Formal Languages and Automata Theory

Assignment 4 [Version 2]

Du Nguyen

Q1.

$$\frac{3}{5} \leq \frac{\#0(w)}{\#1(w)} \leq \frac{5}{8}$$

This condition will become the following after multiplying both sides for 40 and #1(w).

The signs of the inequalities do not change because the number of occurrences of symbol 1 in string w (#1(w)) is a non-negative number, according to the given problem.

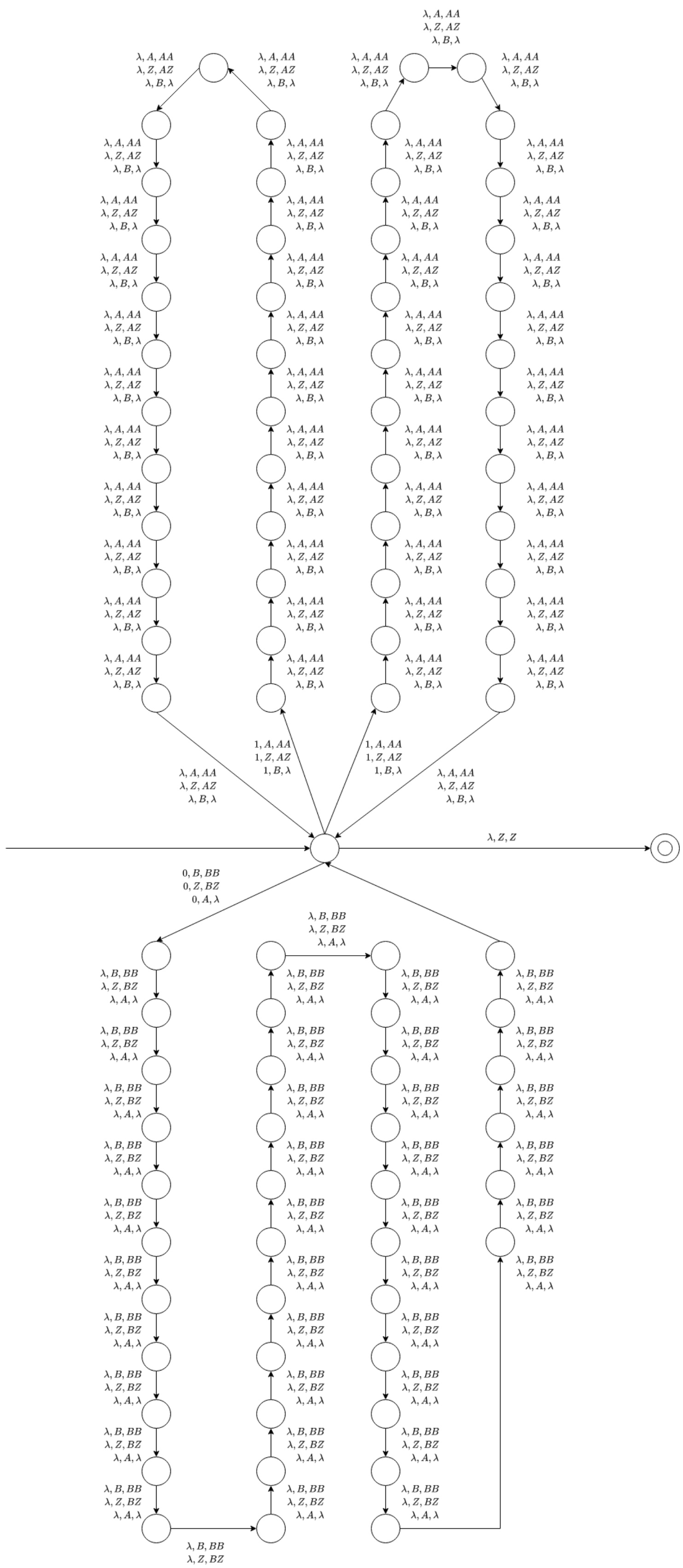
$$24 \leq 40 \times \frac{\#0(w)}{\#1(w)} \leq 25$$

$$24 \times \#1(w) \leq 40 \times \#0(w) \leq 25 \times \#1(w)$$

$40 \times \#0(w) - 24 \times \#1(w) \geq 0$ and $40 \times \#0(w) - 25 \times \#1(w) \leq 0$

The condition suggest we need to keep track of both of the inequalities simultaneously. The former inequality must be non-negative, and the latter inequality must be non-positive. This is done using non-deterministcially by counting down by 24 or 25 for each occurrence of 0, and counting up 40 for each occurrence of 0.

