1. For each of the following languages over the alphabet $\Sigma = \{0, 1\}$, either prove that the language is regular (by constructing an appropriate DFA, NFA, or regular expression) or prove that the language is not regular (by constructing an infinite fooling set and proving that the set you construct is indeed a fooling set for that language).

   (a) $\{0^p 1^q 0^r \mid r = p + q\}$

Solution: Not regular.

Let F be the infinite language 0^*.
Let x and y be arbitrary strings from F.
Then $x = 0^i$ and $y = 0^j$ for some integers $i \neq j$.
Let $z = 10^{i+1}$.
Then $xz = 0^i 10^{i+1} \in L$ (with $p = i$ and $q = 1$ and $r = i + 1$).
But $yz = 0^j 10^{i+1} \notin L$ because $j + 1 \neq i + 1$.
So z is a distinguishing suffix of x and y.
It follows that F is a fooling set for L.
Because F is infinite, L cannot be regular. ■

Solution: Not regular.

Let F be the infinite language $\{0^n 1^n \mid n > 0\}$.
Let x and y be arbitrary strings from F.
Then $x = 0^i 1^i$ and $y = 0^j 1^j$ for some positive integers $i \neq j$.
Let $z = 0^{2i}$.
Then $xz = 0^i 1^i 0^{2i} \in L$ (with $p = q = i$ and $r = 2i$).
But $yz = 0^j 1^j 0^{2i} \notin L$ because $2j \neq i + i$.[a]
So z is a distinguishing suffix of x and y.
It follows that F is a fooling set for L.
Because F is infinite, L cannot be regular. ■

[a] We need $j > 0$ here, because $0^{2i} = 0^i 1^0 0^i$.

Rubric: 5 points = 1 for "not regular" + 4 for proof (standard fooling set rubric). This is more detail than necessary for full credit. These are not the only correct solutions.

(b) $\{0^p 1^q 0^r \mid r = p + q \bmod 2\}$

Rubric: During the exam, we announced a correction to the exam handout, changing "$p + q \bmod 2$" to "$(p + q) \bmod 2$". Despite this announcement, several students correctly read the original "$p + q \bmod 2$" as "$p + (q \bmod 2)$". **Correct solutions for either version of the question will receive full credit.**

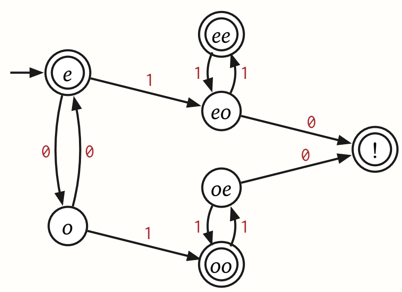
Corrected version: $\{0^p 1^q 0^r \mid r = (p + q) \bmod 2\}$

Solution: Regular.

$$0^{\text{even}}1^{\text{even}} + 0^{\text{odd}}1^{\text{even}}0 + 0^{\text{even}}1^{\text{odd}}0 + 0^{\text{odd}}1^{\text{odd}}$$
$$=$$
$$(00)^*(11)^* + 0(00)^*(11)^*0 + (00)^*1(11)^*0 + 0(00)^*1(11)^* \qquad \blacksquare$$

Solution: Regular. $(00)^*(0 + 1)(11)^*(0 + 1)$ $\blacksquare$

Solution: Regular. This language is accepted by the following DFA; all missing transitions go to a dump state.



The states have the following meanings:

- e — We've read an even number of 0s and no 1s.
- o — We've read an odd number of 0s and no 1s.
- ee — We've read an even number of 0s followed by a positive even number of 1s.
- eo — We've read an even number of 0s followed by odd number of 1s.
- oe — We've read an odd number of 0s followed by a positive even number of 1s.
- oo — We've read an odd number of 0s followed by odd number of 1s.
- ! — We've read $0^p 1^q 0$, where $p + q$ is odd.

$\blacksquare$

Rubric: 5 points = 1 for "regular" + 4 for proof (standard regular expression / DFA / NFA rubric). This is more detail than necessary for full credit. In particular, no explanation or proof of correctness is required for a regular expression, and no state names or explanations are required for a DFA or NFA. Optimizing the regular expression or DFA is not necessary. These are not the only correct solutions.

Uncorrected version: $\{0^p 1^q 0^r \mid r = p + (q \bmod 2)\}$

Solution: Not regular.

Let F be the infinite language 0^*.

Let x and y be arbitrary strings from F.

Then $x = 0^i$ and $y = 0^j$ for some integers $i \neq j$.

Let $z = 110^i$.

Then $xz = 0^i 110^i \in L$ (with $p = i$ and $q = 2$ and $r = i$).

But $yz = 0^j 110^i \notin L$ because $i \neq j + (2 \bmod 2)$.

