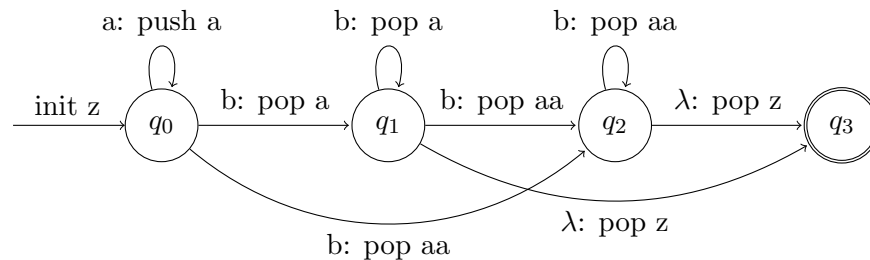


1.

$$\begin{aligned} S &\rightarrow 1SZ|0SE|\epsilon \\ Z &\rightarrow 0S \\ E &\rightarrow 1S \end{aligned}$$

for any string  $w$  of 1's and 0's the CFG can copy the input from left to right, always keeping one  $S$  in the list of variables. While moving left to right any time  $|1| > |0|$  the current state of the CFG will be  $xSZ_1 \dots Z_k x$   $x$  is the generate substring from the start of  $w$  to the current position and  $k = |x|_1 - |x|_0$ . from there anytime a zero is the next character in  $w$  the current  $S$  can go to  $\epsilon$  and the left most  $Z$  can go to  $0S$ , reducing the number of  $Z$ 's in CFG by 1. The inverse is true for  $|0| > |1|$ . Using this parsing method anytime  $|w|_0 = |w|_1$  the only variable in the CFG should be  $S$  which can go to  $\epsilon$  accepting  $w$  as a string in the language. Anytime  $|w|_0 \neq |w|_1$  there will always be at least one  $Z$  or  $E$  in the CFG after all the input is read causing the string to not be in the language.

2.



3.

