

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 valid 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

$$S \rightarrow 00S1$$

Rule 1

$$00S1 \rightarrow 00[1S00]1$$

Rule 2

(b)

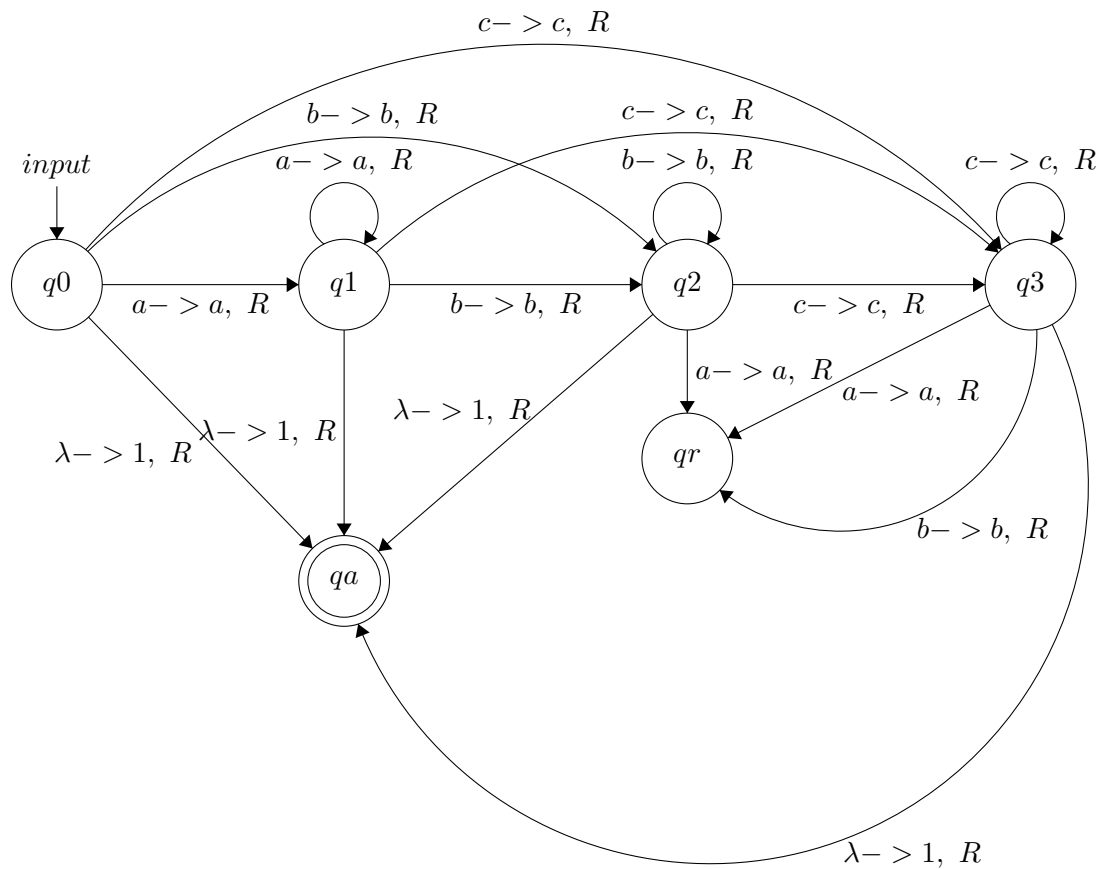
$$00[1S00]1 \rightarrow 00[1[0S1S0]00]1$$

Rule 4

$$00[1[0S1S0]00]1 \rightarrow 001010001$$

Rule 5 for both  $S$

## 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  $\lambda$ , 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 lexicographic, 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.

3 May 2017

3. Question 3

- (a) The main idea behind this machine is to erase pairs of 1's and 0's.

Start cycling through the tape. If you see a 1, remove it, then continue cycling until you reach a 0 or the end. If you reach a 0, remove it as well, and start searching for another 1, repeating the process.

If you reach the end of the input string, start over at the beginning and repeat the process of removing one 1, skipping to the next 0, then removing that 0. If you are able to make it to the end of the tape without removing any digits, you have reached a conclusion. If there are only 0's left, the machine enters the accept state, as there were more 0's than 1's. If there are no digits left, there were exactly as many 0's as there were 1's, and the machine enters the reject state, as it only recognizes words with more 0's than 1's. If there are 1's remaining, it enters the reject state also.

- (b) The runtime of this machine is as follows.

$$r = n + m$$

m = number of 1's, n = number of 0's, r = length of the string

The worst case runtime would be if there are equal numbers of 1's and 0's, or  $m = n = \frac{r}{2}$ . Because of the way this machine operates, the worst case order of digits would be  $\frac{r}{2}$  1's followed by  $\frac{r}{2}$  0's, as the machine would then only remove 2 items per pass.

So, the machine would read  $r$  digits the first pass,  $r - 2$  digits the second pass,  $r - 4$  the third, and so on.

This summation is  $\sum_{i=1}^{r/2} r - 2j = \frac{1}{4}(r - 2) * r$

In terms of order notation, this makes the algorithm  $O(r^2)$

**Alexander Garcia**

3 May 2017

4. Question 4