So z is a distinguishing suffix of x and y.

It follows that F is a fooling set for L.

Because F is infinite, L cannot be regular. ∎

Rubric: 5 points = 1 for "not regular" + 4 for proof (standard fooling set rubric). This is more detail than necessary for full credit. This is not the only correct solution.

2. Let L be any regular language over the alphabet $\Sigma = \{0, 1\}$.

Choose exactly one of the following languages, and prove that your chosen language is regular. (In fact, *both* languages are regular, but we only want a proof for one of them.) Don't forget to tell us which language you've chosen!

(a) $L_1 = \{w \in \Sigma^* \mid \text{take2skip2}(w) \in L\}$.

Solution: Let $M = (Q, s, A, \delta)$ be an arbitrary DFA for the original language L. We design a **DFA** $M_1 = (Q_1, s_1, A_1, \delta_1)$ for L_1 as follows. The construction closely follows HW4.2(b).

$$Q_1 = Q \times \{0, 1, 2, 3\}$$
$$s_1 = (s, 0)$$
$$A_1 = A \times \{0, 1, 2, 3\}$$

$$\delta_1((q, 0), a) = (\delta(q, a), 1)$$
$$\delta_1((q, 1), a) = (\delta(q, a), 2)$$
$$\delta_1((q, 2), a) = (q, 3)$$
$$\delta_1((q, 3), a) = (q, 0)$$

State (q, i) means that M is in state q, and M_1 has read $4k + i$ symbols for some integer k. M_1 passes its next input symbol to M if and only if $i = 0$ or $i = 1$. ■

(b) $L_2 = \{\text{take2skip2}(w) \mid w \in L\}$.

Solution: Let $M = (Q, s, A, \delta)$ be an arbitrary DFA for the original language L. We design an NFA $M_2 = (Q_1, s_1, A_1, \delta_1)$ for L_2 as follows. The construction closely follows HW4.2(a).

$$Q_2 = Q \times \{0, 1, 2, 3\}$$
$$s_2 = (s, 0)$$
$$A_2 = A \times \{0, 1, 2, 3\}$$

$$\delta_2((q, 0), a) = \big\{(\delta(q, a), 1)\big\}$$
$$\delta_2((q, 1), a) = \big\{(\delta(q, a), 2)\big\}$$
$$\delta_2((q, 2), a) = \big\{(\delta(q, 0), 3),\ (\delta(q, 1), 3)\big\}$$
$$\delta_2((q, 3), a) = \big\{(\delta(q, 0), 0),\ (\delta(q, 1), 0)\big\}$$

State (q, i) means that M is in state q, and M_1 has sent $4k + i$ symbols to M, for some integer k. M_1 passes its next input symbol to M when $i = 0$ or $i = 1$, and guesses which symbol to send to M otherwise. ■

Rubric: 10 points, standard transformation rubric. —2 for choosing one language but then proving the other one is regular. These solutions are more detailed than necessary for full credit. In particular, the explanations in gray are not necessary. These are not the only correct solutions.

3. Prove that the following languages are not regular by building an infinite fooling set for each of them. For each language, prove that the set you constructed is indeed a fooling set.

(a) $\{0^p 1^q 0^r \mid r > 0 \quad \text{and} \quad q \bmod r = 0 \quad \text{and} \quad p \bmod r = 0\}$

Solution: Let F be the infinite language 00^*.

Let x and y be arbitrary strings from F.

Then $x = 0^i$ and $y = 0^j$ for some *positive* integers $i \neq j$.

Without loss of generality assume $i > j$. (The case $i < j$ is symmetric.)

Let $z = 1^i 0^i$.

Then $xz = 0^i 1^i 0^i \in L$ (with $p = q = r = i$).[a]

But $yz = 0^j 1^i 0^i \notin L$ (because $p \bmod r = j \bmod i = j > 0$).

So z is a distinguishing suffix of x and y.

It follows that F is a fooling set for L.

Because F is infinite, L cannot be regular. ■

[a] We need $i > 0$ here, because $0 \bmod 0$ is undefined.

Solution: Let F be the infinite language $\{0^n \mid n \text{ is prime}\}$.

Let x and y be arbitrary strings from F.

Then $x = 0^i$ and $y = 0^j$ for some prime numbers $i \neq j$.

Let $z = 1^i 0^i$.

Then $xz = 0^i 1^i 0^i \in L$ (with $p = q = r = i$).

But $yz = 0^j 1^i 0^i \notin L$ (because $p \bmod r = i \bmod j \neq 0$, because no prime number is a multiple of any other).

So z is a distinguishing suffix of x and y.

It follows that F is a fooling set for L.

Because F is infinite, L cannot be regular. ■

Rubric: 5 points, standard fooling set rubric. This is more detail than necessary for full credit. These are not the only correct solutions.

