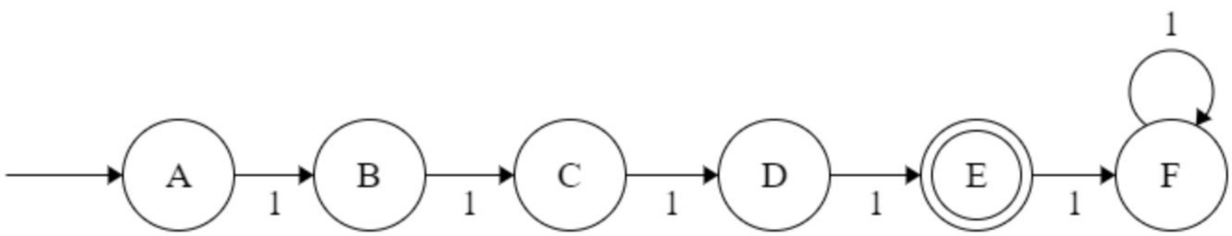
Names: Omar Abdelmotaleb and Benjamin Singleton
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Omar Abdelmotaleb
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Benjamin Singleton

Problem 1. (15 points)

a. Construct an FSA with 6 states to recognize $L_4 = \{1111\}$. Can you reduce the number of states below 6? (Hint: recall a basic property of directed graphs from CS 135!)
b. Use your argument to prove that, for all $k \geq 3$ there is a language that can be accepted by a k-state FSA that cannot be recognized by any FSA with fewer states.

a)



No, the number of states can not be reduced below 6. To recognize the string 1111, four transitions between five states are needed, strung together in a linear path. After the accept state, one additional transition and non-accept state are needed so that the FSA doesn't accept any strings with more than four 1s. Without the additional state at the end, strings longer than four 1s would have nowhere to go and terminate at the accept state.

b) For a k-state FSA, there is a language, whose alphabet consists of only one symbol, that can't be recognized by any FSA with fewer states. From the start state, the states must be arranged in a linear fashion, with the accept state k-1 from the end. Accepted strings will be k-2 in length and the final state will not be an accept state so that any strings greater than k-2 aren't accepted. The additional state ensures that strings greater than k-2 aren't accepted.

Names: Omar Abdelmotaleb and Benjamin Singleton
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Omar Abdelmotaleb
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Benjamin Singleton

Problem 2. (15 points) For any string $w = w_1 w_2 \cdots w_n$, the *reverse* of w, written w^R, is the string w in reverse order, $w_n \cdots w_2 w_1$. For any language A, let $A^R = \{w^R | w \in A\}$. Show that if A is regular, so is A^R.

Proof by construction:
A language is regular if there is a FSA that recognizes. If language A is recognized, to create an FSA that recognizes A^R , first take the FSA of A and reverse the direction of all transitions. Then, make the start state the end state. Create a new state which will now be the start start and uses epsilon transitions to reach all former start states. This new FSA, built by reversing the FSA of A, will recognize A^R and therefore if A is regular so is A^R .

Names: Omar Abdelmotaleb and Benjamin Singleton
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Omar Abdelmotaleb
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Benjamin Singleton

Problem 3. (20 points) In this problem you will design an FSA that checks if the sum of two numbers equals a third number. Each number is an arbitrarily long string of bits. At each step, the input to the FSA is a symbol that encodes 3 bits, one from each number. In other words, the alphabet of the FSA is:

$$\Sigma = \left\{ \begin{bmatrix}0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\1\\1\end{bmatrix}, \begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}1\\0\\1\end{bmatrix}, \begin{bmatrix}1\\1\\0\end{bmatrix}, \begin{bmatrix}1\\1\\1\end{bmatrix} \right\}$$

In the input string $\begin{bmatrix}0\\0\\1\end{bmatrix}\begin{bmatrix}1\\0\\1\end{bmatrix}$ the bits in the top row represent the number 01, the bits in the middle row represent 00 and the third row represents 11. In this case, since 01+00 $\neq$ 11, the FSA must reject the input. On the other hand, the input string $\begin{bmatrix}0\\0\\1\end{bmatrix}\begin{bmatrix}0\\1\\0\end{bmatrix}\begin{bmatrix}1\\1\\0\end{bmatrix}$ will be accepted by the FSA since $001 + 011 = 100$.

