1. Introduction

A language is said to be regular if it has a corresponding Deterministic Finite Automaton (DFA). To show that L^R is a regular language, we have to show that we can construct a Finite Automaton (or more specifically, we'll use an NFA that can be transformed into a DFA) that accepts

L^R={ w^R ∣ w ∈ L (M) }

2. Formal Construction

Given a DFA M described by the 5-tuple

M=(Q,Σ,δ,q0,F)

we create a Non-deterministic Finite Automaton

M^R = (Q′, Σ, δ′, S, {q0})

L^R as follows:

- **States (Q′)**: The states of M are used with the addition of a new initial state S, Q′=Q∪{S}.
- **Alphabet (Σ)**: The alphabet stays the same.
- **Transitions (δ′)**: Reverse all transitions of M.

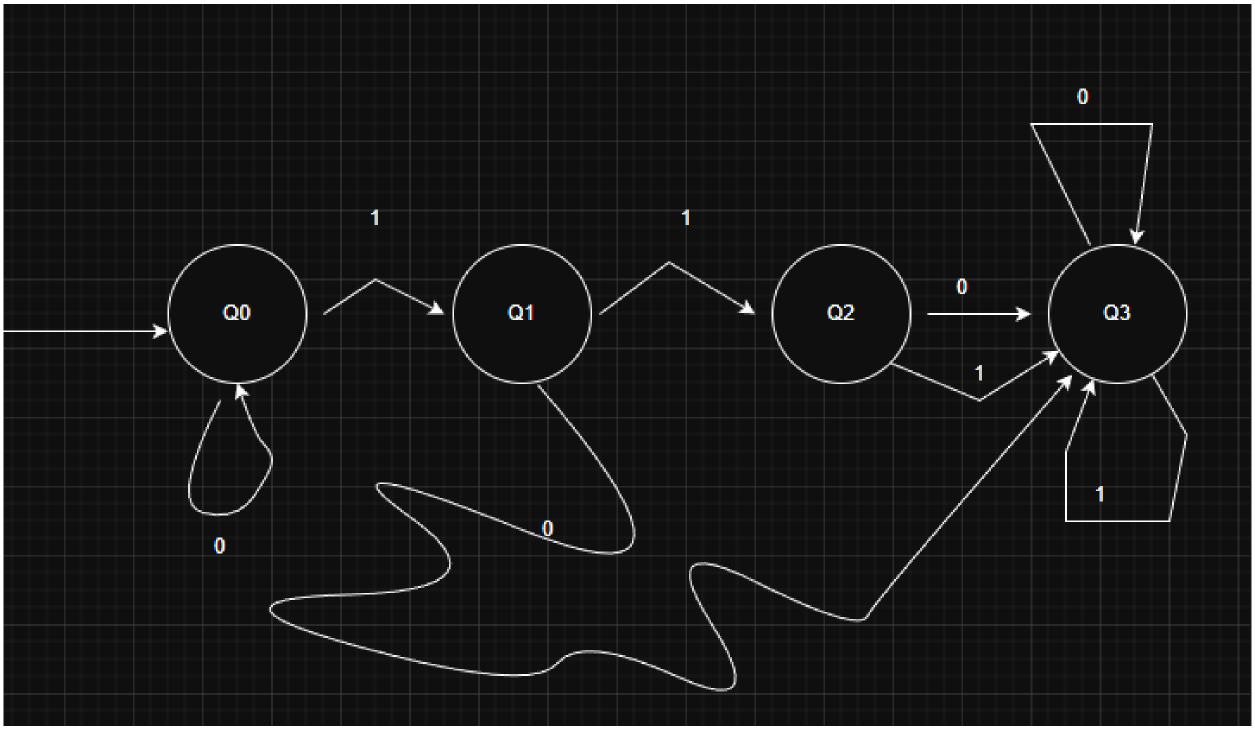
  If δ(p, a) = q in M, then in M^R there is a transition q --a--> p.
- **Start State (Q0′)**: The new initial state is S.
- **Accepting State (F′)**: The original initial state of M is the sole accepting state of M^R. F′= {q0}
- **Initial ϵ-transitions**: Because we have multiple possible end points in M, we add transitions from S to all states that are accepting states of M.

  δ′(S,ϵ)={q∣q∈F}.

3. Visual Example

A comparison of an automaton and its reverse follows from the diagrams given.