(b) $\{0^p 1^q \mid q > 0 \quad \text{and} \quad p = q^q\}$

Solution: Let F be the infinite language $\{0^{n^n} \mid n \geq 0\}$.

Let x and y be arbitrary strings from F.

Then $x = 0^{i^i}$ and $y = 0^{j^j}$ for some integers $i \neq j$.

Let $z = 1^i$.

Then $xz = 0^{i^i} 1^i \in L$ (with $p = i^i$ and $q = i$)

But $yz = 0^{j^j} 0^i \notin L$ (because $i^i \neq j^j$).

So z is a distinguishing suffix of x and y.

It follows that F is a fooling set for L.

Because F is infinite, L cannot be regular. ■

Solution: Let F be the infinite language 0^*.

Let x and y be arbitrary strings from F.

Then $x = 0^i$ and $y = 0^j$ for some integers $i \neq j$.

Let $z = 0^{k^k - i} 1^k$, for some integer k such that $k^k > i$.

Then $xz = 0^i 0^{k^k - i} 1^k = 0^{k^k} 1^k \in L$

But $yz = 0^j 0^{k^k - i} 1^k = 0^{k^k - i + j} 0^k \notin L$ (because $k^k - i + j \neq k^k$).

So z is a distinguishing suffix of x and y.

It follows that F is a fooling set for L.

Because F is infinite, L cannot be regular. ■

Rubric: 5 points, standard fooling set rubric. This is more detail than necessary for full credit. These are not the only correct solutions.

4. Consider the following recursive function:

$$\text{MINGLE}(w,z) := \begin{cases} z & \text{if } w = \varepsilon \\ \text{MINGLE}(x, aza) & \text{if } w = a \cdot x \text{ for some symbol } a \text{ and string } x \end{cases}$$

For example, $\text{MINGLE}(01, 10) = \text{MINGLE}(1, 0100) = \text{MINGLE}(\varepsilon, 101001) = 101001$.

(a) Prove that $|\text{MINGLE}(w,z)| = 2|w| + |z|$ for all strings w and z.

Solution: Let w and z be arbitrary strings.

Assume that $|\text{MINGLE}(x, y)| = 2|x| + |y|$ for all strings x and y such that $|x| < |w|$.

There are two cases to consider (mirroring the definition of MINGLE):

- Suppose $w = \varepsilon$. Then

$$\begin{aligned} |\text{MINGLE}(w,z)| &= |\text{MINGLE}(\varepsilon,z)| && \text{because } w = \varepsilon \\ &= |z| && \text{by definition of MINGLE} \\ &= 2 \cdot 0 + |z| && \text{by math} \\ &= 2|\varepsilon| + |z| && \text{by definition of } |\cdot| \\ &= 2|w| + |z| && \text{because } w = \varepsilon \end{aligned}$$

- Suppose $w = ax$ for some symbol a and string x. Then

$$\begin{aligned} &|\text{MINGLE}(w,z)| \\ &\quad = |\text{MINGLE}(ax,z)| && \text{because } w = ax \\ &\quad = |\text{MINGLE}(x,aza)| && \text{by definition of MINGLE} \\ &\quad = 2 \cdot |x| + |aza| && \text{by the induction hyp. } (y = aza) \\ &\quad = 2|x| + |a| + |z| + |a| && \text{because } |x \bullet y| = |x| + |y| \\ &\quad = 2(|a| + |x|) + |z| && \text{by math} \\ &\quad = 2|ax| + |z| && \text{by definition of } |\cdot| \\ &\quad = 2|w| + |z| && \text{because } w = ax \end{aligned}$$

In both cases we conclude that $|\text{MINGLE}(w,z)| = 2|w| + |z|$. ■

Rubric: 5 points, standard induction rubric. This is more detail than necessary for full credit; in particular, the explanations for each step (in gray) are not required.

(b) Prove that $\mathsf{MINGLE}(w, z \bullet z^R) = (\mathsf{MINGLE}(w, z \bullet z^R))^R$ for all strings w and z.

Solution: Let w and z be arbitrary strings.

Assume that $\mathsf{MINGLE}(x, y \bullet y^R) = (\mathsf{MINGLE}(x, y \bullet y^R))^R$ for all strings x and y such that $|x| < |w|$.