Therefore, the following PDA will satisfy the condition.


λ, A, AA
λ, Z, AZ
λ, B, λ

λ, A, AA
λ, Z, AZ
λ, B, λ

λ, A, AA
λ, Z, AZ
λ, B, λ

λ, A, AA
λ, Z, AZ
λ, B, λ

λ, A, AA
λ, Z, AZ
λ, B, λ

λ, A, AA
λ, Z, AZ
λ, B, λ

λ, A, AA
λ, Z, AZ
λ, B, λ

λ, A, AA
λ, Z, AZ
λ, B, λ

λ, A, AA
λ, Z, AZ
λ, B, λ

λ, A, AA
λ, Z, AZ
λ, B, λ

λ, A, AA
λ, Z, AZ
λ, B, λ

λ, A, AA
λ, Z, AZ
λ, B, λ

λ, A, AA
λ, Z, AZ
λ, B, λ

λ, A, AA
λ, Z, AZ
λ, B, λ

λ, A, AA
λ, Z, AZ
λ, B, λ

1, A, AA
1, Z, AZ
1, B, λ

1, A, AA
1, Z, AZ
1, B, λ

λ, A, AA
λ, Z, AZ
λ, B, λ

λ, Z, Z

0, B, BB
0, Z, BZ
0, A, λ

λ, B, BB
λ, Z, BZ
λ, A, λ

λ, B, BB
λ, Z, BZ
λ, A, λ

λ, B, BB
λ, Z, BZ
λ, A, λ

λ, B, BB
λ, Z, BZ
λ, A, λ

λ, B, BB
λ, Z, BZ
λ, A, λ

λ, B, BB
λ, Z, BZ
λ, A, λ

λ, B, BB
λ, Z, BZ
λ, A, λ

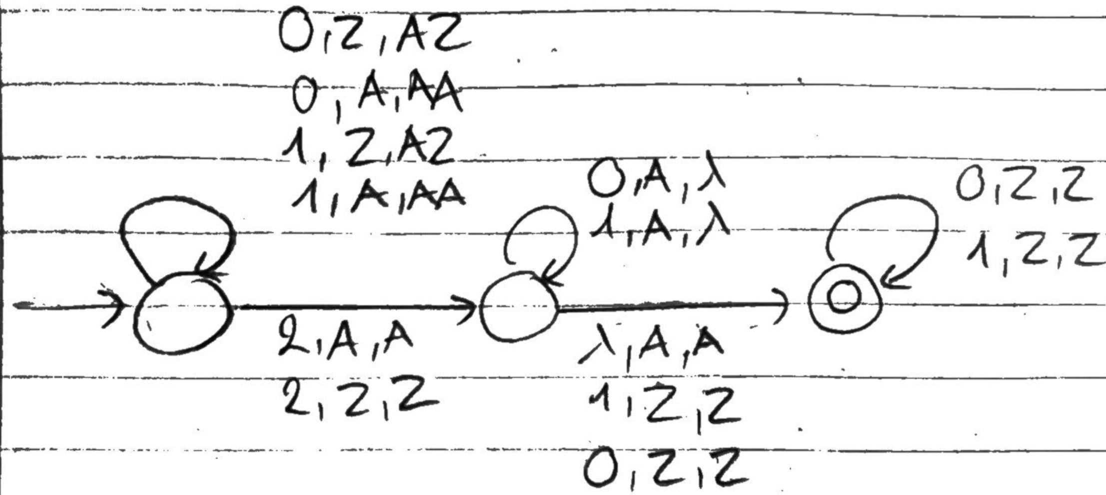
λ, B, BB
λ, Z, BZ
λ, A, λ

λ, B, BB
λ, Z, BZ
λ, A, λ

λ, B, BB
λ, Z, BZ
λ, A, λ

λ, B, BB
λ, Z, BZ
λ, A, λ

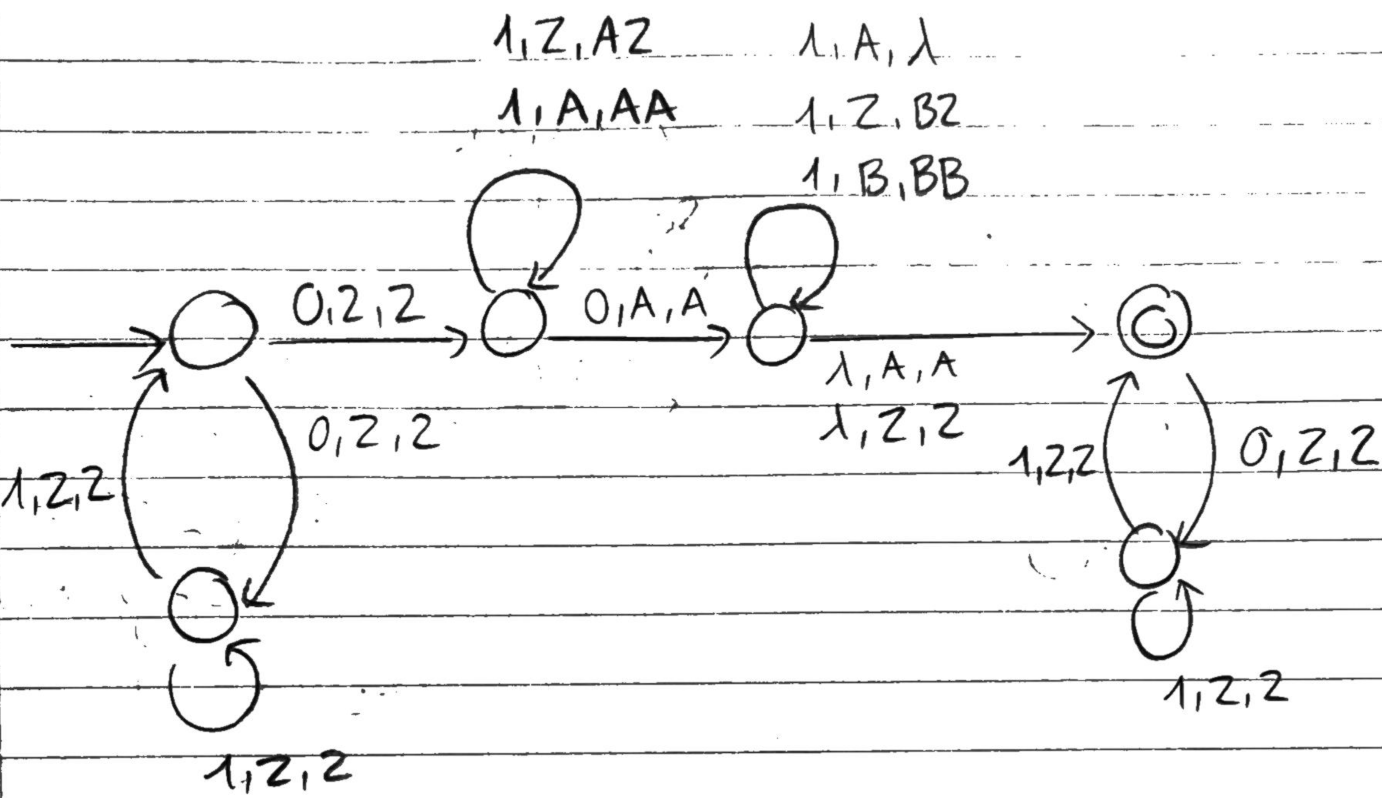
2. $L = \{w_1 . 2 . w_2 : w_1, w_2 \in \{0,1\}^* \text{ and } |w_1| \neq |w_2|\}$



In this PDA, we keep count of length of w_1 by adding A on stack.

