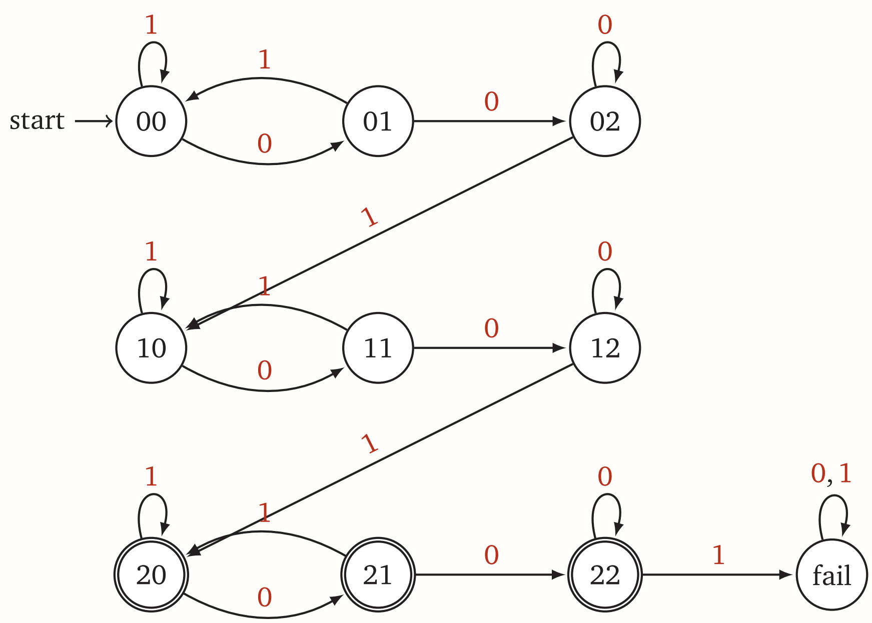
1. Let L be the set of all strings in $\{0,1\}^*$ that contain exactly two occurrences of the substring 001.

   (a) Describe a DFA that over the alphabet $\Sigma = \{0,1\}$ that accepts the language L. Argue that your machine accepts every string in L and nothing else, by explaining what each state in your DFA means. (You may either draw the DFA or describe it formally, but the states Q, the start state s, the accepting states A, and the transition function δ must be clearly specified.)

Solution: The following 10-state DFA accepts the language. Every state except fail is labeled with a pair of integers (i, j), where i is the number of times we have seen the substring 001 and j is the number of 0s (up to 2) we have just read. The machine enters the fail state when it sees 001 for the third time.



■

   (b) Give a regular expression for L, and briefly argue that why expression is correct.

Solution: $(1+01)^*000^*1(1+01)^*000^*1(1+01)^*0^*$

The subexpression $(1+01)^*000^*1$ describes the set of all strings that end with 001 but do not otherwise contain the substring 001. We use two copies of this subexpression to capture the two occurrences of 001. The last subexpression $(1+01)^*0^*$ describes all strings that do not contain the substring 001 at all.

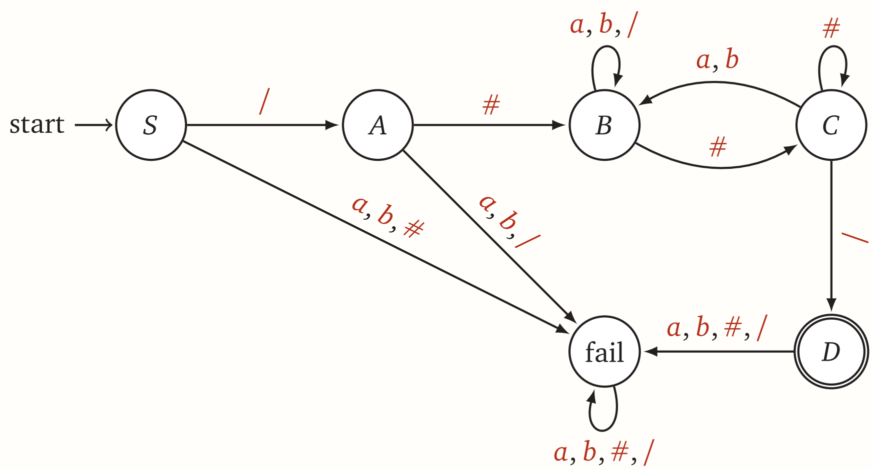
Alternatively: This is the regular expression computed by Han and Wood's algorithm, given the DFA in part (a) as input, if we eliminate states $x0$ and $x1$ before state $x0$, for each x.

