CS 5000: F23: Theory of Computability
Assignment 1

Vladimir Kulyukin
Department of Computer Science
Utah State University

September 2, 2023

Learning Objectives

1. Finite Automata
2. Deterministic Finite Automata (DFA);
3. Non-deterministic Finite Automata (NFA);
4. Epsilon-Closures.

Problem 0: (0 points)

Review the lecture materials `CS5000_F23_Lecture01.pdf` and `CS5000_F23_Lecture02.pdf` in Canvas. You may also want to read the supplementary reading materials on DFA and NFA (`BrooksWeber_DFA.pdf` and `BrooksWeber_NFA.pdf`).

Problem 1 (1 point)

In mobile robotics, the line following problem is to program a robot to follow a given line course. That line course may have the beginning and the end or may be a closed polygon contour. Read `RobotLineAndBlobFollowing.pdf` (included in the zip) for some background on the Junun robot hardware's states and signals, their correspondence to FA states and transitions, and the line following problem.

If you're a text-based learner, `RobotLineAndBlobFollowing.pdf` will suffice. If you're a visual learner, you can watch this video. This is a video from a previous version of my CS5000 when we held a line following competition h on the 4th floor of Old Main. The students designed FA line following controls for the Junun robots.

Design an FA to have a Junun robot follow a black line on the white background. You can design either a DFA or an NFA. Specify your FA in a graph that clearly identifies the start state, each transition, and the final states (if there are any). You can use your favorite graph editing tool to color the final states.

When I said in class that 99% of robotic controls are FA, I wasn't kidding. FA controls are not sensor- or platform-specific. This video shows a different robotic platform (PiCar) following a white line course with computer vision on the 4th floor of Old Main. This video is from a previous final project submission by a student from my CS5600/6600: Intelligent Systems. She designed and trained a neural network to classify frames from the robot's camera into 3 classes (`LINE_ON_LEFT`, `LINE_IN_MIDDLE`, and `LINE_ON_RIGHT`), which enabled the robot to follow the line. While the input processing software is much more sophisticated than in the Junun robot, the line following control is nothing but an FA. This is a great example to drive home the point that raw input processing (i.e., translating raw sensory data into symbols) is frequently much more challenging than top-level control.

Problem 2 (1 point)

Another common problem in mobile robotics is blob following. The objective is to program a robot to follow a blob in front of it. A blob is a generic term for an object identifiable through the robot's sensors (e.g., a camera, sonars, IR sensors, etc.).

Read `RobotLineAndBlobFollowing.pdf` for some background on the blob following problem and the Junun robot's front sonars.

Again, if you're a visual learner, you can watch this video where a Junun robot follows a small cardboard box in my hand with its front sonars.

Design an FA to have a robot (like the Junun in the video) follow a blob with its two front sonars. You can design either a DFA or an NFA. Specify your FA in a graph that clearly identifies the start state, each transition, and the final states (if there are any).

Problem 3 (1 point)

Planning is a branch of AI that studies computational models of problem solving. Given a problem, a planner returns, if possible, a sequence ot steps (executable on a specific hardware/software platform) to solve that problem. For example, given a state of the world with the three blocks A, B, and C on the table, a planner returns a sequence of actions for the robotic camera-arm unit to build a 3-block tower where A is on B, B is on C, and C is on the table.

The Sussman Anomaly is a famous AI planning problem named after its inventor Dr. Gerald Sussman. Dr. Sussman used this problem to illustrate weaknesses of linear planning algorithms (i.e., algorithms that solve problems one step at a time). Read about the Sussman anomaly at this Wiki site and design a DFA for the camera-arm unit that solves it. You do not have to be too formal when you specify your DFA. A clear graph drawing where all states and transitions are specified is perfectly acceptable. You may assume that your camera-arm unit has the same physical and perceptual capabilities as the camera-arm unit we discussed in Lecture 2.

Problem 4 (1 point)

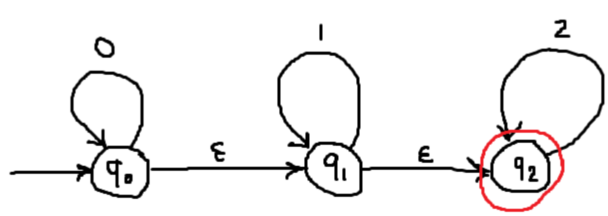

Figure 1: NFA with ϵ-transitions.

Consider the NFA M whose graph is given in Figure 1 and whose transition function δ is specified in Table 1. The symbol $\emptyset$ denotes the empty set, i.e., the set with no elements.

Table 1: Transition function δ for the NFA in Figure 1; the symbol $\emptyset$ denotes the empty set

states/inputs	0	1	2	ϵ
q_0	$\{q_0\}$	$\emptyset$	$\emptyset$	$\{q_1\}$
q_1	$\emptyset$	$\{q_1\}$	$\emptyset$	$\{q_2\}$
q_2	$\emptyset$	$\emptyset$	$\{q_2\}$	$\emptyset$

Use the ϵ-closure (See Slides 52–57 in CS500_F23_Lecture02.pdf) to construct an NFA M' that has no ϵ-transitions and is equivalent to M (i.e., $L(M') = L(M)$). Specify M' both with a graph similar to the one in Figure 1 and the table for its transition function δ' similar to Table 1. I hope that this problem will help you develop an intuition that from an NFA with ϵ-transitions one can construct an equivalent NFA, i.e., the NFA accepting the same language. that doesn't have any ϵ-transitions.

What To Submit

Use your favorite mathematical editor/presentation preparation software to type your solutions and make/draw graphs, convert your document to PDF, and submit it in Canvas as `CS5000_F23_HW01.pdf`. Since I'm 100% Linux, I use GNU Emacs and PDFLatex for LaTeX ediing and compilation, and GIMP for image editing. I heard lots of great stuff from my students and colleagues about Overleaf, an online LaTeX editing platform, but have never used it myself. Please do not submit smartphone snapshots of your handwritten solutions. If I cannot read them, I will have to ask you to re-submit. Remember to write your name and A-number at the beginning of your document. You can simply make it part of the title.

Have fun with the FA robot control designs!