1. We conjecture that the sum follows the formula $S(n) = \frac{n}{n+1}$. Writing the given series in summation notation, we have the following equation:

$$S(n) = \sum_{i=1}^{n} \frac{1}{i(i+1)} = \frac{n}{n+1}$$

Proof. When $n = 1$, the LHS of $S(1)$ is $\frac{1}{1(1+1)} = \frac{1}{2}$, and the RHS is $\frac{1}{2}$, so S(1) is true.

Without loss of generality, suppose that k is an arbitrary integer with $k \geq 1$ such that

$$S(k) = \sum_{i=1}^{k} \frac{1}{i(i+1)} = \frac{k}{k+1}$$

Assuming that this is true for $S(k)$, we must show that $S(k+1)$ is true:

$$\begin{aligned}
\sum_{i=1}^{k+1} \frac{1}{(i+1)(i+2)} &= \sum_{i=1}^{k} \frac{1}{i(i+1)} + \frac{1}{(k+1)(k+2)} \\
&= \frac{k}{k+1} + \frac{1}{(k+1)(k+2)} \\
&= \frac{k(k+2)}{(k+1)(k+2)} + \frac{1}{(k+1)(k+2)} \\
&= \frac{k^2+2k+1}{(k+1)(k+2)} \\
&= \frac{(k+1)^2}{(k+1)(k+2)} \\
&= \frac{k+1}{k+2} \\
&= \frac{k+1}{(k+1)+1}
\end{aligned}$$

■

2. *Proof.* We proceed by strong induction on n.

When $n = 1$, $F_1 = 1 < 2^1$.

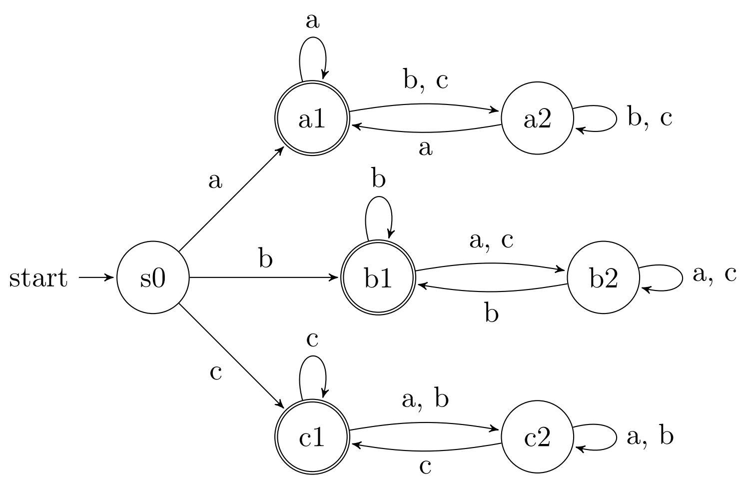
When $n = 2$, $F_2 = 1 < 2^2$.

Suppose that k is an arbitrary integer with $k \geq 2$ such that $F_k < 2^k$. Assuming that this is true for all k from 1 to k, we must show that $F_{k+1} < 2^{k+1}$:

$$\begin{aligned}
F_{k+1} &= F_k + F_{k-1} \\
F_k + F_{k-1} &< 2^k + 2^{k-1} \\
2^k + 2^{k-1} &= 2^k + \frac{2^k}{2} \\
2^k + \frac{2^k}{2} &= \frac{3}{2} \cdot 2^k \\
\frac{3}{2} \cdot 2^k &< 2 \cdot 2^k \\
2 \cdot 2^k &= 2^{k+1}
\end{aligned}$$

■

3. A DFA that takes in a string over the alphabet {a, b, c} and accepts exactly the strings that start and end with the same letter is shown below:



4. (a) A valid regular expression is as follows: A*B+A(A+B+A)*B[AB]*
   (b) A valid regular expression is as follows: c*(ac|bc|c)*[ab]?

5. *Proof.* We proceed by induction on a list of numbers of size m.

   When $m = 0$, L is empty and there are no elements in the list, and thus `(search n L)` correctly produces false, as an empty list cannot possibly contain n.

   Suppose that k is an arbitrary integer with $k \geq 0$ such that `(search n alist)` produces true if and only if n is in L, where L is a list of size k. Assuming that this is true for a list of size k, we must show that this function produces the correct boolean value for a list of size $(k+1)$.

   Because L has a size of at least 1, `(search n L)` examines the first element in the list, and compares this value to the given value of n. If and only if the two values are equal, the comparison produces true, signifying that n is in L. Otherwise, `(search n alist)` is called on the rest of the list, which is of size $(k+1) - 1 = k$.

   By the inductive hypothesis, this function correctly computes the value for a list of size k. Because n must be found in the first element or in the rest of the list if n is in L, the function `(search n alist)` is valid for a list of $(k+1)$ elements.

■