Computational Theory Project Description:

Scanpath Pattern Recognition Using Automata Theory

Computational Theory Course — Fall 2025

Mohammed VI Polytechnic University
UM6P College of Computing Rabat Campus

November 13, 2025

1 Introduction

This document outlines the specifications for the UM6P Computational Theory course research project that will allow you to apply the theoretical concepts you have learned in this course to a real-world problem which is analyzing how medical professionals read electrocardiograms using eye-tracking data. In this project, you will use formal models from automata theory (finite automata, pushdown automata, context-free grammars, and other computational structures) to recognize and classify patterns in eye movement data. Specifically, you will analyze the sequential viewing patterns (called *scanpaths*) that medical experts use when interpreting 12-lead ECGs (See definition in the next slide). Figure 1 shows an example of such a scanpath visualization, where you can see the complex sequence of eye fixations (blue circles numbered in order) as a physician systematically examines different parts of the ECG.

This project will not only help you build and implementing new algorithms from the course, but will also help you conduct original research at the intersection of computer science, medicine, and cognitive science. Your work could potentially be published at international conferences, giving you early experience with the academic research process. More importantly, you will see firsthand how theoretical computer science concepts like regular languages, finite automata, and formal grammars have powerful applications in solving real problems.

This project is intentionally open-ended within a focused problem domain. All teams will work on scanpath pattern recognition for ECG interpretation, but you will each explore different formal models and approaches. Some of you might use finite state automata for simple pattern matching, while others might employ pushdown automata to capture hierarchical viewing strategies, or context-free grammars to model the compositional structure of expert reading patterns. This diversity will enrich our class discussions as you present your different approaches and results.

1.1 Learning Objectives

By completing this project, you will be able to:

- Apply formal models (finite automata, pushdown automata, context-free grammars, Turing machines) to real-world pattern recognition problems
- Design and implement algorithms for sequential pattern analysis
- Analyze the computational complexity of pattern matching problems
- Work effectively in research teams and manage collaborative software projects
- Communicate technical research findings through written papers and oral presentations

