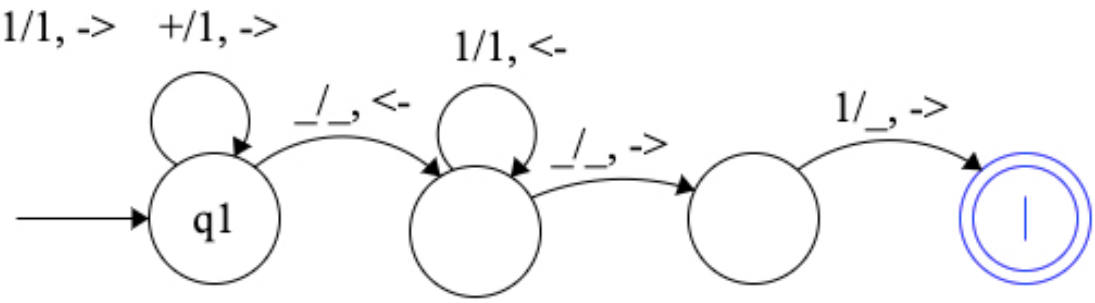
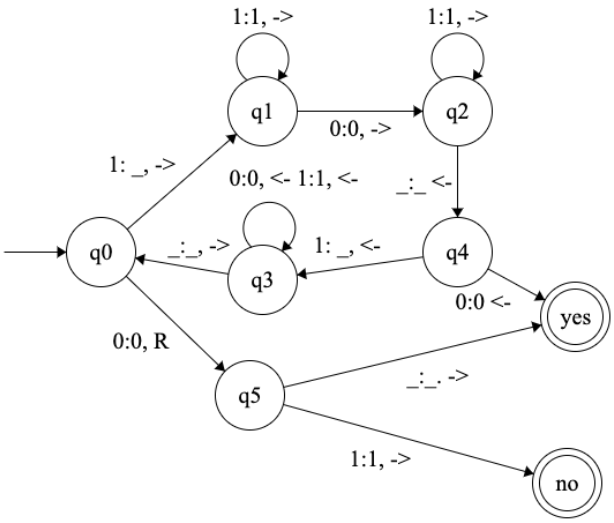
1. Construct a Turing machine that adds two unary numbers, as in Example 8–2, except that it erases the first 1 and changes the separating + to a 1.



2. Draw a Turing machine that takes a string representing two unary numbers, x and y, separated by a 0, and determines whether x ≥ y. For example, the input for x = 3, y = 4 would be 11101111. Use two halt states: one for yes and one for no.

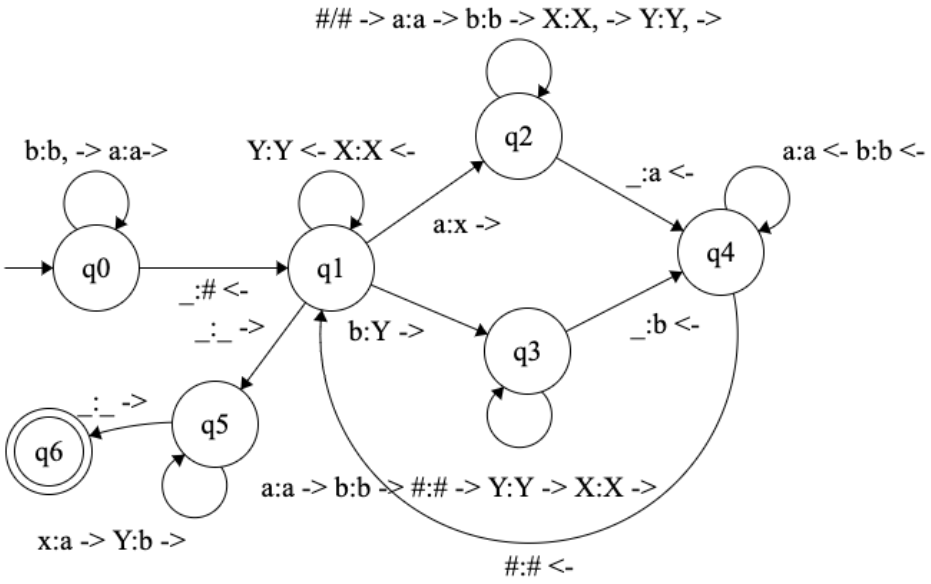


3. Give the trace of your machine in the previous problem processing the strings 11101111 and 11110111.

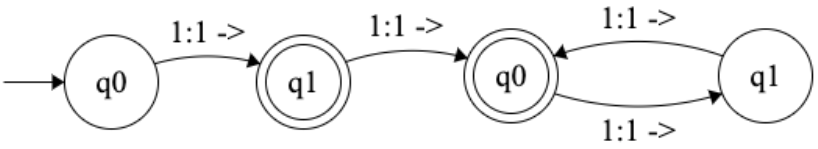
(q0)11101111-_(q1)1101111-_1(q1)101111-_11(q1)01111-_110(q2)1111-_1101(q2)111…._1101111_(q2)-_1101111(q4)-_110111(q3)_-_11011(q3)1_...-(q3)_110111_-_(q0)110111_.....-__110111_(q2)_..._____1011_(q2)__..._____01_(q2)_____...
halt on no

11110111…_111011_….__1101__..___10___... ____0___... halt on yes

4. Draw a TM that computes f(w) = w^R | w ∈ (a + b)*. For simplicity, you can use a delimiter and just copy the letters backwards at the right, so, for example, abb# becomes abb#bba.



5. Draw a TM that accepts unary numbers representing powers of 2. (Hint: Except for 1 and 2, dividing powers of 2 by 2 always gives an even number.)



6. Draw a Turing machine that computes ⌈n/2⌉ for unary numbers.