■

2. In certain programming languages, comments appear between delimiters such as $/\#$ and $\#/$. Let C be the language of all valid delimited comment strings. A member of C must begin with $/\#$ and end with $\#/$ but have no intervening $\#/$. For simplicity, assume that the alphabet for C is $\Sigma = \{a, b, /, \#\}$.

   (a) Give a DFA that recognizes C.

Solution: The six state DFA accepts the language. Since any string must start with $/\#$, the A, B states enforce that. After the $/\#$, any $\#$ would change the state to C, which means if the next character is $/$, the string must end with no further characters. The transition from D to fail enforces this.



■

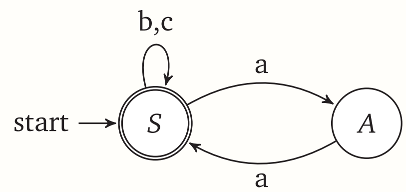
   (b) Give a regular expression that generates C.

Solution: $/\#(\#^*(a+b)+/)^*\#^+/$
The first $/\#$ enforces the string to start with $/\#$. The last part $\#^+/$ makes sure the string ends with $\#/$ and also represents any ≥ 1 number of $\#$s concatenated with $/$. This means the middle part should represent all the possible strings not ending with $\#$. The middle part is exactly implementing this. Since $\#/$ is forbidden in the middle, Every consecutive run of $\#$ must have an a or b after it. So all of the strings that does not end with $\#$ and does not allow $\#/$ is just $(\#^+(a+b)+a+b+/)^* = (\#^*(a+b)+/)^*$. ■

3. For each of the following languages, draw (or describe formally) an NFA that accepts them. Your automata should have a small number of states. Provide a short explanation of your solution, if needed.

   (a) All strings over $\{a, b, c\}^*$ in which every nonempty maximal substring of consecutive a's is of even length.

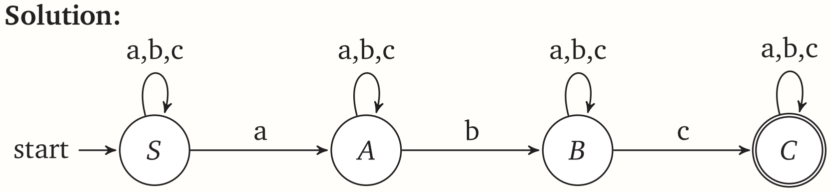
   Solution:

   ■

   (b) $\Sigma^* a \Sigma^* b \Sigma^* c \Sigma^*$

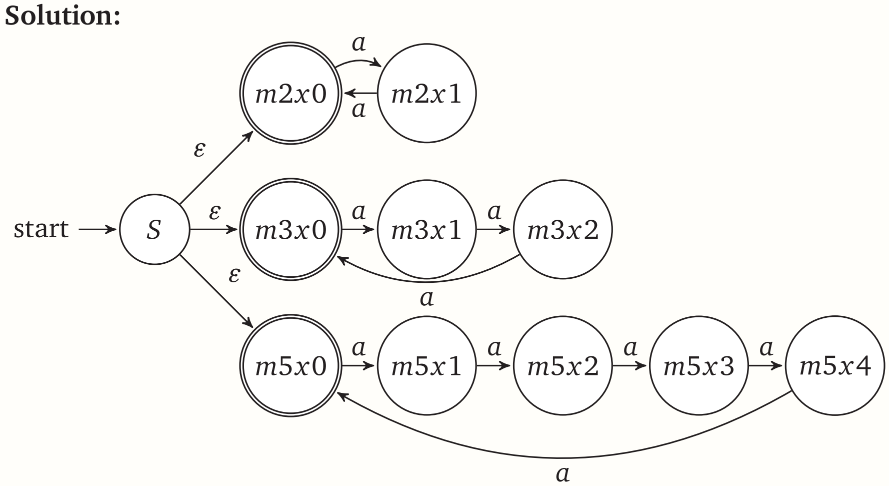
   Solution:

