For each of the following languages over the alphabet $\Sigma = \{0, 1\}$, either prove the language is regular (by giving an equivalent regular expression, DFA, or NFA) or prove that the language is not regular (using a fooling set argument). Exactly half of these languages are regular.

1 $\{0^n 10^n \mid n \geq 0\}$

Solution: **Not regular:** Any two strings $x = 0^i$ and $y = 0^j$ are distinguished by the suffix $z = 10^i$. Thus, 0^* is a fooling set.

2 $\{0^n 10^n w \mid n \geq 0 \text{ and } w \in \Sigma^*\}$

Solution: **Not regular.** Any two strings $x = 0^i$ and $y = 0^j$ where $i < j$ are distinguished by the suffix $z = 10^i$. (It is crucial that $i < j$ here!) Thus, 0^* is a fooling set.

3 $X = \{w0^n 10^n x \mid w \in \Sigma^* \text{ and } n \geq 0 \text{ and } x \in \Sigma^*\}$

Solution: **Regular.** We might as well set $n = 0$, since any bigger value can be absorbed by the attached w and x. Namely, we have

$$X = \{w0^n 10^n x \mid w \in \Sigma^* \text{ and } n \geq 0 \text{ and } x \in \Sigma^*\} = \{w1x \mid w, x \in \Sigma^*\}.$$

This is the set of all strings containing the symbol 1, which is described by the regular expression $(0+1)^*1(0+1)^*$.

4 Strings in which the number of 0s and the number of 1s differ by at most 2.

Solution: **Not regular.** Any two strings $x = 0^i$ and $y = 0^j$ where $i < j$ are distinguished by the suffix $z = 1^{j+2}$. (It is crucial that $i < j$ here!) Thus, 0^* is a fooling set.

5 Strings such that *in every prefix*, the number of 0s and the number of 1s differ by at most 2.

Solution: **Regular.** Keep track of the difference between the number of 0s and the number of 1s seen so far. If this difference is ever less than -2 or greater than 2, reject; otherwise, accept. So we get a six-state DFA, where five of the states are accepting.

6 Strings such that *in every substring*, the number of 0s and the number of 1s differ by at most 2.

Solution: **Regular.** Keep track of the *current* difference between the number of 0s and the number of 1s seen so far. Also keep track of the *maximum* and *minimum* value of this difference seen so far. If the max-difference is ever more than min-difference+2, reject. Crudely, there are at most 45 possible values of (curr-dif, max-diff, min-diff), so we get a DFA with at most 46 states.
Alternatively, we can non-deterministically guess the range of differences ($-2 \leq \mathit{diff} \leq 0$ or $-1 \leq \mathit{diff} \leq 1$ or $0 \leq \mathit{diff} \leq 2$), build a separate DFA for each guess, and combine the three DFAs into a single 10-state NFA.