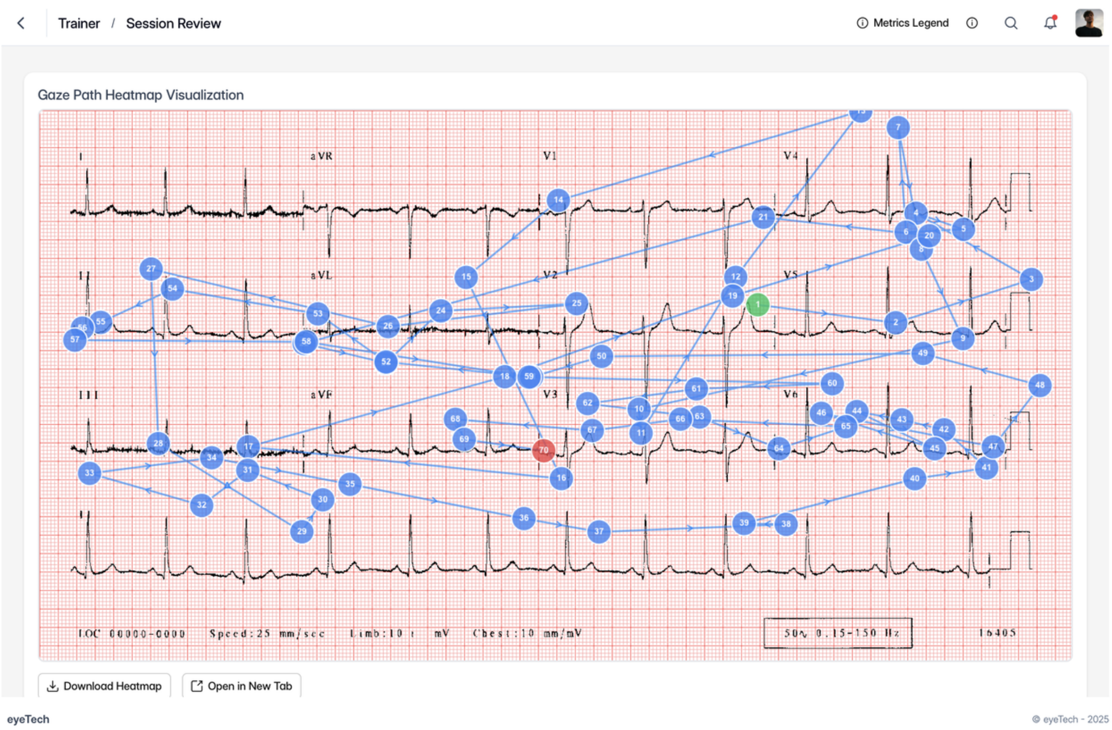

Figure 1: Gaze path heatmap visualization showing the sequential scanpath of a medical professional reading a 12-lead ECG. Blue circles represent fixation points numbered in temporal order, with circle size indicating fixation duration. The green circle highlights a critical diagnostic fixation. Lines connecting fixations illustrate the saccadic eye movements between different ECG leads and features. This visualization demonstrates the complex, non-linear pattern of visual exploration during ECG interpretation.

- Use professional tools like Git/GitHub for version control and collaboration
- Navigate the academic publication process by submitting to international conferences
- Critically evaluate research papers through the peer review process

2 Background and Motivation

2.1 What is a 12-Lead Electrocardiogram?

An electrocardiogram (ECG or EKG) is a medical test that records the electrical activity of the heart over time. A standard 12-lead ECG captures cardiac electrical signals from twelve different perspectives (leads), providing a comprehensive view of the heart's electrical behavior. These twelve leads are organized into:

- **Limb leads** (I, II, III): Record electrical activity in the frontal plane
- **Augmented limb leads** (aVR, aVL, aVF): Enhanced frontal plane recordings
- **Precordial (chest) leads** (V1–V6): Record electrical activity in the horizontal plane

The 12-lead ECG displays these signals simultaneously, typically arranged in a 4×3 grid format, allowing clinicians to assess cardiac rhythm, identify myocardial infarctions, detect conduction abnormalities, and diagnose various cardiac pathologies. Accurate interpretation of 12-lead ECGs is a critical clinical skill that requires systematic visual analysis and pattern recognition.

2.2 What is a Scanpath?

A scanpath is the sequential trajectory of eye fixations and saccades that occurs when a person visually explores a scene or image. In the context of ECG interpretation, a scanpath represents the ordered sequence of visual fixations across different regions of the 12-lead ECG display as a clinician performs diagnostic analysis.

Formally, a scanpath can be defined as:

$$S = \langle (r_1, d_1), (r_2, d_2), \ldots, (r_n, d_n) \rangle$$

where r_i represents the spatial region (ECG lead or feature) fixated at position i in the sequence, and d_i represents the duration of that fixation in milliseconds. Each region r_i might correspond to specific ECG components.

2.3 Eye-Tracking Technology and Scanpath Measurement

Eye-tracking technology uses specialized cameras and infrared illumination to precisely measure the point of gaze on a display. Modern eye-tracking systems operate at sampling rates of 60–1000 Hz, providing millisecond-level temporal resolution of eye movements.

The eye-tracking process involves:

1. **Calibration**: Establishing a mapping between eye position and screen coordinates
2. **Fixation detection**: Identifying periods when the eye remains relatively stationary (typically 150–300 ms)
3. **Saccade detection**: Identifying rapid eye movements between fixations
4. **Scanpath construction**: Ordering fixations temporally to create the sequential viewing pattern

When a medical professional views an ECG (as shown in Figure 1), the eye-tracking system records where they look and for how long. This data reveals the cognitive strategies employed during diagnostic reasoning. As you can see in the figure, the viewing pattern is not random—there is structure to how experts systematically examine different parts of the ECG. Your job in this project is to discover and model these patterns using automata theory.

2.4 The Clinical Importance of Systematic ECG Reading

Expert cardiologists and emergency physicians follow systematic approaches to ECG interpretation to ensure comprehensive analysis and minimize diagnostic errors. While multiple valid systematic approaches exist, established clinical guidelines recommend specific sequences for reviewing ECG components and leads.

