User Manual

This program takes in a button character combination that will simulate the behavior of the lock when characters are entered. The actions are unlock ( if correct sequence entered ) and an alarm ( if the incorrect sequence is entered ). A table ADT is used to store the transition table and action table which models the

behavior of the lock.

1. Executing Program

Turn on computer and go to program execution. Run the executable and enter in the files of transition and action tables. The program will then prompt the user to enter in a letter from A-E four times and if the correct sequence of letters are entered then the lock will unlock if not then it will send an alarm.

1. Input Requirements

Valid file names must be entered and valid letters (A-E) must be entered doesn’t matter if capital or lowercase. The prompt "enter a character from A, B, C, D, E:’ then must enter any input for the letter four times. No other letters will work it will send you an “incorrect input try again” message.

1. Output

The output will either be unlock if the correct sequence of letters is entered by the user or an alarm will output if incorrect input is entered.

Programmer Manual

1. Problem description

The program requires a table ADT to place the values of a pair of values: ie. The transition table takes in the letter and state and outputs into a separate array a unique place for the combination. The action table read in also is implemented at the last read of the table to identify if the user has entered a combo that will unlock or alarm the program. The key element to this program is if the correct sequence of input letters from the user will unlock or alarm the lock.

1. Data Types and Classes

Variables and Functions:

typedef Key key\_type – the Key being the data type used to hold what will be in the table.

Int tableSize- size of table

Pair<key\_type, T> \*the\_table - declaring a table of pairs holding two values the Key and T

Int(\*Mapping)(Key k) – function that maps keys to an array index, key to address mapping the idea is that Mapping will act on a given key in such a way as to return the relative position in the sequence a which we expect to find the key.

Bool insert( Pair<Key,T> kvpair)- return true if item could be added and false if item was not added to the table.

Bool remove( const Key aKey)-erase the key/value pair with specified key from the table and return if successful removed item is replaced with default values for Key and T

Bool isIn(const Key& key) const- if it is in the table the key entered return true and false if not

1. High Level Program Solution

Main Program

Print the prompt for input and enter the transition table file name and open input file. Run through the file and store it into a table in my code twj is the first table input using the mapping function written under the main program. The mapping function runs through the possible states it could be in and if it matches the state in the for loop then it will execute the formula math = 5\*j+int(p.first) (the character portion of the pair) -65

And then jump out of the for loop. This formula is used to place the pair output into a new array of values to set up what a certain pair will map its answer to.

After the user inputs the transition file then they will need to enter the action table and follows the same functionality as the mapping in previous table with transition table. The difference in this reading in is that there is a lot of white space in the file and needs to be signaled when the input is either and ok or fa state to map to when the alarm or unlock will execute. So using a peek function to find when o and f appear then it will allow to add the action values into the table.

Next after entering the files the user will enter the meat of the program and be asked to enter the input of letters from A-E one at a time four times. In the program it will read in and lookup whether the letter is in the spot to either ok or fail the next combination. In the program this will execute three times and on the fourth attempt it will go to the action table to print out whether the combination entered will ether unlock or alarm the program.

After going through program it will ask to run program again and if you want you can enter yes and go through it again entering in the transition table, action table, and inputs for the lock to see whether unlock or alarm with entries otherwise no and done with program.