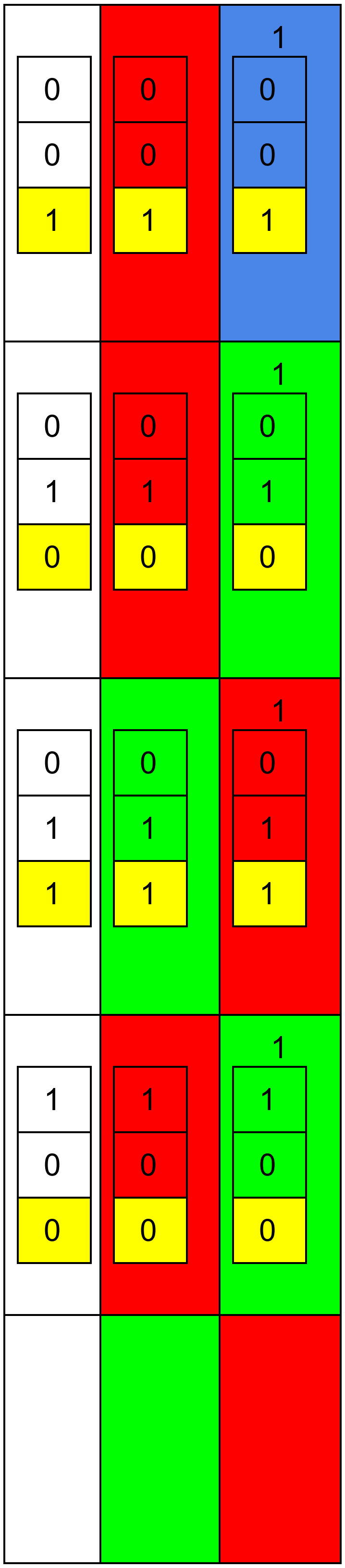
Formally, let $B = \{w \in \Sigma^*: \mathit{the\ bottom\ row\ of\ } w \mathit{\ equals\ the\ sum\ of\ the\ top\ two\ rows}\}$, where Σ^* represents all finite strings over the alphabet Σ. Your goal is to design an FSA for the language B.

To get started, first design a 2-state FSA that recognizes B^R – this should be straightforward because the input arrives least significant bits first and most significant bits at the end. Next, use the technique of Problem 2 to design the final FSA for B.

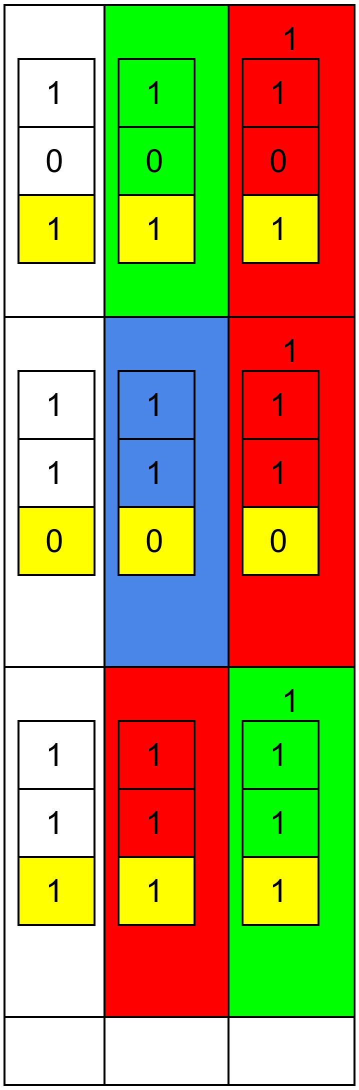
Transition table for B^R
Green = Stay in current state
Blue = Transition from Carry 0 to Carry 1 and vice versa
Red = Go to Sink state

	Carry 0	Carry 1
$\begin{bmatrix}0\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\0\end{bmatrix}$