Then check for an entry of 2 and transition to w_2. Now, subtract the stack for every entry of w_2.
If there are more entries than the length of w_1, accept

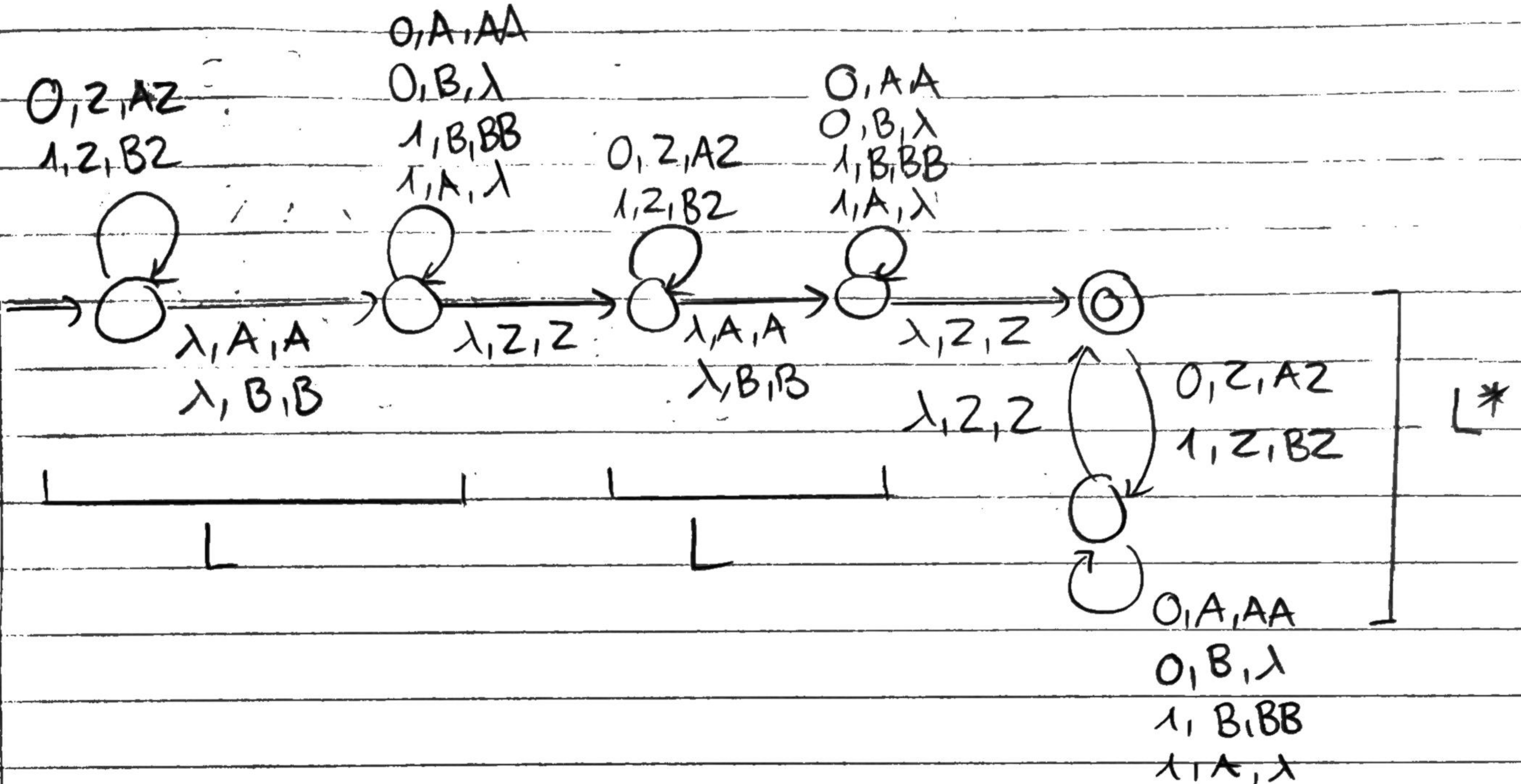
3. $L = \{ w \in (\{0\} \circ \{1\}^+)^* :$ the entries of the list represented by w are not in ascending order $\}$



I will skip a few entries then check where the entry after an entry is less than or equal to the entry right before it.
The stack will be populated by the entry, next, it will be removed, if the stack is emptied, or if there are symbols left on the stack, we will accept the string.

