Basic

(1) **(Minimum pumping length)**

The PL says that every RL has an *associated* ***pumping length*** p, such that every string in the language can be pumped as long as it has at least p symbols.

Note that if p is a pumping length for a language then so is any other length $\geq p$. We define the ***minimum*** *pumping length* to be the smallest such p.

For example, if $L = \mathtt{ab}^* = \{\mathtt{a}, \mathtt{ab}, \mathtt{abb}, \ldots\}$ then its minimum pumping length is 2. This is because the string $w = \mathtt{ab}$ can be pumped by starring the b: $\mathtt{ab}^*$; while the shorter string $w = \mathtt{a}$ cannot be pumped.

For each of the following languages, give the minimum pumping length and justify your answer.

1) $\mathtt{aab}^*$
2) $\mathtt{a}^*\mathtt{b}^*$
3) $\mathtt{aab} + \mathtt{a}^*\mathtt{b}^*$
4) $\mathtt{a}^*\mathtt{b}^+\mathtt{a}^+\mathtt{b}^* + \mathtt{ba}^*$
   The notation $\mathtt{a}^+$ is equivalent to $\mathtt{aa}^*$, i.e. 1 or more a's (as opposed to $\mathtt{a}^*$ which means zero or more a's).
5) $(\mathtt{01})^*$
6) ε
7) $\mathtt{b}^*\mathtt{ab}^*\mathtt{ab}^*$
8) $\mathtt{10}(\mathtt{11}^*\mathtt{0})^*\mathtt{0}$
9) $\mathtt{1011}$
10) Σ^*

(2) **(PL applied to RLs)**

When we try to apply the Pumping Lemma to a Regular Language the **Prover** wins, and the **Falsifier** loses.

Show why **Falsifier** loses when L is one of the following RLs:

1) $(\mathtt{aa})^*$
2) $(\mathtt{aa} + \mathtt{bb})^*$
3) $\mathtt{01}^*\mathtt{0}^*\mathtt{1}$
4) $\{\mathtt{00}, \mathtt{11}\}$
5) $\emptyset$

Hint: For each language, find a suitable value for p and use it.

The following are almost complete proofs that some languages are not regular, using the Pumping Lemma (PL). Complete them by filling in the hidden details. (Some were done in the lecture using less formal notation.)

(3) Show that the language $L = \{0^n 1^n \mid n \geq 0\}$ is not regular.

❶ **Prover** claims L is regular and fixes the value of the pumping length p.

❷ **Falsifier** challenges **Prover** and picks $w = 0^p\boxed{} \in L$ and verifies it has the required length: $|w| = \boxed{} \geq p$.

❸ **Prover** tries to decompose w into three parts $w = \boxed{}$ but sees that the condition $|xy| \leq \boxed{}$ forces y to only contain the symbol $\boxed{}$.
Furthermore, y cannot just be the empty string because of the condition $\boxed{}$
So it is forced to choose $y = 0^d$ for some $d \geq 1$.

❹ **Falsifier** now sees that

$$xy^2z = xy\boxed{}z = 0^p\boxed{}1^p = 0^{p+d}1^p.$$

and hence xy^2z does not belong to L. This is because $d \geq 1 \implies p + d > p$, and hence xy^2z has more $\boxed{}$'s than there are $\boxed{}$'s. (They need to be equal for it to be in the language.)

(Note that we could use any of $xy^3z, xy^4z, \ldots$. In fact, we could have even used $xy^0z = xz$; we end up with less 0's than there are 1's.)

(4) $L = \{ww \mid w \in \{0, 1\}^*\}$.

❶ **Prover** claims L is regular and fixes the value of the pumping length p.

❷ **Falsifier** challenges **Prover** and chooses $w = (0^p1)(0^p1) \in L$.
This has length

$$|w| = (p+1) + (\boxed{}) = \boxed{} \geq p.$$

❸ **Prover** The PL now guarantees that w can be split into three substrings $w = xyz$ satisfying $|xy| \leq p$ and $y \neq \varepsilon$.

❹ **Falsifier** Since

$$w = (0^p1)(\boxed{}) = xyz$$

with $|xy| \leq p$ then we must have that y only contains the symbol $\boxed{}$.
We can then pump y and produce $xy^2z = xyyz \notin L$ because the first half $\boxed{}$ the second half.
So L is not regular.

Basic

(5) $L = \{a^ib^jc^k \mid 0 \leq i < j < k\}$

❶ **Prover** claims L is regular and fixes the value of the pumping length p.

❷ **Falsifier** challenges **Prover** and chooses

$$w = a^{\boxed{}}b^{\boxed{}+1}c^{\boxed{}+2}.$$

Here $|w| = p + \boxed{}\ \boxed{}\ p$

❸ **Prover** writes

$$w = (xy)z = (a^p)b^{p+1}c^{p+2}$$

where xy is a string of $\boxed{}$'s only

❹ **Falsifier** forms

$$xy^2z = a^{p+\boxed{}}b^{p+1}c^{p+2} \notin L$$

because $|y| \geq 1$.

Basic: "pumping down"

(6) $L = \{\texttt{a}^i\texttt{b}^j \mid i > j\}$

❶ **Prover** claims L is regular and fixes the value of the pumping length p.

❷ **Falsifier** challenges **Prover** and chooses

$$w = \texttt{a}^{\square+1}\texttt{b}^{\square}$$

Here $|w| = \square = 2p + 1 \,\square\, p$

❸ **Prover** writes

$$w = (xy)z = (\texttt{a}^{\square})\texttt{ab}^{\square}$$

i.e. xy is a string of $\square$'s only

❹ **Falsifier** forms

$$xy^0z = xz = \texttt{a}^{p+1-\square}\texttt{b}^p \notin L$$

because $|y| \geq 1$. (so $p + 1 - \square \leq p$).