A commonly recommended systematic approach follows this sequence:

1. **Rate assessment**: Determine heart rate from RR intervals
2. **Rhythm analysis**: Assess regularity and identify rhythm abnormalities (typically viewed in lead II or V1)
3. **Axis determination**: Calculate electrical axis using leads I and aVF
4. **P-wave analysis**: Examine atrial activity across leads
5. **PR interval measurement**: Assess atrioventricular conduction
6. **QRS complex analysis**: Evaluate ventricular depolarization morphology
7. **ST segment evaluation**: Identify ischemic changes or injury patterns
8. **T-wave examination**: Assess ventricular repolarization
9. **QT interval measurement**: Evaluate total ventricular action potential duration
10. **Lead-by-lead systematic review**: Examine specific lead groups (inferior leads: II, III, aVF; lateral leads: I, aVL, V5, V6; septal leads: V1, V2; anterior leads: V3, V4)

This systematic approach ensures that no critical features are overlooked and that findings are interpreted in proper clinical context. Experienced clinicians develop efficient scanpath patterns that implement these systematic strategies while adapting to specific clinical scenarios.

2.5 Why Automata-Based Scanpath Representation?

This section explains "Why use automata theory to analyze these patterns? instead of just use statistics or machine learning?".

2.5.1 The Power of Formal Models

Automata theory provides rigorous mathematical frameworks for describing sequential processes and pattern languages. When modeling a scanpath as a string in a formal language, this helps reframe a complex cognitive and medical behavior into theoretical and logical results about decidability, closure properties, equivalence, and complexity to simplify and accurately represent an inherently complex cognitive problem. This theoretical foundation gives tools that purely empirical approaches lack. A scanpath would fundamentally be a *sequence* of states (eye fixations on different ECG regions) with *transitions* between them (saccadic eye movements), which is exactly what automata models represent. A finite automaton has states and transitions. A pushdown automaton adds memory (a stack) to track context. A context-free grammar generates structured sequences using production rules. These formal models are *designed* for sequential pattern analysis.

2.5.2 Matching Complexity to Model Expressiveness

Different classes of automata have different expressive power, and you consider these research questions as the foundations of your research paper and project:

- Can expert ECG reading patterns be described by a *regular language*? If so, would a simple finite automaton suffice, giving the possibility of linear-time pattern matching.
- Do experts use *hierarchical* strategies (like "examine overview, zoom into detail, return to overview")? Then would a stack memory of a pushdown automaton work better.
- Are there *context-sensitive* dependencies where the interpretation of one fixation depends on distant earlier fixations? This might require more powerful models.

By choosing the appropriate automata class for your model, you balance *expressiveness* (can it capture the patterns?) with *efficiency* (can we recognize patterns quickly?). This is a fundamental trade-off in theoretical computer science that you will experience firsthand.

2.5.3 Classification and Recognition

One of your primary goals is to classify scanpaths. For example:

- **Expert vs. Novice**: Expert cardiologists follow systematic strategies, while novices often have erratic, inefficient patterns. Can you design an automaton that accepts expert patterns and rejects novice ones?
- **Strategy Types**: Different experts might use different valid strategies ("rhythm-first" vs. "morphology-first"). Can you design automata to recognize each strategy type?
- **Completeness**: Did the physician examine all critical ECG components? An automaton that requires visiting all necessary states can detect incomplete readings.

The beauty of automata-based classification is its *interpretability*. Unlike a neural network black box, you can *see* what your automaton is doing. The states and transitions have clear meanings ("currently examining Lead II," "transitioning from P-wave to QRS complex"). This interpretability is crucial for medical applications where we need to understand *why* a pattern was classified a certain way. Formal grammars do not just *recognize* patterns—they can *generate* them. Imagine you have defined a context-free grammar that captures expert ECG reading strategies. You can now:

- Generate example expert scanpaths for training purposes
- Create variations to test the robustness of diagnostic systems
- Synthesize benchmark datasets for evaluating other algorithms

This generative capability is unique to formal language approaches and opens up interesting research possibilities for your project.

2.5.4 Formal Analysis and Proofs

With automata-theoretic representations, you can prove properties about scanpaths:

- **Completeness**: Does every string accepted by your automaton visit all required ECG regions? This can be verified by checking reachability of certain states.
- **Equivalence**: Do two different automata recognize the same set of patterns? Formal equivalence checking algorithms can answer this.

- **Minimality**: Is your automaton the simplest possible for the language it recognizes? Minimization algorithms can optimize your model.
- **Decidability**: Can you algorithmically determine if a given scanpath belongs to the expert pattern language? For regular and context-free languages, this is decidable.