   ■

   (c) All strings in $w \in a^*$ of length that is divisible by at least one of the following numbers 2, 3, 5. For full credit your automata should have less than (say) 15 states.

   Solution:

   ■

   (d) All strings in $w \in a^*$ of length that is **NOT** divisible by at least one of the following numbers 2, 3, 5.

   Solution: Flipping the accept / reject state of the NFA in part (c) except the initial state would give the NFA for this problem. With $M = (\Sigma, Q, s, A, \delta)$ as the NFA for part (c), we define the NFA $M' = (\Sigma, Q', s', A', \delta')$ for part (d) as the following:

$$
\begin{aligned}
Q' &= Q \\
s' &= s \\
A' &= Q \setminus (A \cup \{s\}) \\
\delta'(q, \sigma) &= \delta(q, \sigma) \quad \forall q \in Q, \sigma \in \Sigma
\end{aligned}
$$

■

4. (a) For any string $w = w_1 w_2 \dots w_n$, the reverse of w, written w^R, is the string w in reverse order, $w_n \dots w_2 w_1$. For any language L, let $L^R = \{w^R \mid w \in L\}$. Show that if L is regular, so is L^R.

Solution: The first thing to note is that the given statement is equivalent to regular languages being closed under reversal.

Since L is regular, we know that a DFA $M = (Q, \Sigma, \delta, s, A)$ recognizes L. We construct an NFA $M^R = (Q^R, \Sigma, s^R, \delta^R, A^R)$ as follows:

$$Q^R = Q \uplus \{s^R\} \quad \text{(Here, } \uplus \text{ represents disjoint union.)}$$
$$\delta^R(s^R, \varepsilon) = A$$
$$\delta^R(s^R, a) = \varnothing \text{ for all } a \in \Sigma$$
$$\delta^R(q, \varepsilon) = \varnothing \text{ for all } q \in Q$$
$$\delta^R(q, a) = \{q' \in Q \mid \delta(q', a) = q\} \text{ for all } q \in Q, a \in \Sigma$$
$$A^R = \{s\}.$$

M^R effectively reverses the transitions in M. The sentinel start state s^R with outgoing ε-transitions to all accepting states allows the NFA to effectively start at every accepting state in M. (Note that, by definition, a DFA/NFA can only have one starting state.) Because M^R recognizes L^R, L^R is regular. ■

(b) Let

$$\Sigma_3 = \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \ldots, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right\}.$$

Σ_3 contains all size 3 columns of 0s and 1s. A string of symbols in Σ_3 gives three rows of 0s and 1s. Consider each row to be a binary number and let

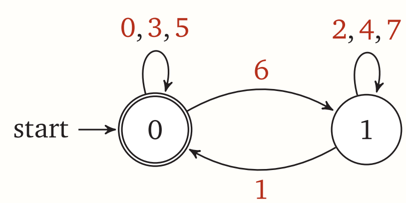
$$B = \{w \in \Sigma_3^* \mid \text{the bottom row of } w \text{ is the sum of the top 2 rows}\}.$$

For example,

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \in B, \quad \text{but} \quad \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \notin B.$$

Show that B is regular. (Hint: Working with B^R is easier. Use the result of part (a).)

Solution: One possible solution approach is to simulate long addition, where the carry bits are kept track of via the states in the constructed automaton. Let each symbol in Σ_3 be denoted by their corresponding decimal value as if reading top to bottom were the same as left to right. For example, $\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$ would be 5. We construct an NFA M, given by the following diagram:



M accepts B^R, which implies that B^R is regular. Because $(B^R)^R = B$, by part (a), B is regular. ■