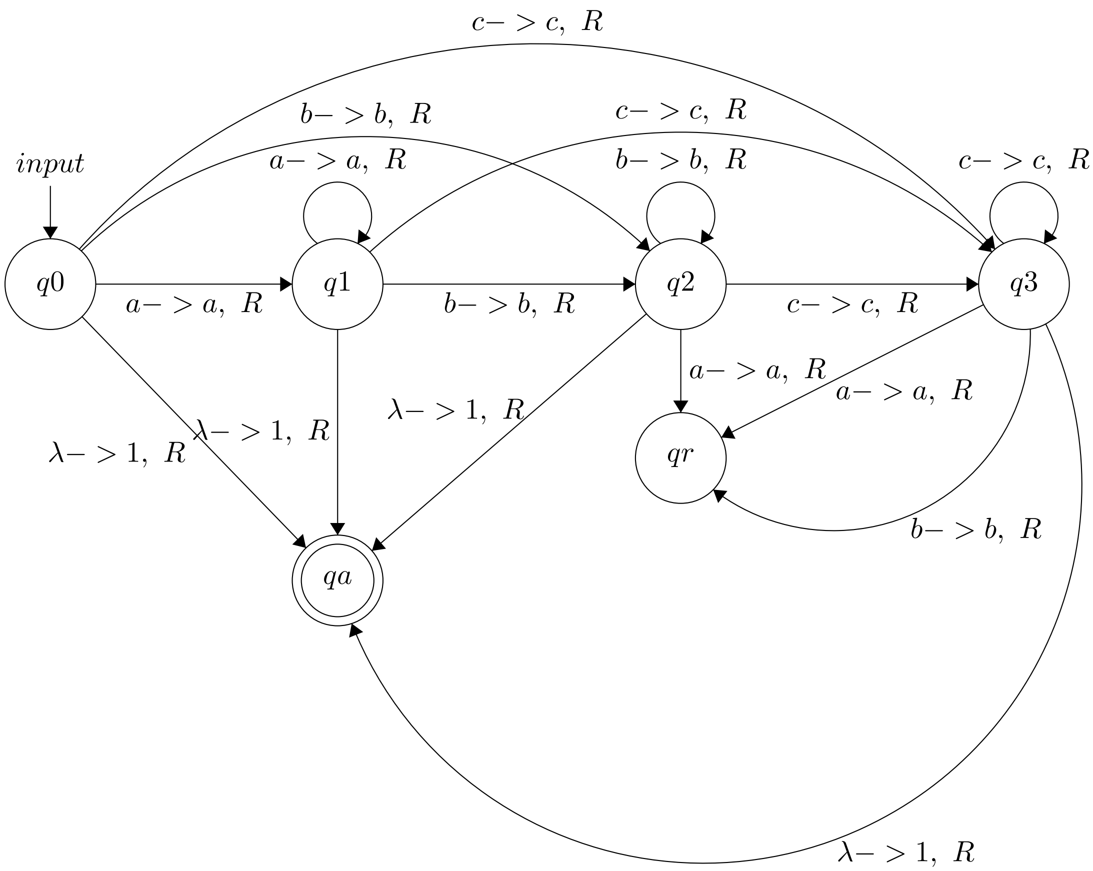
   (b)

$S \rightarrow 00S1$	Rule 1
$00S1 \rightarrow 00[1S00]1$	Rule 2
$00[1S00]1 \rightarrow 00[1[0S1S0]00]1$	Rule 4
$00[1[0S1S0]00]1 \rightarrow 001010001$	Rule 5 for both S

Alexander Garcia

3 May 2017

2. Question 2



The diagram of this machine is quite complicated so this is a summary of its functionality.

The string starts accepting input. Any character in the alphabet, including λ, is sent to a specific state. If a letter that comes after the first letter is seen, the machine immediately goes to the reject state. If the same letter is seen, it stays in the previous state. If a "higher" letter is seen, it goes to that letter's state.

As long as the sequence is lexographic, the machine will stay in one of the three letters' states during the whole input, not modifying the input in any way. When a blank space is seen, if the machine is in one of the three states, it will write a 1, and enter the accept state.

Alexander Garcia

3 May 2017

3. Question 3