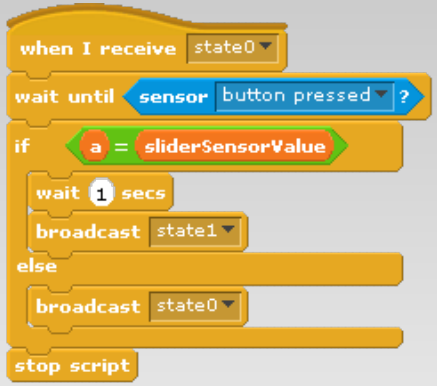
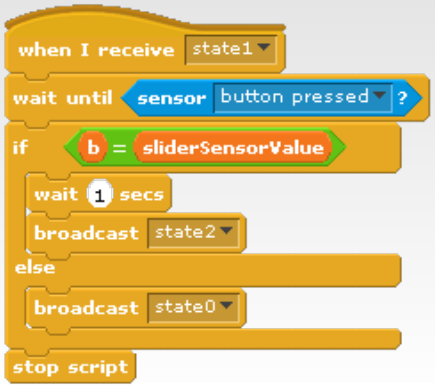
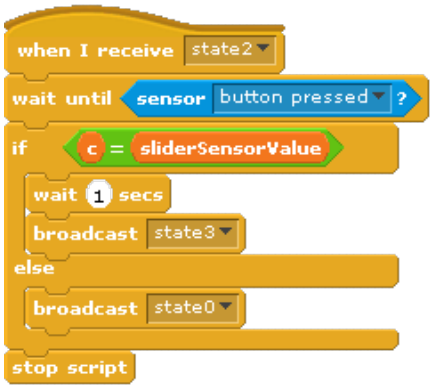
Finite State Machine (FSM)

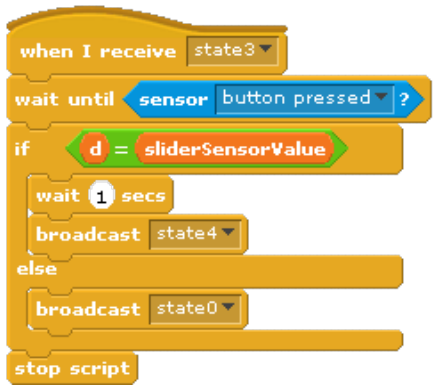
by a safe lock design example

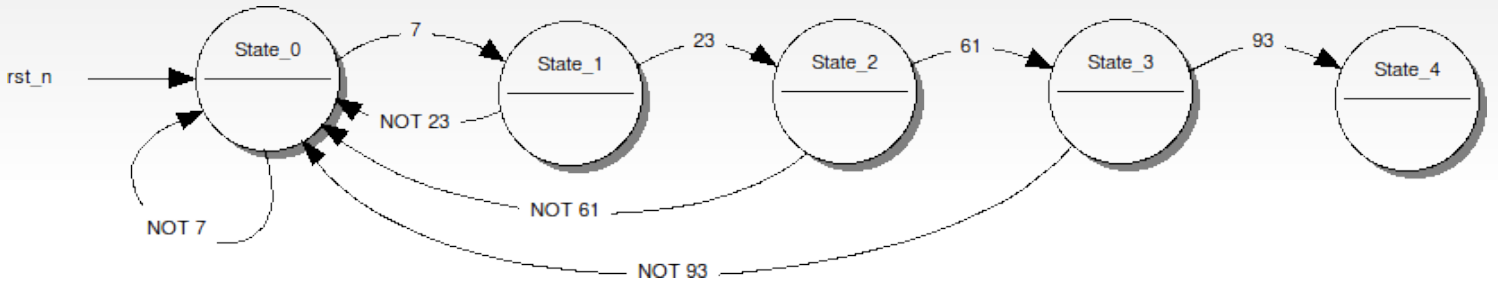
FSMs are used to model systems which have a limited number of ***states***, ***transitions*** between these states and the ***actions*** taken as a result.











- The numbers on the arrows represent what the system senses as an ***input***, thus a ***transition*** from one ***state*** to another can take place. When there is no entry, the system keeps the ***current state***.
- The example below represents an acceptor FSM, parsing the secret ***combination*** which, in our example, would un-lock a safe: 7, 23, 61, and 93. In case an unexpected number is entered, the system returns to the first state: State_0. The safe is unlocked, that is, the system is in the final state (State_4), only if the ***correct numbers*** are entered in the ***correct order***.
- On the right hand side, ***a possible FSM implementation*** in scratch environment is given.