Automata Theory – Make-Up Assignment

225092R – Rathnayaka S J

Sri Lankan National Identity Card (NIC) Validator Using DFA

1. Introduction & Background

National Identity Card (NIC) numbers are a core component of identity verification in Sri Lanka. They are widely used in banking systems, government services, educational institutions, healthcare systems, and election-related processes. Ensuring that NIC numbers follow the correct official format is essential to prevent data entry errors, reduce identity fraud, and maintain database consistency.

Sri Lanka currently uses two NIC formats: -

1.Old NIC format: 9 digits followed by a letter V or X

2. New NIC format: 12 digits

This project implements a Deterministic Finite Automaton (DFA) to validate both NIC formats. DFA is chosen because NIC validation depends only on a fixed sequence of characters and does not require memory beyond the current state.

2. Problem Definition

National Relevance

Automated NIC validation helps government and private institutions quickly verify identity data at the point of entry. A DFA-based validator ensures correctness, efficiency, and transparency, making it suitable for real-world deployment in Sri Lankan systems.

Input Alphabet (Σ)

Σ = {0–9, V, X}

Constraints

- Only digits are allowed in numeric positions
- Letter V or X is allowed only at the end of old NIC format
- No extra characters are allowed
- Validation must be performed using automata logic (no regular expressions or length checks)

3. Automata Model

Type of Automaton -Deterministic Finite Automaton (DFA)

Formal Definition

The DFA is defined as:

$M = (Q, \Sigma, \delta, q_0, F)$

- Q (States): q0, q1, q2, q3, q4, q5, q6, q7, q8, q9, q10, q11, q12, q13, qReject
- Σ (Alphabet): {0–9, V, X}
- q_0 (Start State): q0
- F (Accepting States): {q10, q13}

Meaning of Accepting States

- q10: Valid old NIC (9 digits followed by V or X)
- q13: Valid new NIC (12 digits)

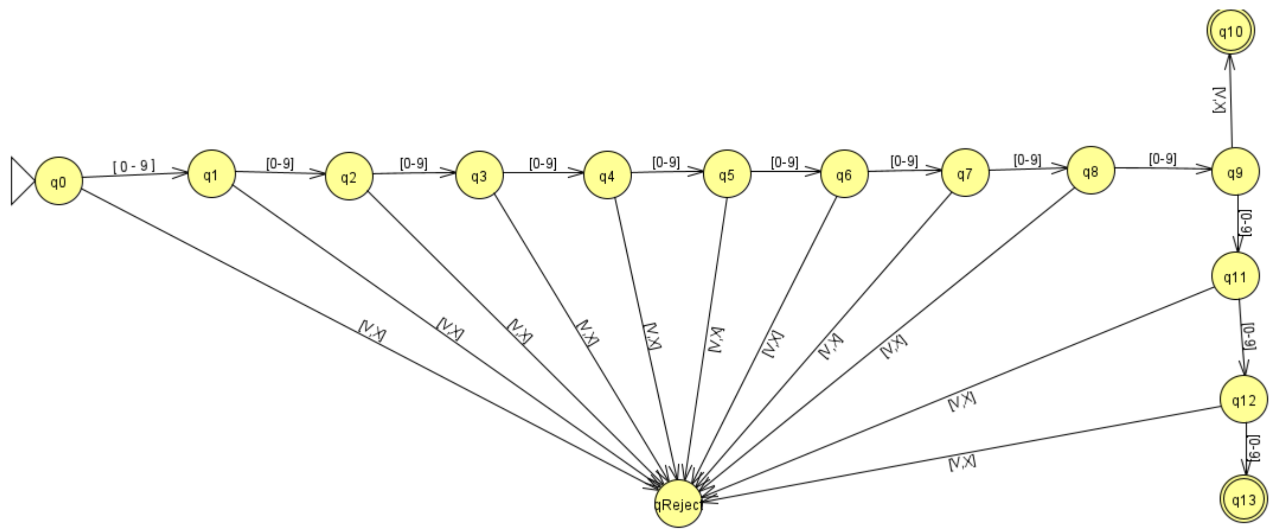
4. Transition Function (δ)

Selected transition rules:

- δ(q0, digit) = q1
- δ(q9, V) = q10
- δ(q9, X) = q10
- δ(q9, digit) = q11
- δ(q12, digit) = q13
- δ(q10, any) = qReject
- δ(q13, any) = qReject
- δ(qReject, any) = qReject

The DFA implicitly enforces correct length by allowing acceptance

only at q10 or q13.



5. Implementation Details

The DFA is implemented in Python using conditional state transitions. Each character is processed sequentially, and the current state is updated based on the transition rules.

Key Characteristics

- No regular expressions used
- No explicit string length checking
- Deterministic transitions
- Dead state ensures DFA completeness

The program outputs either: -

1.ACCEPT – Old NIC

2ACCEPT – New NIC

3.REJECT

6. Evaluation & Testing

The system was tested using: - Valid old NIC numbers - Valid new NIC numbers - Invalid cases (wrong characters, wrong order, extra characters, missing terminator)

All valid cases reached accepting states, and invalid cases correctly transitioned to the reject state.

Complexity Analysis

- Time Complexity: O(n)

- Space Complexity: O(1)

7. Conclusion

This project successfully demonstrates how a Deterministic Finite Automaton can be applied to solve a nationally relevant problem in Sri Lanka. The DFA accurately validates both old and new NIC formats using pure automata theory principles. The solution is efficient, reliable, and suitable for real-world identity validation systems.