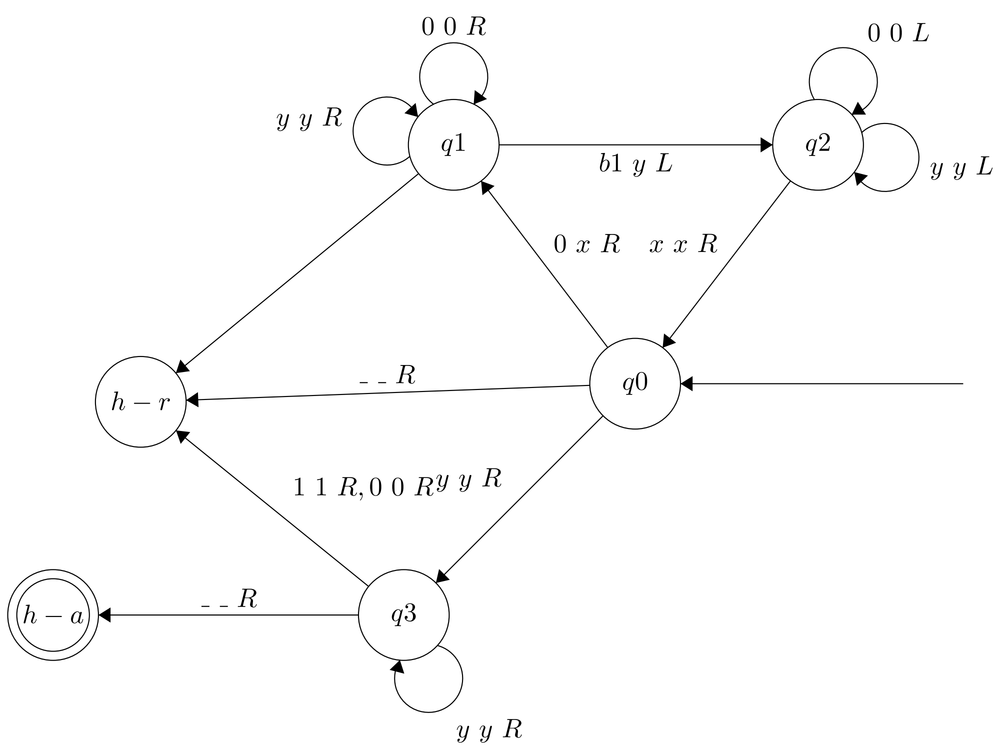
# PS06-03

February 9, 2018

Pictured below is the the $TM-23$ $M$, which can be expressed as $\langle Q, \Sigma, \Gamma, \vdash, \_, \delta, s, t, r \rangle$.
Where $\Sigma = \{0, 1\}$,
$\Gamma = \{\vdash, x, y, \_\} \cup \Sigma$,
$s = q0$,
$t = h - t$,
$r = h - r$.
This machine recognizes $A ::= \{0^n 1^n : n \geq 1\}$, which no DFA can recognize.




An example of a regular language that cannot be recognized by a $TM-23$ machine is $B ::= \{0^n : n \mod 29 = 7 \text{ or } 13\}$ A DFA would have 29 states, 2 of which would be accept states(the 7th and the 13th). The efficient way a turing machine calculates mods is by removing the divisor's amount of 0s until there is fewer than that number, at which point, the remaining number of 0s is the answer. The machine could use the same strategy as the DFA to count 0s, but that would result in too many states. Another way would be to use a counter, like the one in the online Turing simulator used last problem set(for the binary to decimal converter), which would remove zeros and check to see if the decimal counter of zeros was 7, 13 or 29. If it was 7 or 13, it would halt-accept, and if it was 29, it would reset the number to 0. The tape alphabet would be 0-9, $\vdash$ and $\sqcup$. The Tape alphabet length does not violate the $TM - 23$ rule, but the complexity of the system is too much to be expressed in 23 states.