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

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

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

**Solution**

You can design minimal NFAs for the languages and then use the type of discussion shown in the lecture, but it is safer to list the first few strings in the language, in increasing length, and then check to see at what length you can find a substring that you can pump from an accepted string. Check this demo.

1) $\texttt{aab}^* = \{\texttt{aa}, \texttt{aab}, \texttt{aab}^2, \texttt{aab}^3, \texttt{aab}^4, \texttt{aab}^5, \texttt{aab}^6, \ldots\}$, sorted in ascending order with respect to string length.

   We notice that aa cannot be pumped (e.g. if we repeat a once then we get aaa which is not in the language), but starting from aab we can pump b to get $\texttt{aab}^k$ for $k = 0, 1, 2, \ldots$ which are all in the language, so the pumping length for this language is $\boxed{p = 3}$ (the length of aab, the shortest string that can be pumped).

2) $\texttt{a}^*\texttt{b}^* = \{\varepsilon, \texttt{a}, \texttt{b}, \texttt{a}^2, \texttt{b}^2, \texttt{ab}, \texttt{a}^3, \texttt{b}^3, \texttt{aab}, \texttt{abb}, \texttt{a}^4, \texttt{b}^4, \ldots\}$

   $\varepsilon$ is not pump-able, but a or b are, so $\boxed{p = 1}$.

3) $\texttt{aab} + \texttt{a}^*\texttt{b}^* = \{\texttt{aab}\} \cup \{\varepsilon, \texttt{a}, \texttt{b}, \texttt{a}^2, \texttt{b}^2, \texttt{ab}, \texttt{a}^3, \texttt{b}^3, \texttt{aab}, \texttt{abb}, \texttt{a}^4, \texttt{b}^4, \ldots\}$ which is just $\{\varepsilon, \texttt{a}, \texttt{b}, \texttt{a}^2, \texttt{b}^2, \texttt{ab}, \texttt{a}^3, \texttt{b}^3, \texttt{aab}, \texttt{abb}, \texttt{a}^4, \texttt{b}^4, \ldots\} = \texttt{a}^*\texttt{b}^*$ again, so $\boxed{p = 1}$.

4) $\texttt{a}^*\texttt{b}^+\texttt{a}^+\texttt{b}^* + \texttt{ba}^* = \{\texttt{ba}, \texttt{aba}, \texttt{bab}, \texttt{bba}, \texttt{aab}, \} \cup \{\texttt{b}, \texttt{ba}, \texttt{ba}^2, \texttt{ba}^3, \ldots\}$

This is a union of two languages:

- The RegEx $\texttt{a}^*\texttt{b}^+\texttt{a}^+\texttt{b}^*$ gives $p = 2$ as we can loop a or b from its shortest string $\texttt{a}^0\texttt{b}^1\texttt{a}^1\texttt{a}^0 = \texttt{ba}$.
- The RegEx $\texttt{ba}^*$ also gives $p = 2$ as we can loop a from its second shortest string ba.

So, we conclude that the given language has $\boxed{p = 2}$ (the shortest of the two lengths, which happen to be the same in this example).

5) $(\texttt{01})^* = \{\varepsilon, \texttt{01}, \texttt{0101}, \texttt{010101}, (\texttt{01})^4, (\texttt{01})^5, \ldots\}$

So starting from 01 we can set $x = z = \varepsilon$ and $y = \texttt{01}$ in the Pumping Lemma. Hence, $\boxed{p = 2}$, the length of 01.

6) $\varepsilon$

This RegEx represents the language that only contains the empty string: $\{\varepsilon\}$. There is no way of writing $\varepsilon = xyz$ with $y \neq \varepsilon$, so it suffices to let $\boxed{p = 1}$. The language is finite (and therefore regular), and there are no pump-able strings!

**Note: $\varepsilon$ is the only possible string of length zero over any alphabet, and it is not pump-able in any language, so $p$ is always $\geq 1$, unless the language is the empty language $\emptyset$.**

7) $\texttt{b}^*\texttt{ab}^*\texttt{ab}^* = \{\texttt{aa}, \texttt{baa}, \texttt{aba}, \texttt{aab}, \texttt{b}^2\texttt{aa}, \texttt{ab}^2\texttt{a}, \texttt{aab}^2, \texttt{baba}, \texttt{baab}, \texttt{abab}, \ldots\}$

Here, $\texttt{b}^0\texttt{ab}^0\texttt{ab}^0 = \texttt{aa}$ is not pump-able (if pumped then it would produce $\texttt{aa}^+$ which is not of the required form $\texttt{b}^*\texttt{ab}^*\texttt{ab}^*$).

However, all the strings of length 3 are pump-able producing e.g. $\texttt{b}^*\texttt{aa}$ from baa. So $\boxed{p = 3}$.

8) $\texttt{10}(\texttt{11}^*\texttt{0})^*\texttt{0} = \{\texttt{100}, \texttt{10100}, \texttt{101100}, \texttt{1011100}, \texttt{1010100}, \ldots\}$

$\texttt{100} = \texttt{10}(\texttt{11}^*\texttt{0})^0\texttt{0}$ is not pump-able, but $\texttt{10100} = \texttt{10}(\texttt{11}^0\texttt{0})^1\texttt{0}$ is pump-able producing $\texttt{10100} = \texttt{10}(\texttt{10})^*\texttt{0}$, so $\boxed{p = 5}$.