- **Original Automaton (DFA M)**:
  The automaton starts from Q0 and proceeds forward from Q1 through Q2 and



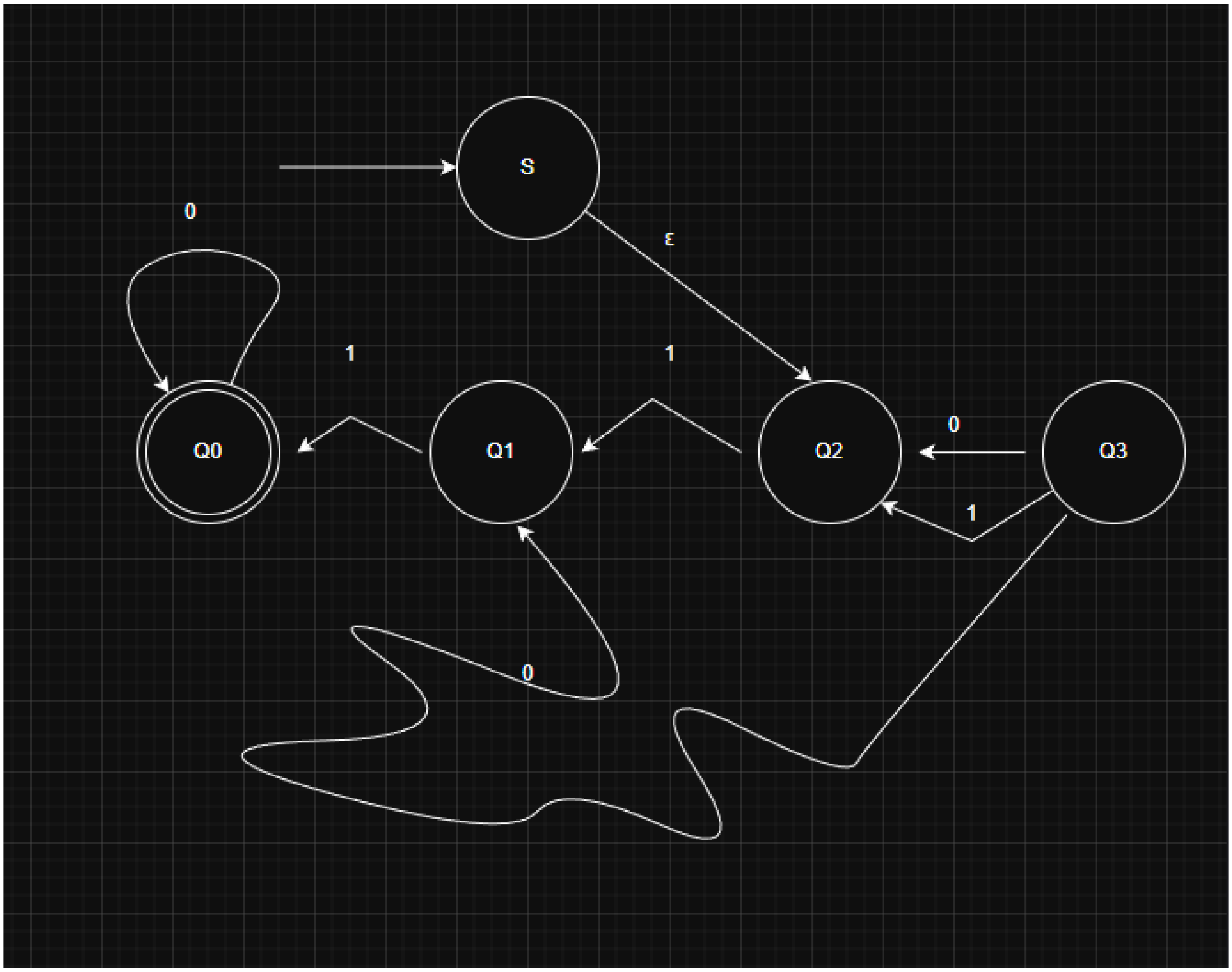
terminates at the accepting state Q2.

- **Reversed Automaton (NFA M M^R)**:
  From ReversedFigure.png, we have:

  - There is a new initial state S.

  - There is an ϵ-transition from S to the original accepting state (Q2 in the original automaton).
  - The arrows are reversed. For example, from Q0 to Q1 on '1', we have from Q1 to Q0 on '1'.

  - The only accepting state of the reversed NFA is Q0, which is the original start state of the DFA, and is marked with a double circle.



4. Conclusion

Since we could build an NFA, M^R, that accepts L^R, and every NFA has an equivalent DFA, we now know that L^R is a regular language. This completes the proof that regular languages are closed under reversal.

References

1. Campbell, R. COMP 382 - Section 1.1: *Finite Automata* (lecture slides). University of the Fraser Valley.
   Referenced for the formal definition of a DFA — $M = (Q, \Sigma, \delta, q_0, F)$ — including how languages are represented (L(M)) and standard conventions for accepting states.

2. Campbell, R. COMP 382 - Sections 1.2–1.3: *Nondeterminism and Regular Expressions* (lecture slides). University of the Fraser Valley.
   Provided the foundation for understanding NFAs, ε-transitions, and the equivalence

between DFAs and NFAs, which supports the reversed-automaton construction.

3. Sipser, M. (2013). *Introduction to the Theory of Computation* (3rd ed.). Cengage Learning.
   Used for exploring closure properties of regular languages and formal definitions in automata theory.

4. diagrams.net (draw.io) — https://app.diagrams.net/
   Used to create diagrams for both DFA and reversed NFA representations.

5. ChatGPT (OpenAI)
   ChatGPT was used for minor clarification and formatting assistance.