**📝 Mid-Semester Quiz: DFA & NFA (JFLAP-Focused)**

**Course:** Automata Theory  
**Duration:** 60 Minutes  
**Total Marks:** 100  
**Resources:** JFLAP installed, pen/pencil for written parts

**Section A: Conceptual Questions (20 marks)**

Answer **all** questions. Each question is worth **5 marks**.

1. Define a Deterministic Finite Automaton (DFA) and describe one key difference between DFA and NFA.

A Deterministic Finite Automaton (DFA) is a theoretical model of computation used to recognize regular languages. It consists of a finite set of states, an input alphabet, a transition function that determines the next state based on the current state and input symbol, a start state, and a set of accepting (or final) states.

1. Can an NFA recognize a language that a DFA cannot? Justify your answer briefly.

No, an NFA cannot recognize any language that a DFA cannot recognize.

Both NFAs and DFAs recognize exactly the same class of languages the set of regular languages. Every NFA can be converted into an equivalent DFA using the subset construction method, which may cause an exponential increase in the number of states but does not change the class of languages recognized. Therefore, NFAs do not have the computational power to recognize languages outside of those recognized by DFAs.

1. Describe how ε-transitions affect the behavior of an NFA.

They increase the flexibility of the automaton, making it possible for the NFA to be in multiple states simultaneously (via epsilon closures).

1. Why might converting an NFA to a DFA cause a “state explosion”? Explain with a brief example.

converting an NFA to a DFA cause a “state explosion” because the process involves constructing new states in the DFA that represent sets of states from the NFA. Since every subset of NFA states may correspond to a unique DFA state, the number of DFA states can be exponential relative to the number of NFA states.

**Section B: JFLAP Design & Construction (50 marks)**

Use **JFLAP** for all design-based tasks. Submit screenshots of your automata with your answers.

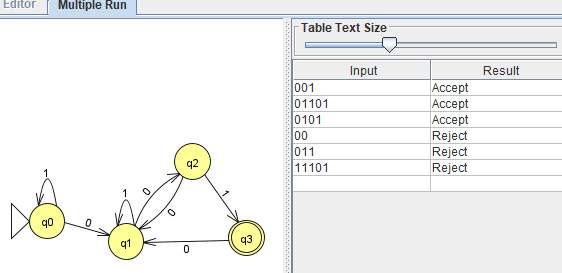
**1. DFA Construction (25 marks)**

**Task:** Construct a DFA that accepts all binary strings over {0,1} that **end with 01 and have an even number of 0s**.

* Create and label your states (minimum 4–5 expected).
* Ensure only valid strings like 01, 1101, 00101 are accepted.
* Invalid examples: 00, 011, 11101.

**Marks Breakdown:**

* Correct transitions (10)
* Proper final states (5)
* Test inputs: Provide results for 3 valid and 3 invalid strings (5)
* Screenshot of DFA (5)



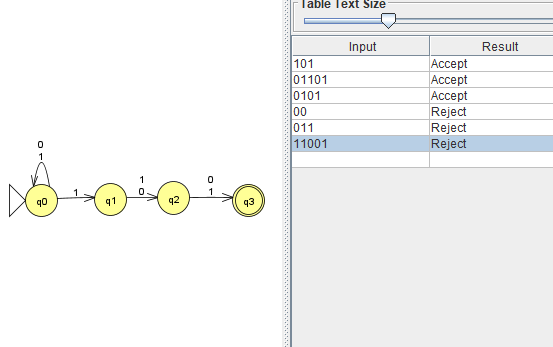
**2. NFA Construction (25 marks)**

**Task:** Design an NFA that accepts strings where **the third symbol from the end is a 1** (e.g., 1010, 110, 01101).

* Use non-determinism to “guess” where the third-last symbol is.
* Use at least 3–4 states and clear transitions.

**Marks Breakdown:**

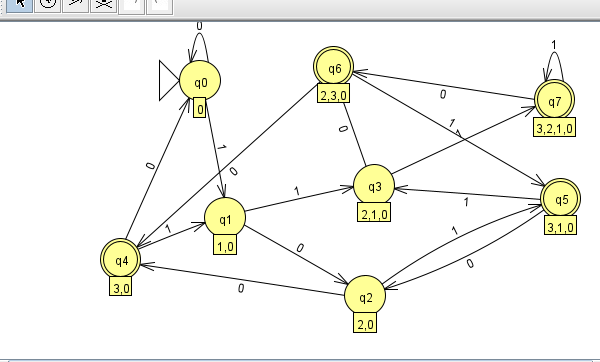
* Correct NFA logic and structure (10)
* Use of non-determinism correctly (5)
* Test inputs: Provide results for 3 accepted and 3 rejected strings (5)
* Screenshot of NFA (5)



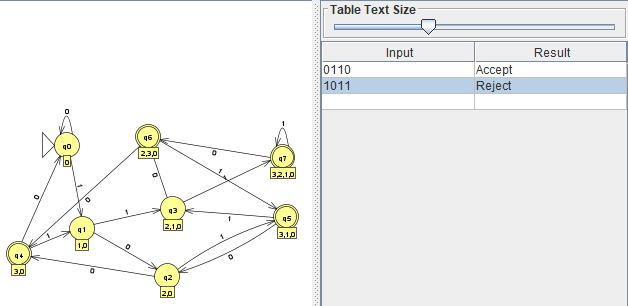
**Section C: Analysis and Conversion (30 marks)**

**1. Convert the NFA (from Section B) to a DFA using JFLAP (15 marks)**

* Use: Convert → Convert to DFA → Do Step
* Provide:
  + Number of DFA states generated: 8
  + Screenshot of the resulting DFA



* + Input test result for string 0110 and 1011



**Marks Breakdown:**

* State conversion accuracy (5)
* Screenshot (5)
* Input test results (5)

**2. Explain the differences between your NFA and the equivalent DFA (15 marks)**

Discuss:

* Number of states:

The DFA had a greater number of states than the NFA

* Readability

The NFA was easier to read and understand

* Ease of design

It was easier to design the NFA

* Determinism

The NFA was not deterministic.

**✅ Submission Checklist**

* Answers to Section A (written)
* Screenshot of DFA with labeled states
* DFA test strings and results
* Screenshot of NFA
* NFA test strings and results
* Screenshot of converted DFA
* Written answers for Section C analysis