(7) $L = \{\texttt{a}^i\texttt{b}^j\texttt{c}^k \mid i > j > k \geq 0\}$

❶ **Prover** claims L is regular and fixes the value of the pumping length p.

❷ **Falsifier** challenges **Prover** and chooses

$$w = \texttt{a}^{\square}\texttt{b}^{p+1}\texttt{c}^0.$$

Here $|w| = \square + (p+1) + 0 \,\square\, \square$.

❸ **Prover** writes

$$w = \texttt{a}^{\square}\texttt{a}^2\texttt{b}^{p+1}\texttt{c}^0 = xyz,$$

where xy can have a maximum of $\square$ symbols, so xy must be a string of $\square$'s only

❹ **Falsifier** forms

$$xy^{\square}z = xz = \texttt{a}^{\square-|y|}\texttt{b}^{p+1}\texttt{c}^0 \notin L$$

because $|y| \geq 1$.

Go through the JFLAP tutorial on: https://www.jflap.org/tutorial/pumpinglemma/regular/ and then try all the "games."

JFLAP plays the role of **Falsifier** and you play the role of **Prover** .

Note that *some of the languages below are actually regular* – in this case, you will need to devise a strategy for **Prover** to always win no matter what **Falsifier** chooses as a challenge string.

JFLAP's notation:

- m is used instead of p (the pumping length).
- i is used instead of k in xy^kz.
- $n_{\mathtt{a}}(w)$: the number of occurrence of the symbol a in the string w. e.g. $n_{\mathtt{a}}(\mathtt{aba}) = 2$ and $n_{\mathtt{b}}(\mathtt{aba}) = 1$.
- w^R: the reverse string of w, e.g. $\mathtt{abb}^R = \mathtt{bba}$.

Assume $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ unless otherwise specified.

The list of languages is as follows:

1. $\{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$ Hint: $\mathtt{a}^p\mathtt{b}^p$
2. $\{w \in \Sigma^* \mid n_{\mathtt{a}}(w) < n_{\mathtt{b}}(w)\}$ Hint: $\mathtt{a}^p\mathtt{b}^{p+1}$
   i.e. language of strings which have less a's than there are b's.
3. $\{ww^R \mid w \in \Sigma^*\}$ Hint: $\mathtt{a}^p\mathtt{b}^{2p}\mathtt{a}^p$
4. $\{(\mathtt{ab})^n\mathtt{a}^m \mid n > m \geq 0\}$ Hint: $(\mathtt{ab})^{p+1}\mathtt{a}^p$
5. $\{\mathtt{a}^n\mathtt{b}^m\mathtt{c}^{n+m} \mid n \geq 0, m \geq 0\}$
6. $\{\mathtt{a}^n\mathtt{b}^\ell\mathtt{a}^k \mid n > 5, \ell > 3, \ell \geq k\}$
7. $\{\mathtt{a}^n \mid n \text{ is even}\}$ Hint: Regular
8. $\{\mathtt{a}^n\mathtt{b}^m \mid n \text{ is odd } \textbf{or } m \text{ is even}\}$ Hint: Regular
9. $\{\mathtt{bba}(\mathtt{ba})^n\mathtt{a}^{n-1} \mid n \geq 1\}$
10. $\{\mathtt{b}^5w \mid w \in \Sigma^* \textbf{ and } 2n_{\mathtt{a}}(w) = 3n_{\mathtt{b}}(w)\}$
11. $\{\mathtt{b}^5w \mid w \in \Sigma^* \textbf{ and } n_{\mathtt{a}}(w) + n_{\mathtt{b}}(w) \equiv 0 \pmod 3\}$
12. $\{\mathtt{b}^m(\mathtt{ab})^n(\mathtt{ba})^n \mid m \geq 4, n \geq 1\}$
13. $\{(\mathtt{ab})^{2n} \mid n \geq 1\}$ Hint: Regular

Warning: The games played by JFLAP are for a specific challenge string. This is only meant to give you a feel for how the general game proceeds. When we write our proofs we are not allowed to choose a fixed value for p.

(1) Let $\Sigma = \{\texttt{0}, \texttt{1}, \texttt{+}, \texttt{=}\}$, and `ADD` be the language given by

$$\{u\texttt{=}v\texttt{+}w \mid u, v, w \text{ are binary integers, and } u \text{ is the sum of } v \text{ and } w \text{ in the usual sense}\}$$

Show that `ADD` is not regular.

(2) Let $L = \{1^{2^n} \mid n \geq 0\}$. Show that L cannot be regular.

Intermediate

(3) $L = \{\mathtt{a}^i\mathtt{b}^j\mathtt{c}^k \mid j \neq i \text{ or } j \neq k\}$

❶ **Prover** claims L is regular and fixes the value of the pumping length p.

❷ **Falsifier** challenges **Prover** and chooses

$$w = \mathtt{a}^p\mathtt{b}^{\boxed{\quad}}\mathtt{c}^{\boxed{\quad}}$$

Here $|w| = p + 2(\boxed{\quad}) \geq p$.

❸ **Prover** writes

$$w = (xy)z = (\mathtt{a}^p)\mathtt{b}^{\boxed{\quad}}\mathtt{c}^{\boxed{\quad}}$$

where xy is a string of a's only

❹ **Falsifier** forms

$$xy^kz = a^{p+(k-1)|y|}\mathtt{b}^{\boxed{\quad}}\mathtt{c}^{\boxed{\quad}}$$

where $k = 1 + \boxed{\quad}/\boxed{\quad}$. This gives $a^{p!+p}\mathtt{b}^{\boxed{\quad}}\mathtt{c}^{\boxed{\quad}}$ which is not in the language.