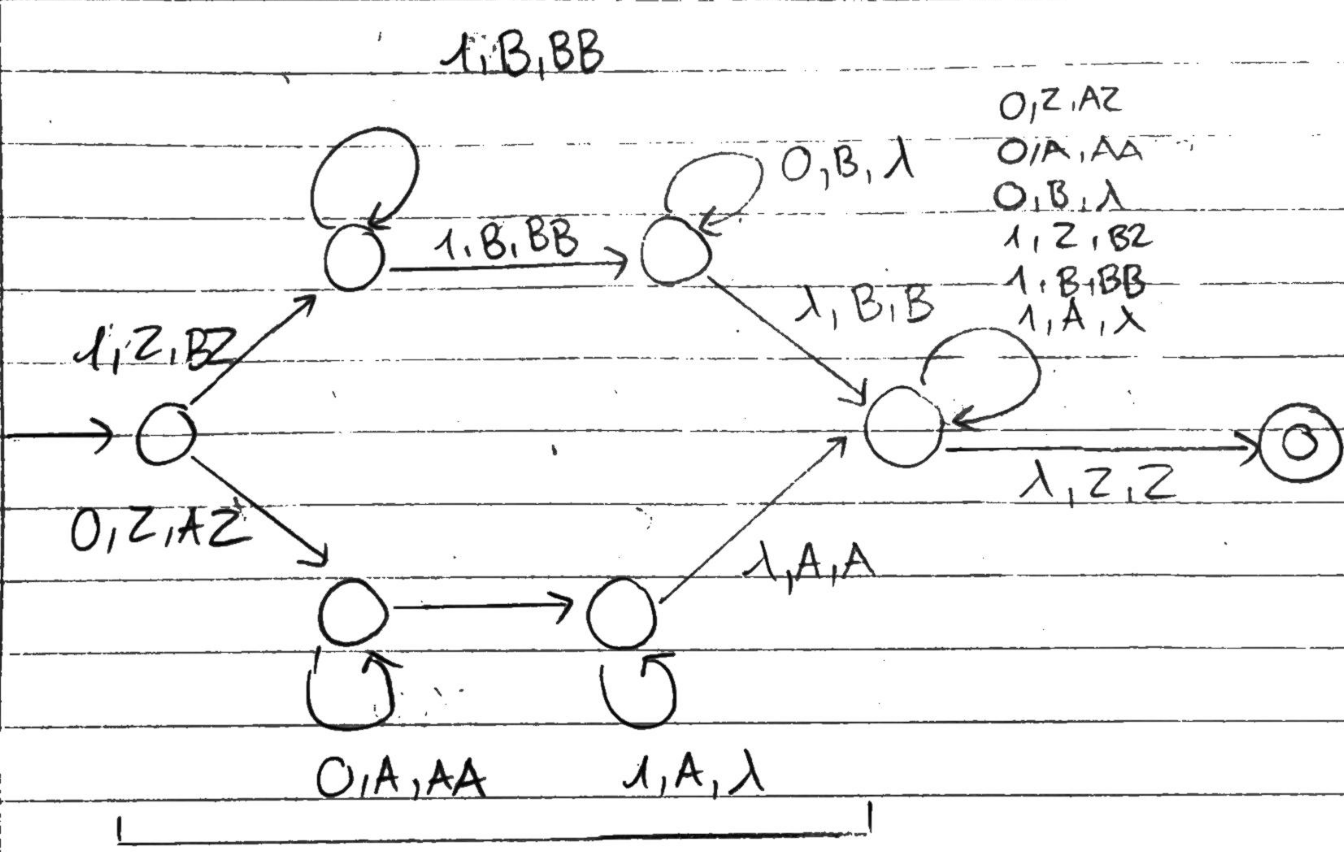
Q4. Let L denote the set of all non-empty binary strings that have equal number of zeroes and ones

PDA for $L \circ L \circ L^*$



Q5.

$L = \{ w \in \{0,1\}^* : \#_0(w) = \#_1(w)$ no non-empty prefix s of w satisfies $\#_0(s) = \#_1(s) \}$

