BLG 311E - Formal Languages and Automata
Assignment #2

"Why did the NFA break up with the DFA?"

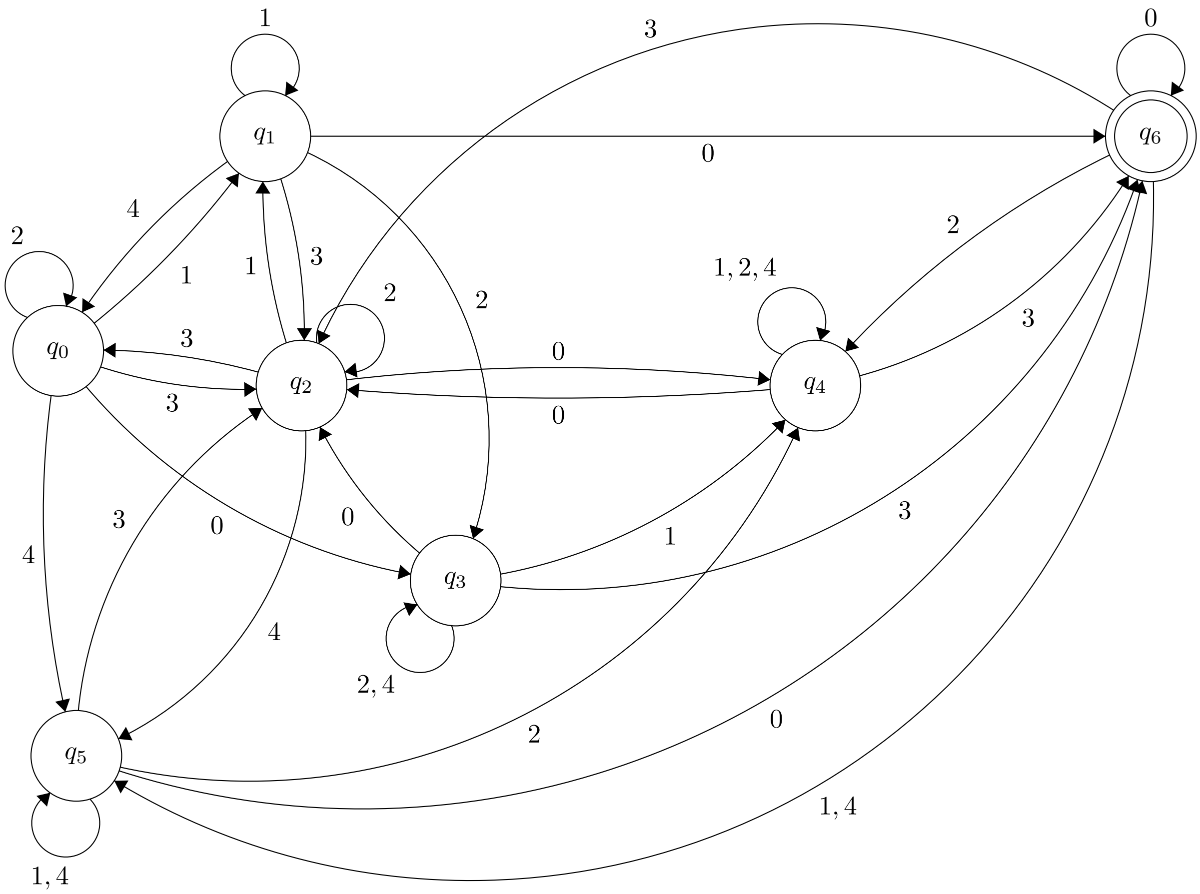
Because the NFA couldn't commit to a single path.

For any issues regarding the assignment, please contact Doğukan Arslan (**arslan.dogukan@itu.edu.tr**).

