1. Remember: A finite state machine cannot count to arbitrarily high numbers, let alone do complex arithmetic:
   (a) yes, (b) no, (c) no, (d) yes
   (e) Yes: use the product construction (parallel execution) to construct a DFA for $L_1 \cap L_2$
   from the DFAs for $L_1$ and $L_2$ (explanation not required).
   (f) Yes: simple structural induction on regular expression for $L$ (explanation not re-
   quired).
   (g) Yes. The question is asking whether $L(M_1) \cap L(M_2)$ is nonempty. But the intersection
   of two regular languages is regular, and the test whether a regular language is (non)empty
   is decidable.
   (h) Yes. $L_1 \subseteq L_2$ iff $L_1 \cap \overline{L}_2$ is empty. But the complement of a regular language
   is regular, the intersection of two regular languages is regular, and the test whether a
   regular language is empty is decidable.

2. (a) There are several possible expressions:
   
   \[
   (1 + 01)^* (\varepsilon + 0) \\
   1^* (01^+)^* (\varepsilon + 0) \\
   (\varepsilon + 0) (1 + 10)^* 
   \]

   (b) Eliminating state $r$ gives:

   \[
   \begin{align*}
   p & \xrightarrow{a+b} q \\
   q & \xrightarrow{a+ba} q \\
   q & \xrightarrow{bb} p 
   \end{align*}
   \]

   An equivalent regular expression is:

   \[
   (a + b)(a + ba)^*bb^* (a + b)(a + ba)^*
   \]

   (c) Eliminating state $q$ from the initial automaton gives:

   \[
   \begin{align*}
   p & \xrightarrow{(a+b)a^*b} r \\
   r & \xrightarrow{a+b} r \\
   r & \xrightarrow{b} p 
   \end{align*}
   \]

   An equivalent regular expression is:

   \[
   ((a + b)a^* b(a+b)^*b)^* (a + b) a^*b(a+b)^*
   \]

   The final equivalent regular expression is thus:

   \[
   (a + b)(a + ba)^*bb^* (a + b)(a + ba)^* + ((a + b)a^*b(a+b)^*b)^* (a + b)a^*b(a+b)^*
   \]
3. (a) \( \{q, s\} \)

<table>
<thead>
<tr>
<th>( a )</th>
<th>( b )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \rightarrow p )</td>
<td>( qr )</td>
</tr>
<tr>
<td>( *qr )</td>
<td>( r )</td>
</tr>
</tbody>
</table>

(b) \( \begin{array}{c|c|c|c|}
*qs & \emptyset & q & \emptyset \\
*q & \emptyset & q & \emptyset \\
r & r & s & \emptyset \\
* & \emptyset & \emptyset & \emptyset \\
\end{array} \)

4. **Proof**

Suppose that \( L \) is regular.

Let \( n > 0 \) be the pumping lemma constant.

Choose \( w = 1^n + 0^n = 1^n \) (or, \( w = 10^n + 10^n = 10^{n+1} \), or \ldots).

By the pumping lemma, \( w \) can be broken into \( w = xyz \), where (a) \(|xy| \leq n\), (b) \(|y| \geq 1\), and (c) for all \( k \geq 0 \), \( xy^kz \in L \).

As \(|xy| \leq n\) and \(|y| \geq 1\), \( y = 1^k \), for \( 1 \leq k \leq n \).

By the pumping lemma, \( xy^0z \), which equals \( 1^{n-k} + 0^n = 1^n \), is an element of \( L \).

But \( 1^n \) is not the sum of \( 1^{n-k} \) and \( 0^n \), so \( 1^{n-k} + 0^n = 1^n \) is **not** an element of \( L \).

But this contradicts the fact that we just showed it **is** an element of \( L \).

So \( L \) cannot be regular.