9) 1011

This RegEx represents the language that only contains one string: $\{\texttt{1011}\}$. If we pump any symbol then the length of the resulting string will be at least 5, so it cannot be a member of this language. Hence, it suffices to let $\boxed{p = 5}$. The language is finite (and therefore regular), and there are no pump-able strings!

10) $\Sigma^* = \{\varepsilon, \ldots\}$ is the language of all possible strings over the alphabet $\Sigma$.

In particular, if a is a symbol then $\texttt{a}^*$ is also in $\Sigma^*$, so $\boxed{p = 1}$.

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

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

**Solution**

1) Check this demo.
2) $\{\texttt{00}, \texttt{11}\}$

   This is a finite language, so **Prover** chooses $p = 3$. **Falsifier** cannot choose a string that is long enough. ($|w| \geq 3$ but the two available strings are only 2 symbols long.)
3) $(\texttt{aa} + \texttt{bb})^*$

   **Prover** chooses $p = 2$ and $y = \texttt{aa}$ or $\texttt{bb}$, depending on the string chosen by the **Falsifier** .
4) $\texttt{01}^*\texttt{0}^*\texttt{1}$

   **Prover** chooses $p = 3$ and $y = \texttt{0}$ or $\texttt{1}$, depending on the string chosen by the **Falsifier** .
5) $\emptyset$

   **Falsifier** has no strings to choose from! **Prover** may set $p = 0$ or any other value.

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 $w = 0^p \boxed{1^p} \in L$ and verifies it has the required length: $|w| = \boxed{2p} \geq p$.

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

❹ **Falsifier** now sees that

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

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 $\boxed{0}$'s than there are $\boxed{1}$'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

$$|w| = (p+1) + (\boxed{p+1}) = \boxed{2p+2} \geq p.$$

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

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

with $|xy| \leq p$ then we must have that $y$ only contains the symbol $\boxed{0}$.
We can then pump $y$ and produce $xy^2z = xyyz \notin L$ because the first half $\boxed{\text{no longer matches}}$ the second half.
So $L$ is not regular.

(5) $L = \{\mathsf{a}^i\mathsf{b}^j\mathsf{c}^k \mid 0 \leq i < j < k\}$

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

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

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

Here $|w| = p + \boxed{(p+1)+(p+2)} \boxed{\geq} p$

❸ **Prover** writes

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

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

❹ **Falsifier** forms

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

because $|y| \geq 1$.

Basic: "pumping down"

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

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

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

$$w = \mathtt{a}^{\boxed{p}+1}\mathtt{b}^{\boxed{p}}$$

Here $|w| = \boxed{(p+1)+p} = 2p+1 \boxed{\geq} p$

❸ **Prover** writes

$$w = (xy)z = (\mathtt{a}^{\boxed{p}})\mathtt{ab}^{\boxed{p}}$$

i.e. $xy$ is a string of $\boxed{\mathtt{a}}$'s only

❹ **Falsifier** forms

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

because $|y| \geq 1$. (so $p+1-\boxed{|y|} \leq p$).

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

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

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

$$w = \mathtt{a}^{\boxed{p+2}}\mathtt{b}^{p+1}\mathtt{c}^0.$$

Here $|w| = \boxed{p+2} + (p+1) + 0 \boxed{\geq} \boxed{p}$.

❸ **Prover** writes

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

where $xy$ can have a maximum of $\boxed{p}$ symbols, so $xy$ must be a string of $\boxed{\mathtt{a}}$'s only

❹ **Falsifier** forms

$$xy\boxed{0}z = xz = \mathtt{a}^{\boxed{p+2}-|y|}\mathtt{b}^{p+1}\mathtt{c}^0 \notin L$$

because $|y| \geq 1$.

Optional – JFLAP

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 $\mathtt{a}$ 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 $\mathtt{a}$'s than there are $\mathtt{b}$'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

**Solution**

Check this demo.

**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 $\Sigma = \{0, 1, +, =\}$, and ADD be the language given by

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

Show that ADD is not regular.

**Solution**

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

❷ **Falsifier** challenges **Prover** and chooses $w$ to be $1^p = 0^p + 1^p$

❸ **Prover** can only have 1's in $y$, so $y = 1^d$ for some $d \geq 1$

❹ **Falsifier** constructs $xyyz$ and finds it to be

$$1^{p+d} = 0^p + 1^p$$

which is not correct.

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

**Solution**

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

❷ **Falsifier** challenges **Prover** and $w = 1^{2^p}$

❸ **Prover** writes

$$x = 1^a, y = 1^b, z = 1^{2^p - a - b}$$

where $1 \leq b \leq p$.

❹ **Falsifier**

$$xyyz = 1^{2^p + b}$$

The next string after $1^{p^2}$ in terms of length is $1^{2^{p+1}} = 1^{2^p + 2^p}$ but

$$2^p < 2^p + b < 2^p + 2^p.$$

because

$$1 \leq b \leq p < 2^p$$

So $xyyz \notin L$.

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

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

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

$$w = \mathsf{a}^p\mathsf{b}^{\boxed{p!+p}}\mathsf{c}^{\boxed{p!+p}}$$

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

❸ **Prover** writes

$$w = (xy)z = (\mathsf{a}^p)\mathsf{b}^{\boxed{p!+p}}\mathsf{c}^{\boxed{p!+p}}$$

where $xy$ is a string of a's only

❹ **Falsifier** forms

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

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