These are the kinds of rigorous questions you can answer with automata theory that would be difficult or impossible with purely statistical approaches.

2.5.5 Summary

This project gives you the chance to *deeply understand* automata theory by applying it by designing one to solve a real problem is entirely. Throughout the project, you will need to answer questions like:

- How do I choose my alphabet (what symbols represent fixations)?
- How many states do I need?
- What should the acceptance condition be?
- Is my language regular, context-free, or more complex?

Answering these questions will deepen your understanding of the theory in ways that solving textbook exercises never could. This is what research is about: taking theoretical tools and figuring out how to apply them to messy, real-world problems.

3 Research Project: Scope and Variations

All teams in this course will work on the same general problem: scanpath pattern recognition for ECG interpretation. However, you will approach this problem using different formal models and techniques. This means that while everyone is working toward the same goal, the paths to get there will be different.

3.1 Core Problem

Given a dataset of recorded scanpaths from medical professionals reading 12-lead ECGs, you will design a formal automata-based model that can accomplish one or more of the following:

1. Represent systematic ECG reading strategies as formal languages
2. Classify scanpaths according to expertise level or diagnostic strategy type
3. Detect deviations from recommended systematic approaches
4. Predict next likely fixations given partial scanpath sequences
5. Evaluate scanpath quality or completeness

3.2 Choosing Your Formal Model

This is one of the most important decisions you will make for your project: which type of automaton or formal model will you use? There is no single "correct" answer. Different models have different strengths and weaknesses. Your choice should be justified based on:

- **Expressive power**: Can this model capture the patterns you want to recognize?
- **Computational complexity**: How efficiently can you recognize patterns with this model?

- **Learnability**: Can you construct this model from training data?
- **Interpretability**: Can you (and others) understand what your model is doing?

Below are some options to consider. You are not limited to these. If you have another idea, propose it!

3.2.1 Option 1: Finite State Automata (FSA) Approach

Model scanpaths as regular languages where states represent ECG regions or diagnostic phases, and transitions represent saccadic movements. This approach is suitable for:

- Simple sequential patterns without hierarchical structure
- Fast linear-time pattern matching
- Direct implementation as state machines
- Easy visualization and interpretation

Example application: Classify scanpaths as "systematic" vs. "non-systematic" by defining an FSA that accepts sequences following the recommended clinical reading order.

3.2.2 Pushdown Automata (PDA) Approach

Extend FSA with stack memory to model hierarchical or nested scanning behaviors. Suitable for:

- Capturing "zoom in → examine detail → return to overview" patterns
- Tracking context during multi-level analysis
- Modeling backtracking and revision behaviors

Example application: Model the pattern where clinicians examine overall rhythm (push context), focus on specific abnormality (nested state), then return to overview (pop context).

3.2.3 Context-Free Grammar (CFG) Approach

Define production rules that generate valid scanpath sequences. Suitable for:

- Compositional pattern descriptions
- Hierarchical decomposition of strategies
- Generative modeling of expert behavior
- Parse tree construction for scanpath analysis

Example application: Define grammar rules such as:

$$\begin{aligned}\text{Strategy} &\to \text{Overview DetailedScan Verification}\\ \text{DetailedScan} &\to \text{LeadScan} \mid \text{LeadScan DetailedScan}\\ \text{LeadScan} &\to \text{V1} \mid \text{V2} \mid \cdots \mid \text{aVF}\end{aligned}$$

3.2.4 Regular Expression Approach

Define pattern templates using regular expression syntax. Suitable for:

- Compact pattern specification
- Fast matching using existing regex engines
- Expressing optional, repeated, or alternative patterns

Example application: Define expert pattern as:
`(Rhythm)(Axis)(P-wave)(QRS)+(ST-segment)(T-wave)`
where `+` indicates one or more examinations of QRS complexes.

3.2.5 Probabilistic Automata / Markov Chain Approach

Extend deterministic automata with transition probabilities. Suitable for:

- Modeling variability in expert strategies
- Learning probabilistic models from data
- Predicting likely next fixations
- Comparing scanpath likelihood under different models

Example application: Learn transition probabilities $P(r_j|r_i)$ representing likelihood of fixating region r_j after r_i, then use these probabilities to evaluate how "expert-like" a given scanpath is.