Q1. You have discovered a new type of restriction enzyme called *Formalase* that cuts DNA at specific recognition sequences. When your enzyme encounters a DNA sequence that starts with a "T" and ends with a "G", with an odd number of "A"s and an even number of "C"s (including zero), it cuts the DNA at specific positions, resulting in two fragments of DNA with "sticky ends" that can be used in molecular biology techniques such as cloning, PCR, and DNA sequencing.

To model the behaviour of Formalase, create a Deterministic Finite Automaton (DFA) by drawing state transition diagram and table. Keep in mind that a DNA sequence consists of A, T, G, and C. Possible acceptable strings are TAG, TACCG, TCCACCG, TCCAAAG etc.

Q2. A software developer was tasked with writing a program to control a complex system of interconnected sensors. To implement this, the developer created a DFA to model the behavior of the system. However, the developer soon discovered that the DFA was very large and difficult to manage, with many states depending on the values of multiple other states. This made it hard to reason about the behavior of the system and identify potential failure modes. Help the developer by making the system simpler. (Note: Initial state is q_0)



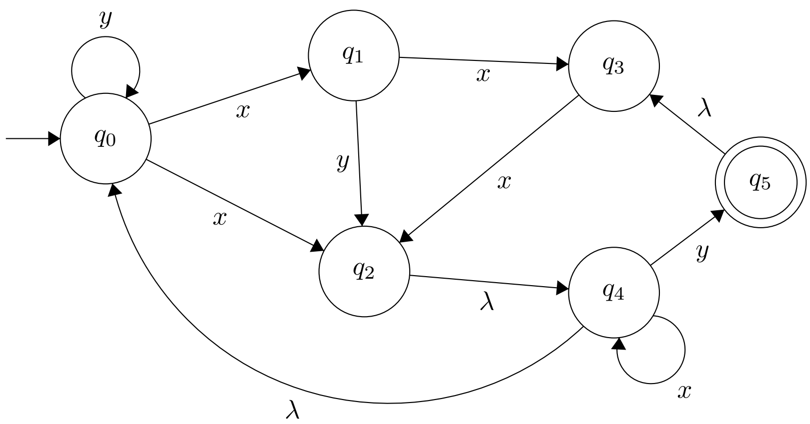
q_0
q_1
q_2
q_3
q_4
q_5
q_6
1
3
0
4
2
1
1
3
2
2
3
3
0
0
1, 2, 4
2
3
3
0
0
4
1
3
4
2, 4
2
0
1, 4
1, 4

Q3. You work as a security guard at a factory that produces cookies shaped in letters. The cookie-making machine sometimes malfunctions and produces cookies with the letter combination "OOPS". To ensure that only these cookies are thrown away, you need to create a NFA that recognizes the language of all "OOPS" cookies. The cookies have an even number of "O"s followed by an odd number of "P"s, and they must end with "S". Draw a NFA that represents the language of these cookies. Possible acceptable strings are OOPSS, OOOOPPPS etc.

Q4. A group of adventurers discovered an ancient temple deep in the jungle. Inside the temple, they found a treasure map written in a strange language consisting only of the letters 'x' and 'y'. However, the map was encrypted with a code that required the adventurers to solve a puzzle.

The puzzle consisted of a machine that would only unlock if the adventurers entered a string of 'x's and 'y's that followed a very specific pattern. The pattern was described by a strange diagram on the wall, which is given below, that recognized the language of valid input strings.

The adventurers quickly realized that given diagram resembles a DFA (obviously not) but they could not figure out how it works since they had not completed the non-mandatory homework assignments during the Formal Languages and Automata lectures. Help the adventurers by converting given diagram to a more understandable automata (for them) so that they can unlock the puzzle and find the treasure.



Q5. Stephen Cole Kleene was an American mathematician, born on January 5, 1909. One of Kleene's most famous contributions to computer science was the development of regular expressions, which are now widely used in programming languages and text editors. Kleene Theorem states that:
"Any language is regular which is constructed by applying closed language operations on regular languages. e.g. if we know L_1 and L_2 are regular languages, any language built by applying a closed operation on them produces a new regular language."

