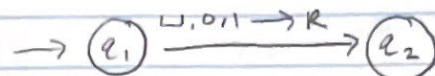


Homework 7

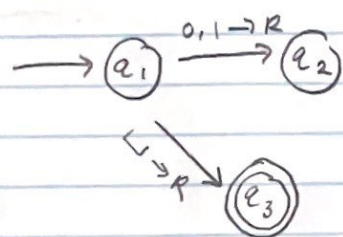
Problem 1

1. $A = \emptyset = \{\epsilon\}$



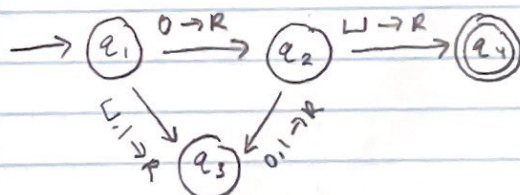
q_1 start state, q_2 reject state

2. $B = \{\epsilon\}$



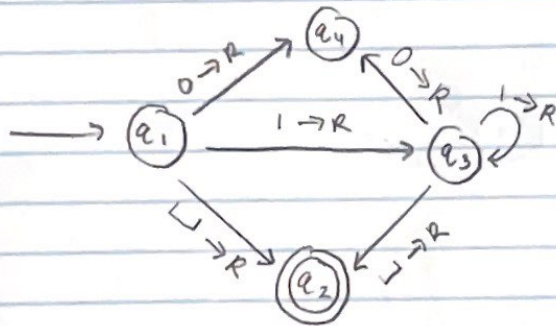
q_1 start state, q_2 reject state, q_3 accept state

3. $C = \{0\}$



q_1 start state, q_3 reject state, q_4 accept state

4. $D = \{1\}^*$



q_1 start state, q_4 reject state, q_2 accept state

2.1

This Turing machine works by first converting the string's 0's and 1's in the first half of the string to a's and x's respectively and converting the 0's and 1's in the second half of the string to b's and y's respectively. Suppose the first character in the string is 1 and the last character of the string is 0. It will first convert the first character to x and then immediately convert the last character of the string to b. It will then go back to the first half of the string and convert the second character and then immediately the second to last character. It will work inward to convert the string. Once the string is converted to just a's, b's, x's, and y's, it will attempt to match both halves of the string. It will first try to match the last character in the first half of the string to the last character in the second half of the string. For example, if the last character of the first half of the string is a and the last character of the second half of the string is b, then it will first override a to ! indicating we have already visited the character and then go to the last character of the second half of the string. If it happens to be b which in this case it is it will override b to # and then head back to the second to last character in the first half of the string. Suppose if the last character in the second half of the string wasn't b then it would reject. The reason we convert the string's characters to a's, b's, x's and y's is so we can know what 0's and 1's are in the first half of the string and what 0's and 1's are in the second half of the string. This is what allows us to effectively match both halves of the string. The Turing machine will attempt to match the last character of the first half of the string to the last character of the second half of the string and then it will move onto the second to last character of the first half to the second to last character in the second half and so on and so forth. If there happens to be a mismatch at any point the Turing machine will reject. The tape alphabet is the set $\{0, 1, a, b, x, y, !, \#, _ \}$.

