1) States are 0, 1, 2, 3, 4, representing 0 mod 5, 1 mod 5, 2 mod 5, 3 mod 5, and 4 mod 5. \[ \Sigma = \{ 0, 1, 2 \} \]. In this solution, we'll accept \( E' \), assuming its value is 0.

3) The basic idea here is that the top row represents an even # of 1s and the bottom row represents an odd # of 1s. The state number mod 4 represents how many zeroes the string has, with states \( q_3, q_7 \) representing "3 or more 0s".
4) We assume that all strings in \( L \) must have at least one 1.

5) Consider the following NFA:

The language of this NFA is \( \Sigma^* - \varepsilon \).

6) Using Tommy's transformation, the NFA for the language \( \Sigma \) would be

But this NFA accepts \( \Sigma^* \), which is NOT \( \Sigma^* - \varepsilon \). Specifically, the string 1 is accepted by both machines, proving that the second does NOT accept the complement language of the first.

7) We will show that \( B \) is regular by creating a DFA that accepts \( B^R \). Our DFA will have 3 states representing these points in our computation:

\( q_0 \): What we have read in so far adds up correctly.

\( q_1 \): What we have read in matches all the answer bits, BUT we have a carry bit we are expecting in the next character to read in.

\( q_2 \): There is already a mistake in the answer bits. Nothing in this state can be accepted, regardless of what is read in in the future.
Explanation of transitions:

$q_0 \rightarrow q_0$: In all 3 of these, the first two bits add to the third, maintaining equality.

$q_0 \rightarrow q_1$: The parity of the bottom bit is the same as the sum of the top two, but a carry bit is produced, so the current string is no longer valid.

$q_0 \rightarrow q_2$: The parity of the answer bit is incorrect. All strings with this prefix are NOT in the designated language.

$q_1 \rightarrow q_0$: This fixes the previous carry bit problem.

$q_1 \rightarrow q_1$: The carry bit has cascaded to the next bit.

$q_1 \rightarrow q_2$: The current bit is now incorrect, so all strings with this prefix are NOT in the language.

8) $L_1 = 1(011)^*0$

$L_2 = (011)^*0101(011)^*$

$L_3 = (0110)(0110)(0110)(0110)(0110)$

$L_4 = E U 1(0111)^*(0110)$

$L_5 = (0^{*10^*})(0^{*10^*10^*})^* U 1^{*10^*10^*}$

Note: For each of these, it's important to include all possible strings that fit the given criteria.
9) $D_1: \begin{array}{c}
\begin{array}{c}
\text{GNFA-4:}
\end{array}
\end{array}$

1) $\begin{array}{c}
\text{RIP out 1:}
\end{array}$

2) $\begin{array}{c}
\text{RIP out 2:}
\end{array}$

Thus a R.E. that expresses the same language as $D_1$ is $a^*b(a\cup ba^*b)^*.$

3) $\begin{array}{c}
\text{RIP out 3:}
\end{array}$

Final R.E.: $\begin{array}{c}
\varepsilon U (a\cup b)a^*b[(b\cup a\cup b)a^*b]^+(a\cup \varepsilon)
\end{array}$
10) We will show that $L_1$ is not regular by showing that it does not satisfy the pumping lemma.

Let $p$ be the pumping length for $L_1$, assuming it's regular. Consider $S = 0^p1^p2^p$. The pumping lemma claims that $S = xyz$ with $|xy| \leq p$, $|y| > 0$.

Then $x = 0^i$, $y = 0^j$, $z = 0^{p-i-j}1^p2^p$.

Consider $xyz^2 = 0^i1^p2^p \notin L_1$, since $j > 0$.

This contradicts the pumping lemma. Thus $L_1$ is **NOT** regular.

We will show that $L_2$ is not regular by showing that it does not satisfy the pumping lemma.

Let $p$ be the pumping length for $L_1$, assuming it's regular. Consider $S = 0^p1^{p+1}$. The pumping lemma claims that $S = xyz$ with $|xy| \leq p$, $|y| > 0$.

Then $x = 0^i$, $y = 0^j$, $z = 0^{p-i-j}1^{p+1}$.

Consider $xyz^2 = 0^i1^{p+1}2^{p+1}$. The length of this string is $2(p+j+1)$. Since $p+j+1 \leq p+2j$, it follows that the first half of this string is all 0s while the second half is not. Thus, $xyz^2 \notin L_2$, contradicting the pumping lemma, proving $L_2$ is not regular.

We will show that $L_3$ is not regular by showing that it does not satisfy the pumping lemma. Let $p$ be the pumping length and consider the string $S = 0^p$. We must have $S = xyz$ with $x = 0^i$, $y = 0^j$, $z = 0^{p-i-j}$ and $1 \leq j \leq p$. Consider $xy^2z = 0^i1^p2^p$.

We claim that $xy^2z \notin L_3$. Note that $2^{p+j} \geq 2^p$.

The next smallest string in $L_3$ has length $2^{p+1}$.

We must show that $2^{p+j} < 2^{p+1}$. $2^{p+j} \leq 2^p \cdot 2^j < 2^p \cdot 2^j = 2^{p+j}$. It follows that $xy^2z \notin L_3$ and $L_3$ is not regular.
11) \(L_1\) is regular. Here is a DFA that accepts it:

```
\[ \text{State a represents that nothing has been read in.} \]
\[ \text{States b and c represent that the string read in so far satisfies the given criteria and starts with 0 or 1, respectively.} \]
\[ \text{State d represents a string starting with 0 and ending with 1.} \]
\[ \text{State e represents a string starting with 1 and ending with 0.} \]
```

Basically, any string starting and ending with the same character is accepted, since it contains an equal number of 01 and 10 transitions.

\(L_2\) is not regular. We will show that it does not satisfy the pumping lemma. Consider the string \(S = (00)^p(11)^p\), where \(p\) is the pumping length.

\(S \in L_2\). Let \(S = xyz\) with \(x = 0^j, y = 0^j, z = 0^{2p-1-j}(11)^p\). Consider \(xy^2z = 0^{2p+3j}1^{2p}\). \(xy^2z \notin L_2\) because it has \(2p+j-1\) occurrences of 00 and \(2p-1\) occurrences of 11. Since \(j > 0\), \(2p+j-1 \neq 2p-1\) and \(xy^2z \notin L_2\).

It follows that \(L_2\) is NOT regular.
12) Here is a drawing of the original DFA:

![DFA Diagram]

Algorithm

| [0, 2] | $S[1, 5] = \{0, 2\}, \text{on a} |
| [0, 4] | $S[1, 5] = \{0, 2\}, \{0, 4\} \text{ on a} |
| [1, 3] | $S[2, 4] = \{1, 3\} \text{ on a, b} |
| [1, 5] | $D[1, 5] = 1 \text{ by a, b} |
| [2, 4] | $D[0, 2] = 1 \text{ recursively} |
| [3, 5] | $D[3, 5] = 1 \text{ by a} |
| [3, 6] | $D[3, 6] = 1 \text{ by a} |

Thus the states that can "stay together" are $\{q_1, q_3\}$, $\{q_2, q_4\}$, and $\{q_5, q_6\}$.

Here is the minimized DFA:

![Minimized DFA Diagram]

This is the language of the set of strings with length 1, or 3 or more.