CS330 Project Memo

Date: November 2022
From: Sovannratana Khek
To: ProfessorVirgil
Subject: Answers to project questions

Development process:

Firstly, I decided I'd use python for the assignment because it is able to create a simple GUI and it is easily executable even without making the .exe file that I had in the set up process. This left me with 2 tasks, find how to make a GUI with python (I knew it was possible, just didn't learn it yet) and make a python file into an executable (assumed it was possible because python is already a script). I ended up learning how to use tkinter which is built into python for GUI creation and found a library to make a .exe (if I didn't I'd have left it to run as a script). After making a simple GUI, I started thinking about the checking algorithm. Luckily, the state machine was straight forward in that one digit/input is either correct or not. Thus I implemented the state machine using an array of every digit with some pointer that would follow the state machine diagram I attached. For the testing algorithm I used the Password class I created and its functions alongside a RNG to just run checks until the lock unlocked.

Findings:

My estimating was way off, I underestimated randomness (probably would have helped if I remembered things from AP Stats). I originally ran a 10,000 iteration test, but my laptop fans went up to 5000+ rpm only on the heavy games I play. Thus I changed the test amount and found 70 iterations to be not as frightening (results are attached).

Finite Automata:

The program takes in a string of digits and changes from locked to unlocked while running thus it doesn't really have an "accepted input". If we work on the assumption that an accepted ending is one where the program ends in an unlocked state, then the accepted input string x ends in 80771 (unlocks system) with any combination of digits afterward that aren't 807714 (locks system).

let the set A = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }
let the set B = { all x's where x an infinite sequences of A } = a set of strings made up of elements in A of any length [0, infinity)
let the set C = { all x's where x is in B and doesn't have '80774' in it } = a set of strings from set B that don't have the combination '80774' in them
If y is a string accepted by the FA, then it is a language of the FA and y = a + '80771' + b where a is some element in set A and b is some element in set B.