For the union ($L(M_1) \cup L(M_2)$), intersection ($L(M_1) \cap L(M_2)$), complement ($L(\overline{M_2})$), concatenation ($L(M_1) \cdot L(M_2)$), Kleene star ($L(M_2)^*$), reversal ($L(M_2')^R$), homomorphism ($h(L(M_1)) \rightarrow h(a) = \lambda, h(b) = 101$), and inverse homomorphism ($h^{-1}(L(M_1)) \rightarrow h(0) = \lambda, h(1) = aab$) operations and given diagrams of machines M_1 and M_2, state whether regular languages recognized by a finite automaton is closed under that operation or not. If so, draw the resulting state transition diagram.

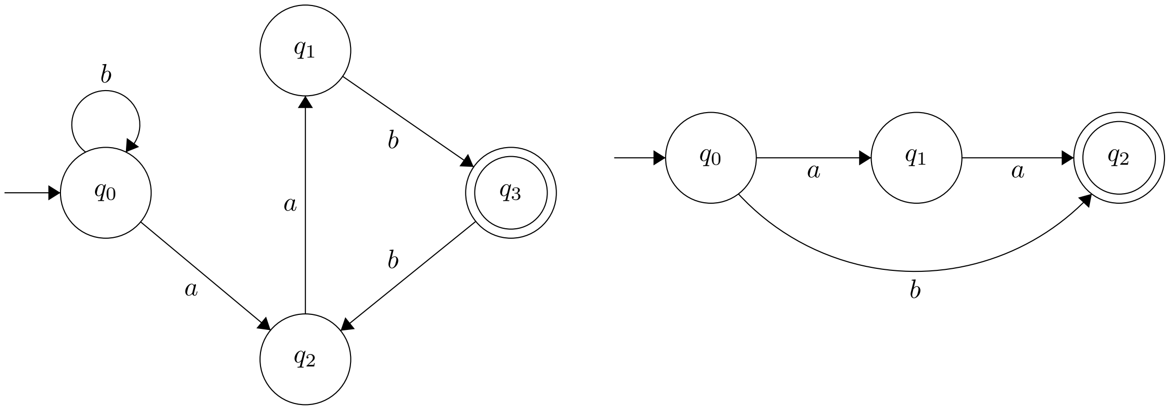
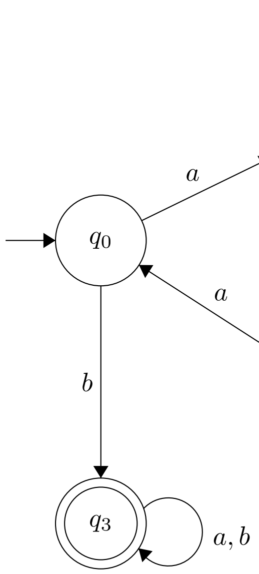


Figure 1: State transition diagrams for M_1 and M_2.

Q6. For the following regular expressions; a^+b^+ and abb^*a^*, create another regular expression that does not recognize the intersection set of two given regular expressions. You can follow these steps:

1. Create an NFA systematically from the two given regular expressions. You may simplify the resulting NFAs heuristically to make the following steps easier.
2. Convert the resulting NFAs to DFA.
3. Intersect the two DFAs obtained from the NFAs.
4. Take the complement of the intersection obtained in the previous step.
5. Convert the resulting automaton to a regular expression.

Q7. Your computer got attacked by a Petya malware, which is type of ransomware that encrypts files in your computer and overwrites the master boot record (MBR) of your computer, making it unable to boot. You paid the ransom in exchange for the decryption key to unlock the files and the hackers gave you the following DFA but not the key itself. Find the regular language recognized by the given DFA *systematically* so that you can save your Formal Languages and Automata homework, which is in your computer.



Q8. Consider the given languages and apply pumping lemma to prove that given languages are not regular.

- $L = \{0^n 1^n 2^n | n >= 0\}$,
- $L = \{0^m 1^n | n > m\}$