3.2.6 Graph-Based Representation

Model scanpaths as paths in directed graphs where nodes represent ECG regions. Suitable for:

- Network analysis of fixation patterns
- Centrality measures to identify critical regions
- Path similarity metrics for scanpath comparison
- Community detection for strategy clustering

Example application: Construct scanpath graph and compute PageRank scores to identify most important ECG leads for diagnostic decision-making.

3.2.7 Other Approaches

Students may propose alternative formal models (e.g., Petri nets, timed automata, tree automata) with instructor approval. The proposal must justify why the chosen model is appropriate for scanpath representation and demonstrate clear advantages over simpler alternatives.

4 Project Requirements

4.1 Team Structure

- Projects must be completed in teams of 2–3 students
- All team members must contribute meaningfully to all aspects of the project
- A contribution statement must be included in the final deliverable describing each member's specific contributions

4.2 Required Deliverables

4.2.1 Technical Paper (Due: December 9–11, 2025)

A research paper of 4–6 pages formatted using the ACM/Springer (depending on the target conference) single-column template. The paper must include:

- **Abstract**: Concise summary of the problem, approach, and key results
- **Introduction**: Motivation and overview of the approach
- **Related Work**: Survey of relevant literature on scanpath analysis and automata applications
- **Formal Model**: Mathematical definition of the chosen automata representation including:
  - Formal notation and definitions
  - State space or alphabet description
  - Transition function or production rules
  - Acceptance conditions or recognition criteria
- **Methodology**: Description of algorithms, data structures, and implementation
- **Evaluation**: Experimental results including:
  - Description of datasets used
  - Evaluation metrics (accuracy, precision, recall, etc.)
  - Comparison with baseline approaches
  - Theoretical complexity analysis
- **Discussion**: Interpretation of results, limitations, and implications
- **Conclusion**: Summary and future work directions
- **References**: Properly formatted citations
- **Acknowledgments**: Disclosure of any AI/LLM tools used in the project

4.2.2 GitHub Repository

A public or private GitHub repository containing:

- **README.md**: Project overview, setup instructions, usage examples
- **Source code**: Well-organized, documented implementation
- **Data**: Sample scanpath datasets or instructions for obtaining data
- **Documentation**: API documentation, algorithm descriptions
- **Examples**: Demonstration scripts showing key functionality
- **Visualizations**: Diagrams of automata, sample outputs
- **Paper PDF**: Final version of the technical paper

The repository should follow software engineering best practices including:

- Meaningful commit messages
- Logical organization of code into modules/packages
- Consistent code style and formatting
- Version control throughout the development process

4.2.3 Oral Presentation (Pending time management)(December 9–11, 2025)

A 15-minute presentation including:

- Problem motivation and background (2 minutes)
- Formal model explanation with visual diagrams (4 minutes)
- Live demonstration or detailed walkthrough (5 minutes)
- Results and evaluation (3 minutes)
- Conclusions (1 minute)
- Questions and answers (5 minutes)

Presentations should use clear visualizations, avoid excessive text on slides, and engage the audience effectively.

4.2.4 Peer Reviews (Pending Time Management)

Each team will review the draft papers of two other teams (assigned by the instructor). Reviews should be constructive, specific, and approximately one page in length. The peer review process helps students develop critical evaluation skills and provides valuable feedback to authors before final submission.

Peer reviews are due December 8, 2025, one day before final presentations.

5 Example Dataset and Experimental Design

5.1 Dataset Requirements

For this project, you have two options for obtaining scanpath data:

5.1.1 Option 1: Open-Source Real Datasets (Preferred)

You are encouraged to search for existing open-source eye-tracking datasets related to medical image interpretation or similar visual search tasks (of even different medical images of the same problem). Real datasets provide authentic patterns and make your work more directly applicable. Successfully finding and using a real open-source dataset will earn you additional credit (see grading section).

Potential sources to explore:

- Academic data repositories (OpenNeuro, OSF, Zenodo)
- Eye-tracking research group websites
- Medical imaging competitions and challenges
- Publications with associated data releases

If you find a real dataset, even if it is not specifically about ECG reading (for example, radiograph interpretation, pathology slide examination), you can adapt your project to that domain with instructor approval.