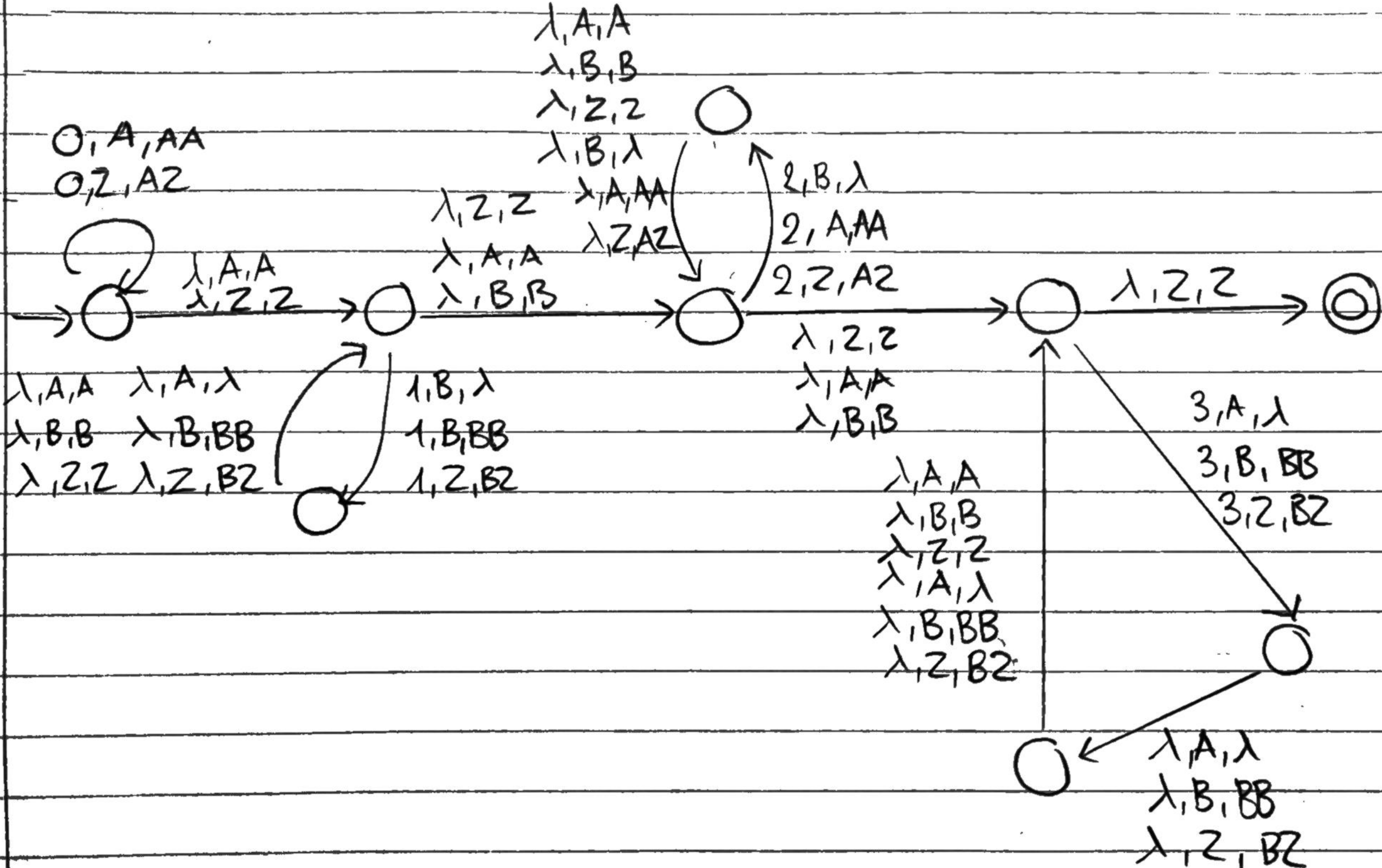


I will keep the length count of all non-empty proper prefix.
Starting with either 0 or 1, if the prefix is 01 or 10, the string must be reject.
The first part of the machine is to ensure that prefix cannot be in the form 01, 10, or have equal number of zeros and ones.
The second part is to make w satisfies. $\#_0(w) = \#_1(w)$

Q6. $L_1 \cap L_2$:

$$L1 = \{ w \in \{0\}^* \circ \{1\}^* \circ \{2\}^* \circ \{3\}^* : \#_0(w) - 1\#_1(w) + 2\#_2(w) \geq 2\#_3(w) \}$$

$$L2 = \{ w \in \{0\}^* \circ \{1\}^* \circ \{2\}^* \circ \{3\}^* : \#_0(w) - 2\#_1(w) + \#_2(w) \leq 3\#_3(w) \}$$



Q7.

$L = \{ w \in (\{0\} \circ \{1\}^+)^* :$ the largest entry in the list represented by w is at least twice the smallest in entry in the list represented by w $\}$