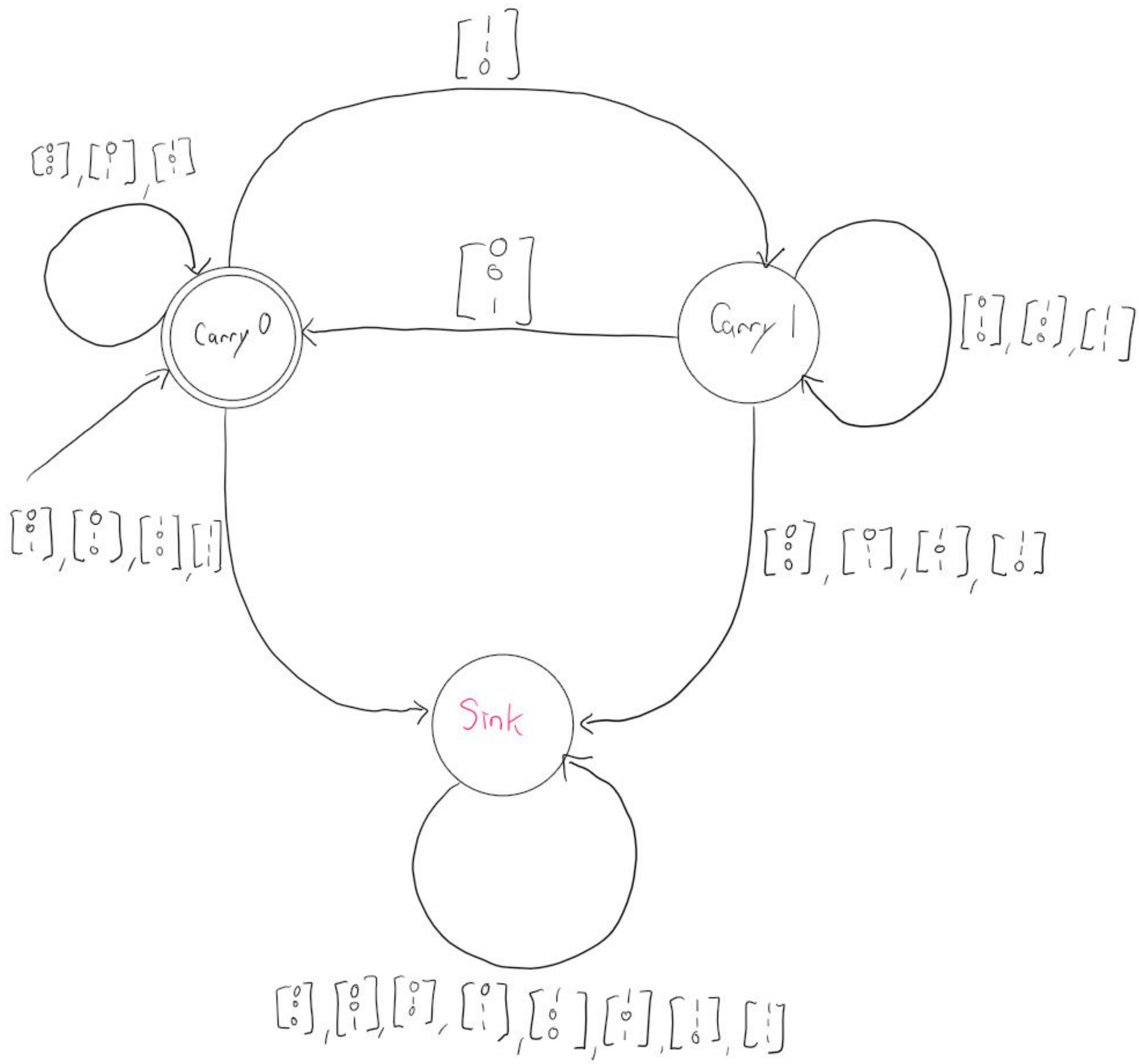
Names: Omar Abdelmotaleb and Benjamin Singleton
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Omar Abdelmotaleb
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Benjamin Singleton



Names: Omar Abdelmotaleb and Benjamin Singleton
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Omar Abdelmotaleb
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Benjamin Singleton