5.1.2 Option 2: Synthesized Datasets

If you cannot find suitable open-source data, you may create a synthesized (simulated) dataset for your project using LLMs. While synthesized data does not capture the full complexity of real human behavior, it can still effectively demonstrate your automata-based models and algorithms.

To create a convincing synthesized dataset:

- Base your simulation on clinical guidelines for systematic ECG reading (as described in Section 2.4)
- Add realistic variability (experts do not follow identical patterns)
- Include both "expert" patterns (systematic, complete) and "novice" patterns (erratic, incomplete)
- Document your synthesis methodology clearly in your paper

Creating a well-designed synthesized dataset is legitimate research work and can be part of your contribution. However, real datasets will receive higher evaluation scores due to their authenticity.

5.2 Sample Scanpath Data Format

Students will work with scanpath data structured as temporal sequences of fixations. A sample data format (CSV) is:

```
participant_id,expertise,trial_id,fixation_id,x,y,duration_ms,ecg_region
P001,expert,T01,1,120,350,250,Lead-II-P-wave
P001,expert,T01,2,280,350,180,Lead-II-QRS
P001,expert,T01,3,420,350,210,Lead-II-T-wave
P001,expert,T01,4,120,490,195,Lead-III-QRS
...
```

Each row represents a single fixation with:

- Participant identifier and expertise level
- Trial/session identifier
- Fixation sequence number
- Screen coordinates (x, y)
- Fixation duration in milliseconds
- Semantic label for ECG region (e.g., Lead-II-P-wave)

5.3 Abstraction to Symbolic Sequences

For automata-based analysis, continuous scanpath data must be abstracted to symbolic sequences. This involves:

1. **Spatial discretization**: Map continuous coordinates to discrete ECG regions
2. **Temporal segmentation**: Define fixation vs. saccade boundaries
3. **Semantic labeling**: Assign meaningful symbols (e.g., P-wave, QRS, T-wave)
4. **Sequence construction**: Order fixations temporally to form input strings

Example abstraction:

$$\text{Raw scanpath:} \quad [(120, 350, 250), (280, 350, 180), (420, 350, 210), \ldots]$$
$$\text{Symbolic sequence:} \quad \langle P, Q, T, Q, R, S, P, \ldots \rangle$$

where symbols represent ECG features: P (P-wave), Q (QRS complex), T (T-wave), R (rhythm check), S (ST-segment).

5.4 Evaluation Metrics

Projects should evaluate their models using appropriate metrics:

- **Classification accuracy**: Percentage of scanpaths correctly classified (expert vs. novice)
- **Precision and recall**: For multi-class strategy recognition
- **Pattern matching efficiency**: Time complexity for recognizing scanpath patterns
- **Model compactness**: Number of states/rules required to represent patterns
- **Prediction accuracy**: For models that predict next fixations
- **Completeness detection**: Ability to identify whether all recommended ECG regions were examined

6 Grading Criteria

The project constitutes 30% of the final course grade and might take some of the final exam percentage, distributed as follows (Pending modifications):

Component	**Weight**
Technical paper quality	20%
GitHub repository and implementation	15%
Oral presentation	10%
Theoretical rigor and correctness	15%
Peer review participation	10%
Dataset quality (real data +5%, synthesized +3%)	5%
Conference submission (mandatory)	25%
Total	**100%**

Table 1: Project grading distribution

6.1 Conference Submission Requirement (25%)

Conference submission is mandatory for this project. This is an essential part of your learning experience, exposing you to the academic publication process. The 25% allocated for conference submission is distributed as:

- **Timely submission to an approved venue (15%)**: You must submit your work to one of the three recommended conferences (CHI 2026, HCII 2026, ETRA 2026) or propose an alternative venue with instructor approval. The submission must be completed by the specified deadline.
- **Submission quality (10%)**: Your submission must meet the venue's formatting requirements, be well-written, and represent serious work worthy of review.

Finding alternative venues: If you identify a different conference or workshop that seems more appropriate for your specific work, you may propose it to the instructor. The venue must be:

- A peer-reviewed academic conference or workshop
- Have a submission deadline between December 2025 and March 2026
- Be relevant to your project topic (HCI, eye-tracking, medical AI, formal methods, etc.)

