



Term Project Brief

Individual Assessment with weight 30%

Objective

Design a project that bridges automata theory with real-world applications through **research** and **implementation**, fostering critical analysis, coding skills, and collaborative problem-solving.

The suggested project consists of two parts:

The first part is intended to be mini-research project on a real-life application of automata models.

The second part aims to implement a simulation, using any programming language, an automata model, or to design a real-life application using automata models, which is always considered a good practice. It helps refine conceptual understanding and enhance problem-solving skills.

Description

You can work in groups of up to six students. Each group should submit a copy of the project.

First part (research part):

Every team will be required to choose a research topic on an application of automata theory in real-life. The report should include a description of the research topic, background and problem statement, existing solutions, a comparison between the state-of-the-art methodologies proposed to solve this problem, and future work that may be promoted.

The report is to be written in IEEE-paper format (6-8 pages). It should have:

An abstract: Clearly state the problem, methodology, and contribution,

An introduction: Motivate the problem (e.g., "Why finite automata are efficient for a certain real-time application?"),

A related works: Compare 3–4 state-of-the-art approaches (e.g., automata-based vs. ML-based),

Analysis: Discuss strengths/weaknesses of automata-based solutions (e.g., speed vs. adaptability),

Conclusions and Future work, and

Proper references list.

Suggested Topics (Topics could include, but not limited to)

- **AI-Powered Chatbots:** Using finite automata for natural language understanding.
- **DNA Computing and Automata:** Exploring biological computing models based on automata theory.
- **Cybersecurity Applications:** Intrusion detection using pattern matching with finite automata.
- **Automata in Blockchain:** Modeling consensus algorithms using state machines.
- **Human-Computer Interaction:** Gesture recognition using pattern recognition automata.
- **IoT and Smart Devices:** Automata-driven device control and automation.
- **Automata in Robotics:** Motion planning and behavior modeling using state machines.
- **Finite Automata for Real-Time SQL Injection Detection:** How can DFAs efficiently detect SQL injection patterns in web traffic?
- **Patient Monitoring Systems with Timed Automata:** Using timed automata to model and detect abnormal vital sign patterns (e.g., irregular heartbeats).

Second part (implementation part):

Students must design and implement a system related to implement an automata model machine for a real-life system that uses automata models. The document should clearly describe the design and implementation of your system, as well as how to install and use your system.

Deliverables Each project submission must include:

- **Project Report:** A detailed document explaining the problem statement, theoretical background, implementation details, and results.
- **Source Code:** Well-documented code with comments explaining key components.
- **Graphical User Interface (GUI):** A user-friendly GUI must be included to interact with the system.
- **README File:** Instructions for setting up and running the project.

The project report should have three parts:

1. **Relevance:** a short (one sentence) summary of how your project relates to the class.
2. **Abstract:** Short summary describing the idea of the project (two sentences)
3. **Content:**
 - **Introduction:** A general description of the area of your project and why you're doing it.
 - **Problem Specification:** A clear and brief technical description of the problem you're addressing.
 - **Approach:** A description of how you went about trying to solve the problem. A short readme describing the implemented machines, the used programming language, the accepted language, etc.
 - **Results and Analysis:** What happened when you evaluated your system. Screen shots of the outputs of the program
 - **Conclusions:** What did you learn from doing the project? What did you demonstrate about how to solve your problem?
 - **References:** Complete list of sources you used in completing your project, with explanations of what you got from each.

Implementation Requirements

- The project must be implemented in a programming language such as Python, Java, or C++.
- The system should take user input and provide appropriate outputs.
- The implementation should correctly model the chosen automaton or theoretical concept.
- The project should include a testing mechanism to validate correctness.
- A GUI must be included to ensure ease of interaction with the implemented system.

Suggested Topics (Topics could include, but not limited to)

1. **Chatbot with Pattern Matching:** Using finite automata to recognize and respond to user queries.
2. **Spell Checker:** Utilizing regular expressions and finite state machines to detect misspelled words.
3. **Vending Machine Simulation:** Designing an FSM to handle product selection and transactions.
4. **Web Form Validator:** Employing regular expressions for input validation.

5. **PDA-based Arithmetic Expression Evaluator:** Parse and evaluate expressions with parentheses.
6. **Cybersecurity Intrusion Detection:** Applying automata theory to recognize security threats.

It is your responsibility to come up with reasonable test data.

- Presentations are demanded from all students (presentation for 10 to 15 min.).

Evaluation Scheme

The grading of the project is based on the following scheme:

• Interest and originality of the research idea	2
• Research report as described above	10
• Technical report as described above	5
• Running the proposed automata model	5
• Discussion	8

Submission Dates

Deliverable	Submission Method	Deadline
Reports & code	Turn it in	Week 12
Discussion & practical run	15 minutes	Week 13

Submission Guidelines

- Submit all files in a compressed folder ([team number].zip).
- Include all dependencies and a guide for running the project.
- Submission Platform: Teams