A Combination Lock

Bart Massey
2016-02-25

We want a combination lock that will accept a code consisting of three numbers in a specified range and then allow the unlock button to unlock the door. Any error should cause the lock to reset.

$$CODE == 0 \mathinner{..} 35$$

We will treat the combination as fixed.

$$combination : \mathrm{seq}\, CODE$$

$$\# combination = 3$$

The last turn direction of the lock will be important. We can either have turned left (counter-clockwise) or right (clockwise).

$$DIRECTION ::= left \mid right$$

Turn

$$code : CODE$$
$$direction : DIRECTION$$

We will keep track of the last position and direction of the dial. When the device is turned on, we don't know where the dial was.

$$HISTORY ::= unsetted \mid setted\langle\langle Turn \rangle\rangle$$

The state of the lock consists of the current position of the dial, the number of digits solved so far, and the last direction turned.

Lock

$$solved : 0 \mathinner{..} \# combination$$
$$from : HISTORY$$

The starting state of the lock has no solved digits, and has not been turned yet. [We will use this schema primed, so we do not prime it here.]

Init

$$Lock$$

$$solved = 0$$
$$from = unsetted$$

Our initial turn should be to the left. A right turn resets the lock.

StartLeft

$$\Delta Lock$$
$$turn? : Turn$$

$$from = unsetted$$
$$solved' = 0$$
$$(turn?.direction = left \land from' = setted(turn?)) \lor$$
$$(turn?.direction = right \land from' = unsetted)$$

The normal operation is to accept a turn in the opposite of the previous direction to a position. If the position and direction are the correct next code in the combination, the user has solved the code and moves to the next.

$$
\begin{array}{|l}
\hline
\textit{SuccessfulTurn} \\
\Delta Lock \\
turn? : Turn \\
\hline
\forall f : Turn \mid setted(f) = from \bullet \\
\quad (turn?.direction \neq f.direction) \land \\
\quad (f.code = combination(solved + 1)) \\
solved' = solved + 1 \\
from' = setted(turn?) \\
\hline
\end{array}
$$

If the user keeps spinning in the direction they have been previously spinning, we assume that they are not done entering this number yet, and we ignore the spin. We allow the user to spin multiple full turns before stopping on the correct number.

$$
\begin{array}{|l}
\hline
\textit{PartialTurn} \\
\Delta Lock \\
turn? : Turn \\
\hline
\forall f : Turn \mid setted(f) = from \bullet \\
\quad turn?.direction = f.direction \\
solved' = solved \\
from' = setted(turn?) \\
\hline
\end{array}
$$

If the user turns in the opposite direction from the wrong position, we reset the lock.

$$
\begin{array}{|l}
\hline
\textit{WrongTurn} \\
\Delta Lock \\
Init \\
turn? : Turn \\
\hline
\forall f : Turn \mid setted(f) = from \bullet \\
\quad (turn?.direction \neq f.direction) \land \\
\quad (turn?.code \neq combination(solved + 1)) \\
\hline
\end{array}
$$

If we finish the last digit we are done.

$$
\begin{array}{|l}
\hline
\textit{Done} \\
Lock \\
\hline
solved = \# combination \\
\hline
\end{array}
$$

The overall behavior of the lock is to start reset, and to become more solved or reset until all the digits have been solved.

$$CombinationLock == Init' \fatsemi StartLeft \lor PartialTurn \lor SuccessfulTurn \lor WrongTurn \fatsemi Done$$