Discuss your alternative venue choice with the instructor before committing to it.

6.2 Detailed Rubrics

6.2.1 Technical Paper (20%)

- **Problem formulation (4%)**: Clear statement of research question and objectives
- **Formal model definition (7%)**: Mathematically rigorous automata specification
- **Technical correctness (6%)**: Sound theoretical analysis and valid algorithms
- **Writing quality (3%)**: Clear organization, proper grammar, appropriate academic style

6.2.2 GitHub Repository (15%)

- **Code quality (6%)**: Well-structured, documented, maintainable
- **Documentation (4%)**: Clear README and usage instructions
- **Reproducibility (3%)**: Sufficient information for others to replicate results
- **Version control (2%)**: Meaningful commits and proper Git usage

6.2.3 Oral Presentation (10%)

- **Content clarity (4%)**: Clear explanation of problem and approach
- **Visual aids (3%)**: Effective diagrams and demonstrations
- **Delivery (2%)**: Engaging presentation style, time management
- **Q&A (1%)**: Thoughtful responses to questions

6.2.4 Theoretical Rigor (15%)

- **Formal notation (3%)**: Correct use of mathematical symbols and definitions
- **Model design (7%)**: Appropriate choice of automata class for the problem
- **Analysis depth (5%)**: Complexity analysis, theoretical properties, proofs where applicable

6.2.5 Dataset Quality (5%)

- **Real open-source dataset (5%)**: Successfully identified and used authentic eye-tracking or visual search data
- **Synthesized dataset (3%)**: Well-designed simulated data with documented methodology
- **Dataset documentation (applies to both)**: Clear description of data format, variables, and preprocessing

7 Timeline and Milestones

All major milestones are scheduled on Mondays and Wednesdays to align with our course meeting times. Please mark these dates in your calendar:

Date	**Milestone**
Monday, November 18, 2025	Project Announcement
Wednesday, November 20, 2025	Team formation due Proposal feedback session
Monday, November 25, 2025	Project Checkin check-in meetings (Week 1)
Wednesday, November 27, 2025	
Monday, December 1, 2025	Rough Draft of Problem Statement, Problem Description
Wednesday, December 3, 2025	1st Draft of Paper due
Sunday, December 7, 2025	**Final deliverables due**

Table 2: Project timeline (all deadlines at 11:59 PM)

7.1 Proposal Requirements (Due Monday, December 18)

By November 18, your team must submit a 1-page proposal including:

- Team member names and contact information
- Chosen formal model (FSA, PDA, CFG, etc.) with brief justification (2-3 sentences explaining why this model is appropriate)
- Preliminary approach description (what will your automaton represent? what patterns will it recognize?)
- Dataset plan: Will you use real data (if so, from where?) or synthesized data (if so, how will you generate it?)
- Expected outcomes and success criteria (how will you know if your project succeeded?)
- Potential challenges you foresee and ideas for addressing them

The proposal is your commitment to a direction, but you can refine your approach as you work. The main purpose is to ensure you have thought through your project plan and received early feedback from the instructor.

8 Conference Submission (Mandatory)

Important: Submitting your work to an academic conference is a required component of this project, worth 25% of your grade. This requirement serves multiple pedagogical purposes:

- You will experience the complete research cycle, from idea to peer-reviewed publication
- You will learn how to format academic papers according to conference standards
- You will understand submission systems and academic deadlines
- You will gain confidence in presenting your work to the international research community
- If accepted, you will have a genuine publication on your academic record

Many of you will find this intimidating at first. That is normal! Remember: acceptance is not required for full credit. What matters is that you complete the work to a professional standard and submit it for review. Even if your paper is not accepted, you will have gained invaluable experience with the publication process.

8.1 Target Conferences

You must submit to one of the following three recommended conferences, or propose an alternative venue (see below).

8.1.1 CHI 2026 Posters (Recommended)

- **Conference**: ACM CHI Conference on Human Factors in Computing Systems
- **Location**: Barcelona, Spain
- **Dates**: April 13–17, 2026
- **Submission deadline**: January 22, 2026
- **Format**: 6-page extended abstract
- **Publication**: ACM Digital Library (non-archival, open access with no APC)
- **Website**: `https://chi2026.acm.org/authors/posters/`

CHI is the premier international conference on human-computer interaction. The Posters track welcomes "nascent research" and "provocations," making it ideal for undergraduate student work. Non-archival publication means students can later expand their work into full conference papers.

