

Prompt: A memo describing what you have done, how you have done it, and your findings. In the memo clearly state what is the language of the FA you have implemented. Also indicate what is the regular expression corresponding to that language. Attach the state transition diagram for the FA.

Code Overview:

This programming assignment was done on Python, specifically Python 3.6.

This assignment was tested in a linux environment.

This code takes in a string as an input. This is for exception handling reasons listed below.

Exception: Values that are greater than 9 or less than 0, this includes letters and symbols, are considered a valid incorrect input, this is handled by resetting the FSM.

Part 1:

Variables:

Code = "59823" # Our code Student ID: A204(59823)

LockStatus = True # current status of lock

State = 0 # Current state in FSM

Methods:

LockCheck(key, LockStatus)

Parameters:

(int) Key → the unlock or lock value (1 is unlock || 4 is lock)

(Boolean) LockStatus → the current status of the door (i.e. locked or unlocked)

Return Value:

(Boolean) LockStatus → returns the updated status of the door (i.e. locked or unlocked)

getDigit(number, n)

Parameters:

(int) number → The access code

(int) n → the current State of the FSM

Return Value:

(int) number // 10**n % 10 → returns the needed value of the code, based on the current state

Part 2:

Variables:

seconds = 0 #

Code = random.randint(0,99999) #random value between 0 and 99999 inclusive

LockStatus = True # current status of lock

State = numDigits(Code) - 1 # final state in FSM

currState = State #Current State

Methods:

LockCheck(key, LockStatus)

Parameters:

(int) Key → the unlock or lock value (1 is unlock || 4 is lock)

(Boolean) LockStatus → the current status of the door (i.e. locked or unlocked)

Return Value:

(Boolean) LockStatus → returns the updated status of the door (i.e. locked or unlocked)

def getDigit(number, n)

Parameters:

(String) number → The access code, this is in string to fix corner case where code begins with 0

(int) n → the current State of the FSM

Return Value:

(String) number[n] → returns the needed value of the code, based on the current state

numDigits(num)

Parameters:

(int) num → this is our access code

Return Value:

(int) count → this represents the length of numbers, to calculate the number of states

DoorCheck(door)

Parameters:

(Boolean) door → Holds the value for whether the door is currently unlocked or not. Uses this to calculate how many seconds it took to crack the code.

Return Value:

(Boolean) door, whether it is true or false

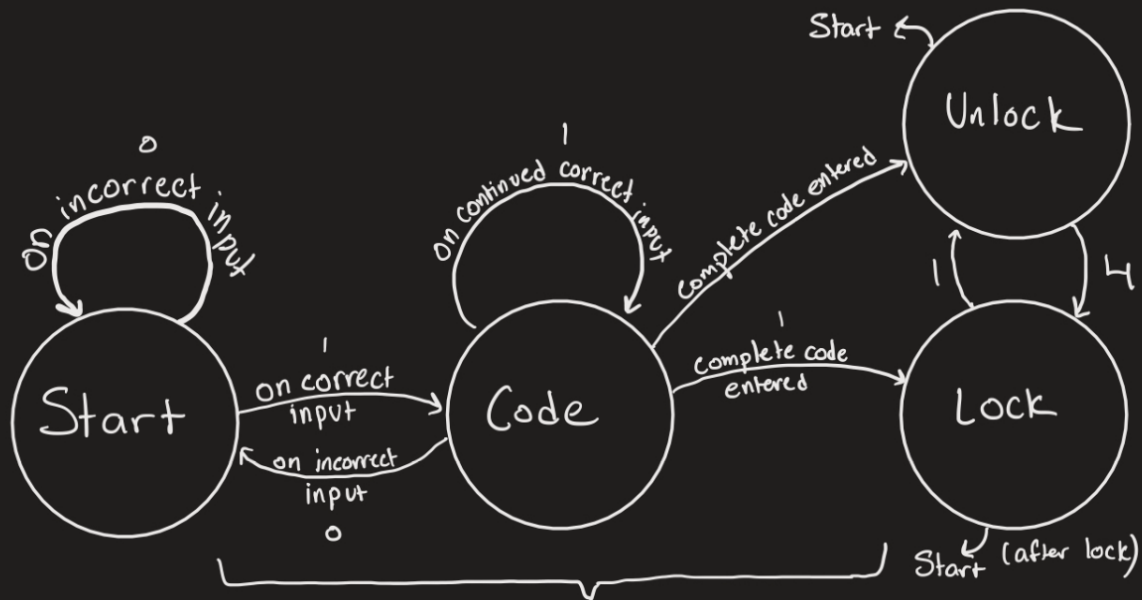
Time to break code: Values are from running part 2 in a 1000 count loop

Average: 245,342 seconds

Minimum: 5 seconds (Best case)

Maximum: 843,423 seconds

State Transition Diagram:



this is a mesh of these states, since length of code is variable, it's n .

1 = correct input
0 = incorrect input

