Question 1: A Post-Turing Program for Caesar’s Code

Caesar’s code is probably the simplest way of encoding a message, and it can also be cracked very easily. Let us consider its easiest form for an alphabet \( A = \{a, b, c\} \). For a given input string, we replace each letter ‘a’ with ‘b’, each letter ‘b’ with ‘c’, and each letter ‘c’ with ‘a’. The resulting string is the encoded message.

Write a Post-Turing program that computes this encoding for a given input string \textit{strictly}.

Question 2: From Quadruples to Quintuples

Consider the following Turing machine on the alphabet \( A = \{a, b\} \):

\begin{align*}
q_1 & \text{ a R } q_2 \\
q_1 & \text{ b R } q_2 \\
q_1 & \text{ B R } q_2 \\
q_2 & \text{ a b } q_1 \\
q_2 & \text{ b a } q_1
\end{align*}

(a) Describe in words the function that this machine computes. Is it a strict computation?

(b) Translate this machine into an equivalent quintuple Turing machine using exactly the scheme that we discussed in class. Write down all resulting quintuples (no state transition diagram necessary).
Question 3: The Turing Machine Competition!

Build a Turing machine on the alphabet $A = \{a, b\}$ that computes a function $f(x)$ strictly. $f(x)$ outputs only the symbol “a” if $x$ is a palindrome and outputs only the symbol “b” otherwise. A palindrome is a symmetrical word, i.e., if we reverse the order of its letters, it is still the same word. For example

- $f(bba) = b$
- $f(baab) = a$
- $f(abab) = b$
- $f(aaa) = a$
- $f(babab) = a$
- $f(0) = a$

**Hint:** Your machine could start by checking the first symbol of the string and, depending on the result (‘a’ vs. ‘b’), take one of two paths. In each path, it first deletes the current symbol, moves all the way to the rightmost symbol of the string and checks whether it is the same symbol as the original leftmost one. If so, the program deletes the rightmost symbol, goes back to the (new) leftmost symbol, and repeats the whole thing. Otherwise, it removes all symbols from the tape, outputs “b”, and terminates. If no more symbols are one the tape without having found any mismatch, the program outputs “a” and terminates.

Write down the Turing machine in quadruple notation and as a state transition diagram. Also give the sequence of configurations during the computation of $f(aba)$.

Whoever builds the Turing machine with the fewest internal states that correctly computes $f$ will get some bonus points!

You can test your machines using the Haskell code that I will upload.