Recitation - Week 2

CSE 355: Theory of Computation

Problem Sets (Module 1)

Problem 1

Prove that if $a, b \in \mathbb{R}$ are real numbers and $a < b < 0$, then $a^2 > b^2$.

Problem 2

Prove that $\sqrt{3}$ is irrational.

Problem 3

Prove that if $n \in \mathbb{N}$ is a natural number and $n \geq 5$, then $2^n > n^2$.

Solutions

1. Direct Proof

2. Proof by Contradiction

3. Proof by Induction

Installation

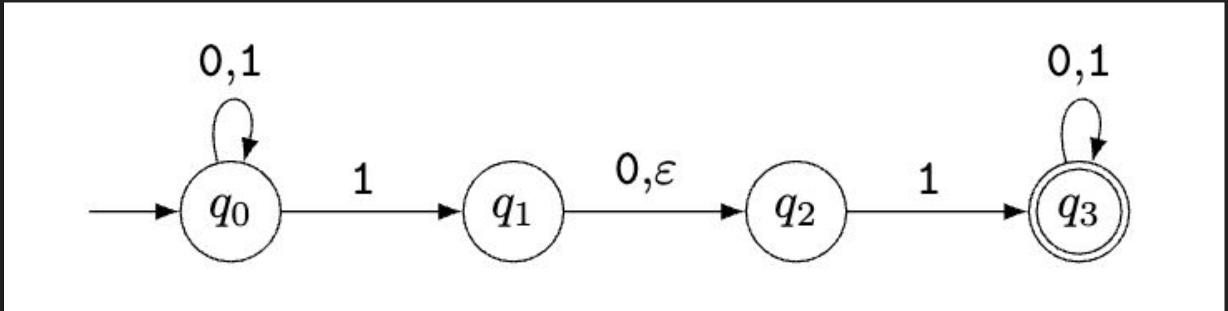
1) Open CMD and run “python3 –version” or “python –version” to check if the installed version is >3.10. If it is already installed, you may skip the installation.
2) Go to https://www.python.org/downloads/ and click on Download python 3.12.5 for Windows. You can choose your version of choice by scrolling down but it should >3.10
3) Depending on the command that you used to check the version earlier, use the command “pip install cse355-machine-design” for python or “pip3 install cse-355-machine-design” for python3 in your machine’s terminal or CMD.

Machine Design (Module 3)

Let's look at a demo where we use the installed package & python to create a NFA (Nondeterministic Finite Automata) that recognizes the language L.

<u>Example:</u>

L = {w ∈ {0, 1}* | w contains 101 or 11 as a substring}



NFA in a Nutshell

An NFA (Nondeterministic Finite Automaton) is a computational machine represented as a graph where nodes are states and edges are transitions based on input symbols. Unlike a DFA, an NFA can have multiple transitions for the same input or even transitions without consuming any input (epsilon transitions). It accepts a string if any path leads to an accepting state.

Key Components:

- Q : Set of States - All possible states.

- ∑ : Alphabet Set - Symbols the NFA can read.

- δ : Transition Function - Defines transitions, allowing multiple possibilities for each input.

- q_0 : Start State - Initial state where processing begins.

- F : Final States - Accepting states; if the NFA reaches any of these after input processing, the string is accepted.

How It Works

- The NFA explores multiple paths simultaneously. It accepts the string if at least one path ends in a final state, providing flexibility and simplicity in recognizing languages compared to a DFA.
- Do not hesitate to ask questions if something is unclear but know that this is yet to be covered in class.

Example

For the language (L = {w ∈ {0, 1}∗ | w contains 101 or 11 as a substring}), the NFA will be defined as:

- Q = {q_0, q_1, q_2, q_3}
- ∑ = {“0”, “1”}
- δ =

Input/State	q_0	q_1	q_2	q_3
0	q_0	q_2	-	q_3
1	q_0, q_1	-	q_3	q_3
_(ε)	-	q_2	-	-

- q_0 = q_0
- F = {q_3}

Let’s construct this in Python

Open your favorite IDE and follow along as we reconstruct the NFA in python using the machine design package we installed.

1. Import necessary modules
    a. In our case, “from cse355-machine-design import NFA, registry”
2. Create a function that returns a NFA “object”
3. Evaluate and Submit

Evaluation and Submission

<u>Displaying State Diagram:</u> `example().display_state_diagram()`

<u>Answer Evaluation:</u> `example().evaluate("010110", 1)`

<u>Answer Submission:</u> `example().submit_as_answer(1)`

`registry.export_submissions()`

Congratulations!

You just created a NFA! If you had any problems going through the demo. You may look at the slides again for reference or contact me for help.

You may also look at this github repo for solution and expected result:

https://github.com/tanay-jaiman/CSE355_Recitations/tree/main/Module2

Thank You!!