CS232 Project1
sp2015
streller

PROBLEM STATEMENT:

Given a four button, five character combination lock; write a program which will simulate the behavior of the lock when five characters are entered. The actions are unlock (if correct sequence entered) and an alarm (if the incorrect sequence is entered)/ A table ADT must be used to store the transition table and action table of the finite state machine (FSN) which models the behavior of the lock. The correct combination will be given in a file

PROBLEM BACKGROUND :

Some of the best known of all table-driven algorithms are based on a model known as a finite state machine (FSM). A finite state machine has five parts:

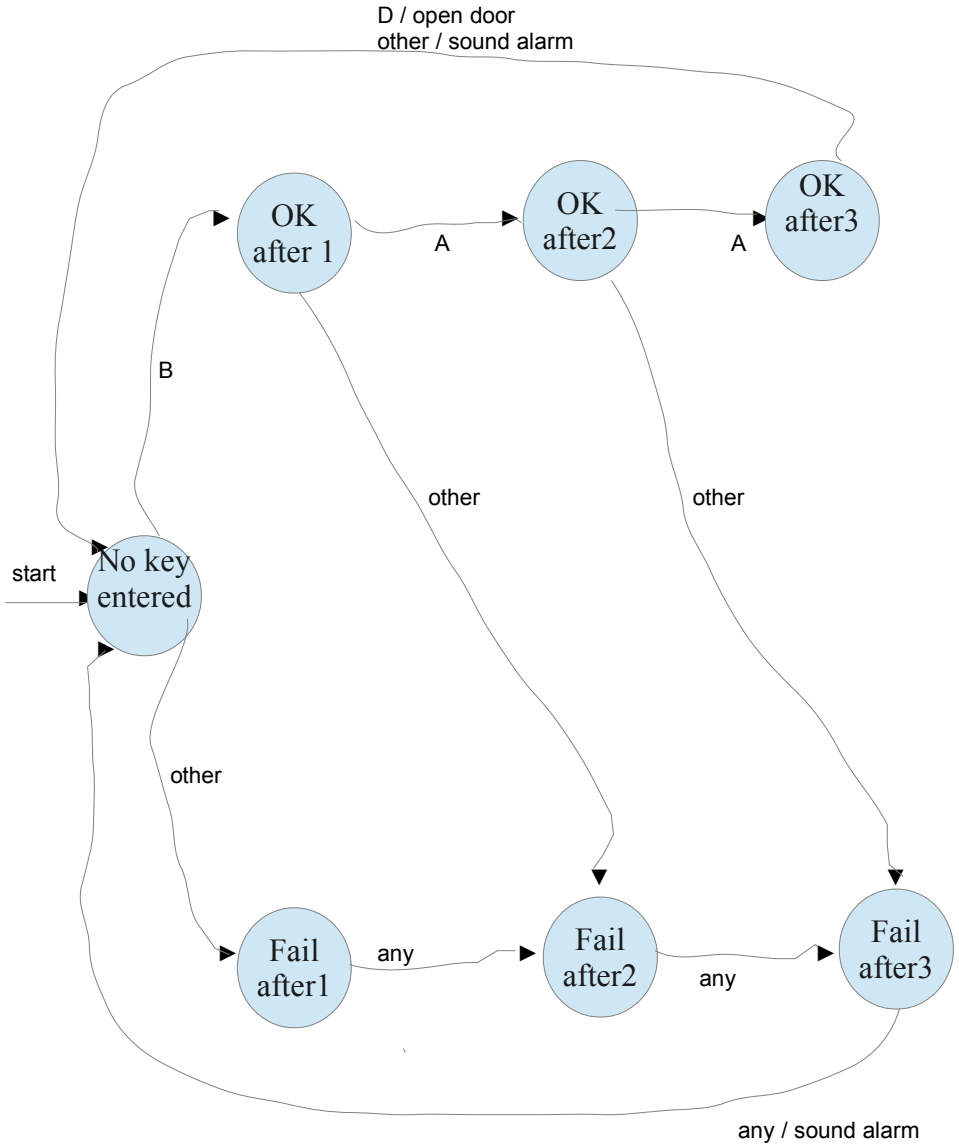
- A set of states
- A start state (which is an element of the set of states)
- A set of potential conditions (and/or events)
- A transition function that defines how to progress from one state to the next
- An action function that associates an action with each transition

State machines are typically pictured in state diagram form as a group of bubbles (one for each state) and arcs (one for each transition and associated action) The transition arrow is labeled with a condition or event (or both) under which such a transition occurs. This condition or event is the first part of any transition label (to the left of a forward slash, "/", symbol).
Many state machines can be directly translated into highly efficient table-driven algorithms. Such algorithms are based on the insight that transitions and actions can be characterized as two tables. One of these tables, often called the transition table, stores the transition function of the state machine. The other table, sometimes called the action table, stores the action associated with all transitions. The transition table and the action table have an identical number of rows and columns. Each table has a separate row for each state and a separate column for each possible event or condition from the state machine. The entries in the transition table indicate the state following the transition, sometimes called the next state.
The action table also contains entries that are appropriate for the outgoing transition from a state and an event or condition. The entries in the action table are actions.

The lock :

one of the five characters A – E will be used for each button

Finite State Machine for “BADD” combination



In other words, the entry in the OK After 1 row and the column labeled B indicates that from state OK After 1 a B event (pressing the B key) should make a transition into the Fail After 2 state. The action table also contains entries that are appropriate for the outgoing transition from a state and an event or condition. The entries in the action table are actions.

The action table for the above contains many empty entries to represent transitions that have no associated action.

CODE:

A table is an unordered group of “records”, each of which has a designated field called the key field Records within the table are identified by the value of the key field. For the operations provided by the table see pair.h and table.h.. The table is an appropriate ADT for information retrieval systems in which individual records must be accessed frequently (and quickly) but the entire collection of records must never be processed sequentially.
Complete the table implementation
use table to store the transition table
use table to store action table
the lock will always be the characters A , B , C , D , E.
the combination will always be a four character sequence
Must use accompanying files – with no additions/changes
See lab1 for specs on file layout, etc

DELIVERABLES:

hard :In a bound folder

1. documented source code
2. user manual.
3. Programmer manual(s) (one for each class also)

soft: in a zipped file, called CS232_P1_yourLastName,
sent to streller@ecc.edu with the subject CS232_P1

1. all source code
2. release version executable

Due Date : 9:00am 27 February 2015

Weekly status reports are to be electronically submitted by 6am on
13 February 2015
20 February 2015
The *13 February status report* must ALSO INCLUDE a detailed analysis/comparison of the table provided and the one you created for lab1. What are the differences, how far off was your table compared to the provided table. Your analysis will be part of your lab 1 grade.

These zipped updates must include source code and a pdf file explaining the status/progress of the project(what has been completed, what needs to be done, etc). The first status report is also to include an analysis of your (Lab1) table and the provided one for this project.
Send to streller@ecc.edu with the subject cs132_P1_UPDATE_yourLastName