8.1.2 HCI International 2026 Posters

- **Conference**: HCI International 2026
- **Location**: Montreal, Canada
- **Dates**: July 28–August 2, 2026
- **Submission deadline**: Expected February 2026 (TBD)
- **Format**: 6–8 page short paper
- **Publication**: Springer CCIS series
- **Website**: `https://2026.hci.international/posters.html`

HCII is a large international conference covering diverse HCI topics. Poster papers are published in Springer's Communications in Computer and Information Science series.

8.1.3 ETRA 2026

- **Conference**: ACM Symposium on Eye Tracking Research & Applications
- **Location**: Marrakech, Morocco
- **Dates**: June 1–4, 2026
- **Website**: `https://etra.acm.org/2026/`

ETRA is the premier conference specifically focused on eye-tracking research. The proximity to Morocco makes this particularly attractive. Students should monitor the website for poster or demonstration track announcements with later deadlines than the full paper track.

8.2 Proposing Alternative Venues

If you discover a conference or workshop that seems particularly well-suited to your specific work, you may propose it as an alternative to the three recommended venues. To propose an alternative venue:

1. Identify the venue name, website, and submission deadline
2. Explain why this venue is more appropriate for your work than the recommended options
3. Verify that it is a legitimate peer-reviewed academic venue (not predatory)
4. Discuss with the instructor at least two weeks before the submission deadline

The instructor will help you evaluate whether the venue is appropriate and will approve suitable alternatives. Good alternative venues might include:

- Workshops at major conferences (CVPR, NeurIPS, AAAI, etc.)
- Domain-specific conferences (medical imaging, cognitive science, education technology)
- Regional HCI conferences
- Student research competitions at major venues

8.3 Submission Timeline

You will work on your conference submission in parallel with completing the course requirements:

- **December 7, 2025**: Final course deliverables due (paper, GitHub, presentation)
- **December 7 – January 22, 2026**: Polish your paper for CHI 2026 Posters submission
- **January 22, 2026**: CHI 2026 Posters deadline (if choosing this venue)
- **February 2026**: HCII 2026 expected deadline (if choosing this venue)

You will have approximately 6-7 weeks after the course ends to refine your work for conference submission. The instructor will be available for consultation during this period to review drafts and provide feedback.

9 Resources and Support

9.1 Software Tools

Students may use any programming language or software tools appropriate for their chosen approach. Recommended tools include:

- **Python**: Libraries such as NetworkX (graphs), Graphviz (visualization), NumPy, Pandas
- **JFLAP**: Interactive tool for designing and testing automata
- **Automata Tutor**: Educational platform for automata
- **LaTeX**: For typesetting the technical paper
- **Git/GitHub**: For version control and collaboration

9.2 Use of Large Language Models

Students are permitted and encouraged to use AI assistants (ChatGPT, Claude, GitHub Copilot, etc.) for:

- Debugging code
- Generating boilerplate code
- Improving writing clarity
- Brainstorming ideas
- Explaining technical concepts

However: Core intellectual contributions—problem formulation, automata design, theoretical analysis, and interpretation of results—must be the students' own original work.

Disclosure requirement: All use of AI tools must be disclosed in the paper's Acknowledgments section, specifying which tools were used and for what purposes.

9.3 Office Hours and Support

The instructor will provide dedicated office hours for project support:

- **November 18–20**: Proposal feedback sessions (by appointment)
- **November 25–December 2**: Weekly office hours for technical questions
- **December 3–6**: Final consultation hours before presentations

Students are encouraged to seek help early when encountering difficulties rather than waiting until deadlines approach.

10 Academic Integrity

All submitted work must represent the original intellectual contribution of the team members. Violations of academic integrity include:

- Copying code or text from other teams without attribution
- Submitting work primarily generated by AI without meaningful human contribution
- Plagiarizing published papers or online resources
- Falsifying experimental results or data

Proper citation of all sources (papers, code repositories, software tools) is required. When building on existing work, clearly distinguish your original contributions from prior art.

Violations will be handled according to university academic integrity policies and may result in failure of the project or the course.