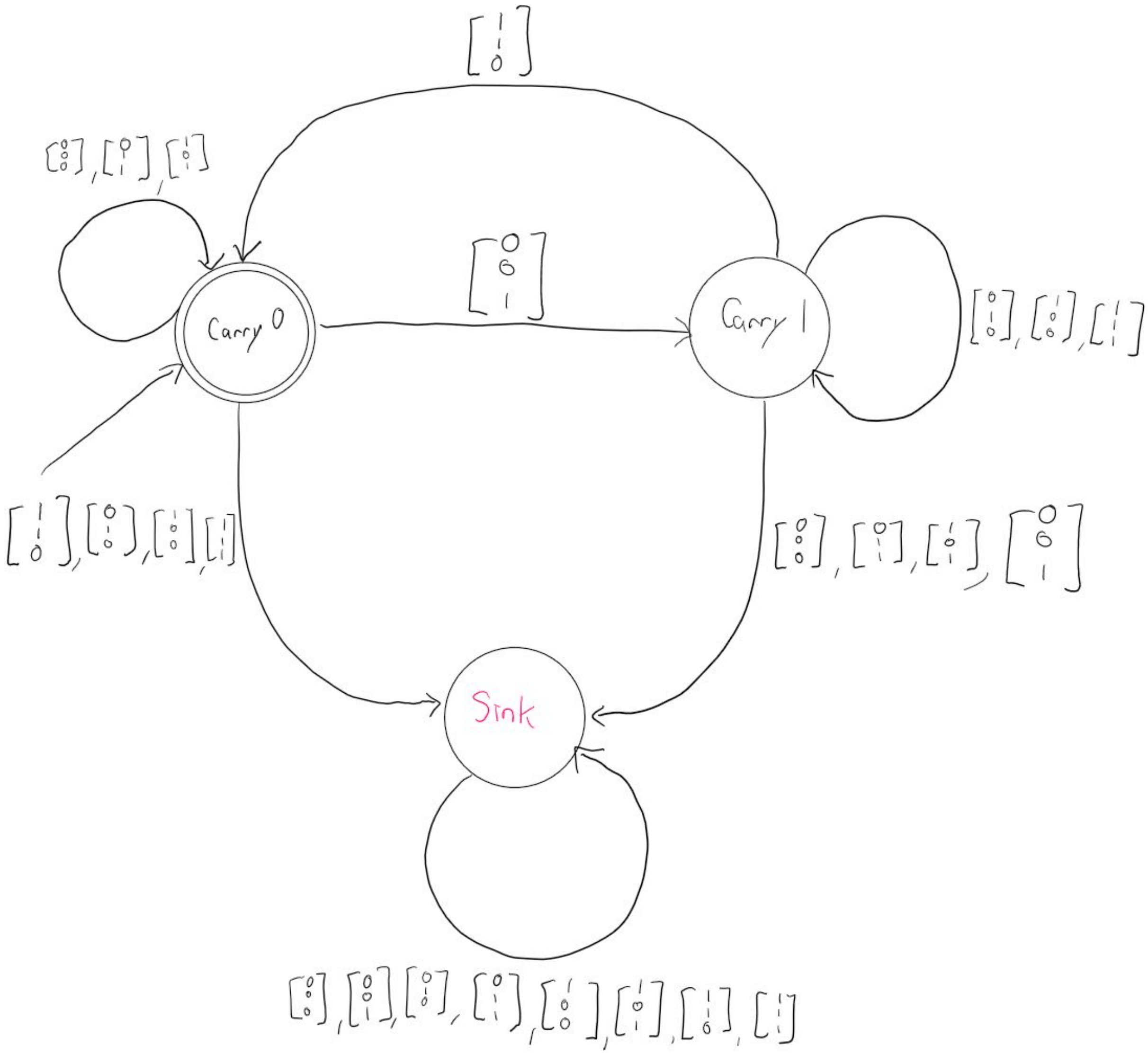


Names: Omar Abdelmotaleb and Benjamin Singleton
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Omar Abdelmotaleb
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Benjamin Singleton



3-state DFA for B^R.

Names: Omar Abdelmotaleb and Benjamin Singleton
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Omar Abdelmotaleb
Pledge: I pledge my honor that I have abided by the Stevens Honor System. -Benjamin Singleton



Final 3-state DFA for B.