There are two cases to consider (mirroring the definition of MINGLE):

- Suppose $w = \varepsilon$. Then

$$\begin{aligned}
&\mathsf{MINGLE}(w, z \bullet z^R) \\
&\quad = \mathsf{MINGLE}(\varepsilon, z \bullet z^R) && \text{because } w = \varepsilon \\
&\quad = z \bullet z^R && \text{by definition of MINGLE} \\
&\quad = (z \bullet z^R)^R && \text{because } (zy)^R = y^R z^R \text{ and } (z^R)^R = z \\
&\quad = (\mathsf{MINGLE}(\varepsilon, z \bullet z^R))^R && \text{by definition of MINGLE} \\
&\quad = (\mathsf{MINGLE}(w, z \bullet z^R))^R && \text{because } w = \varepsilon
\end{aligned}$$

- Suppose $w = ax$ for some symbol a and string x. Then

$$\begin{aligned}
&\mathsf{MINGLE}(w, z \bullet z^R) \\
&\quad = \mathsf{MINGLE}(ax, z \bullet z^R) && \text{because } w = ax \\
&\quad = \mathsf{MINGLE}(x, az \bullet z^R a) && \text{by definition of MINGLE} \\
&\quad = \mathsf{MINGLE}(x, az \bullet (az)^R) && \text{by definition of } \cdot^R \\
&\quad = \big(\mathsf{MINGLE}(x, az \bullet (az)^R)\big)^R && \text{by the induction hyp. } (y = az) \\
&\quad = \big(\mathsf{MINGLE}(x, az \bullet z^R a)\big)^R && \text{by definition of } \cdot^R \\
&\quad = \big(\mathsf{MINGLE}(ax, z \bullet z^R)\big)^R && \text{by definition of MINGLE} \\
&\quad = \big(\mathsf{MINGLE}(w, z \bullet z^R)\big)^R && \text{because } w = ax
\end{aligned}$$

In both cases we conclude that $|\mathsf{MINGLE}(w, z)| = 2|w| + |z|$. ■

Rubric: 5 points, standard induction rubric. This is more detail than necessary for full credit; in particular, the explanations for each step are not required.

5. For each statement below, write "Yes" if the statement is always true and write "No" otherwise, and give a brief (one short sentence) explanation of your answer. Read these statements very carefully—small details matter!

(a) If L is a regular language over the alphabet $\{0, 1\}$, then $\{w1w \mid w \in L\}$ is also regular.

Solution: NO — Consider $L = 0^*$.

The language $\{w1w \mid w \in 0^*\} = \{0^n10^n \mid n \geq 0\}$ is not regular; 0^* is a fooling set. ■

(b) If L is a regular language over the alphabet $\{0, 1\}$, then $\{x1y \mid x, y \in L\}$ is also regular.

Solution: YES — $L \bullet 1 \bullet L$.

The definition of language concatenation implies $L \bullet 1 \bullet L = \{x1y \mid x, y \in L\}$. The concatenation of two regular languages is regular, by definition. ■

(c) The context-free grammar $S \to 0S1 \mid 1S0 \mid SS \mid 01 \mid 10$ generates the language $(0+1)^+$

Solution: NO — Can't generate the string 1.

This grammar generates all strings with the same number of 0s and 1s. In particular, every string generated by this grammar has even length. ■

(d) Every regular expression that does not contain a Kleene star (or Kleene plus) represents a finite language.

Solution: YES — No loops!

A star-free regular expression R cannot match strings longer than R. ■

(e) Let L_1 be a finite language and L_2 be an arbitrary language. Then $L_1 \cap L_2$ is regular.

Solution: YES — $L_1 \cap L_2$ is finite

...and every finite language is regular. ■

(f) Let L_1 be a finite language and L_2 be an arbitrary language. Then $L_1 \cup L_2$ is regular.

Solution: NO — Suppose $L_1 = \varnothing$ and L_2 is not regular.

Then $L_1 \cup L_2 = L_2$ is not regular! ■

(g) The regular expression $(00+01+10+11)^*$ represents the language of all strings over $\{0,1\}$ of even length.

> **Solution:** YES — Brute force FTW! ■

(h) The ε-reach of any state in an NFA contains the state itself.

> **Solution:** YES — through zero ε-transitions
> This is part of the formal definition of ε-reach. ■

(i) The language $L = 0^*$ over the alphabet $\Sigma = \{0,1\}$ has a fooling set of size 2.

> **Solution:** YES — $F = \{0,1\}$
> ε is a distinguishing suffix! ■

> **Solution:** YES — $L \neq \varnothing$ and $L \neq \Sigma^*$
> $\varnothing$ and Σ^* are the only two languages recognized by 1-state DFAs. ■

(j) Suppose we define an ε-DFA to be a DFA that can additionally make ε-transitions. Any language that can be recognized by an ε-DFA can also be recognized by a DFA that does not make any ε-transitions.

> **Solution:** YES — Every ε-DFA is an NFA.
> In particular, the incremental subset construction can be used to transform any ε-DFA into a standard DFA. ■

> **Rubric:** 10 points total = ½ for each correct yes/no + ½ for each explanation. These are not the only correct explanations. The gray text is not necessary for credit.