

IMPORTANT: This proof was automatically translated from original in Czech language to English!

(0.5 point) Decide whether the following language L_1 is regular or not. If the language is not regular, formally prove this claim. If the language is regular, describe it using one of the formalisms for describing regular languages (i.e., FA (Finite Automaton), RG (Regular Grammar), or RE (Regular Expression)). Also, find the smallest pumping lemma constant p (whose existence is guaranteed by the pumping lemma for regular languages) and explain why your constant is correct and minimal.

$$L_1 = \{w : w \in \{a, b\}^* \wedge |w|_a \bmod 3 = |w|_b\} \cap \{a^m b^j b^i : m, j, i \in \mathbb{N}_0 \wedge j < m\}$$

Note: Using the notation $|x|_a$ to denote the number of symbols a in the string x .

Solution:

- (1) The language L_1 is the intersection of two languages L_l and L_r (with l as "left", r as "right"), i.e., $L_1 = L_l \cap L_r$. Let's examine the individual languages L_l and L_r :

Language L_l :

$$L_l = \{w : w \in \{a, b\}^* \wedge |w|_a \bmod 3 = |w|_b\}$$

L_l represents all possible strings over the alphabet $\Sigma = \{a, b\}$, where the number of symbols b is equal to the number of symbols a modulo 3. As a consequence, the number of symbols b is at least 0 and at most 2. Examples of possible strings from this language:

$$L_l = \{\epsilon, ab, ba, aabb, abab, bbaa, baab, aaa, aaaab, aabaa, aaaaabb, aaaaaa, \dots\}$$

Language L_r :

$$L_r = \{a^m b^j b^i : m, j, i \in \mathbb{N}_0 \wedge j < m\}$$

Examples of strings:

$$L_r = \{a, ab, abb, abbb, abbbb, \dots, aab, aabb, aabbb, aaabbbb, \dots\}$$

We notice that L_r specifies a certain structure of the string—all symbols a occur only on the left, and symbols b only on the right.

Furthermore, we see that the minimal string is $w = a$, because we have the condition $j < m$, and we also know that the smallest possible j is 0 (since $j \in \mathbb{N}_0$), which means that m cannot be 0 (otherwise, we would have a contradiction that j must be less than zero and at the same time greater than or equal to zero). That is, essentially, $m \geq 1$, i.e., $m \in \mathbb{N}$ but excluding zero.

Another observation is that i can be arbitrary and does not depend on any other variable. That is, for all $m \in \mathbb{N}$, if we fix $j = 0$, then using i we can generate any number of symbols b , and it will always hold that $j < m$.

If we combine both observations above, we can simplify the language L_r as follows:

$$L_r = \{a^m b^j b^i : m, j, i \in \mathbb{N}_0 \wedge j < m\} = \{a^m b^i : i \in \mathbb{N}_0 \wedge m \in \mathbb{N}\}$$

In summary, we obtain the simplified intersection of the languages:

$$L_1 = \{w : w \in \{a, b\}^* \wedge |w|_a \bmod 3 = |w|_b\} \cap \{a^m b^i : i \in \mathbb{N}_0 \wedge m \in \mathbb{N}\}$$

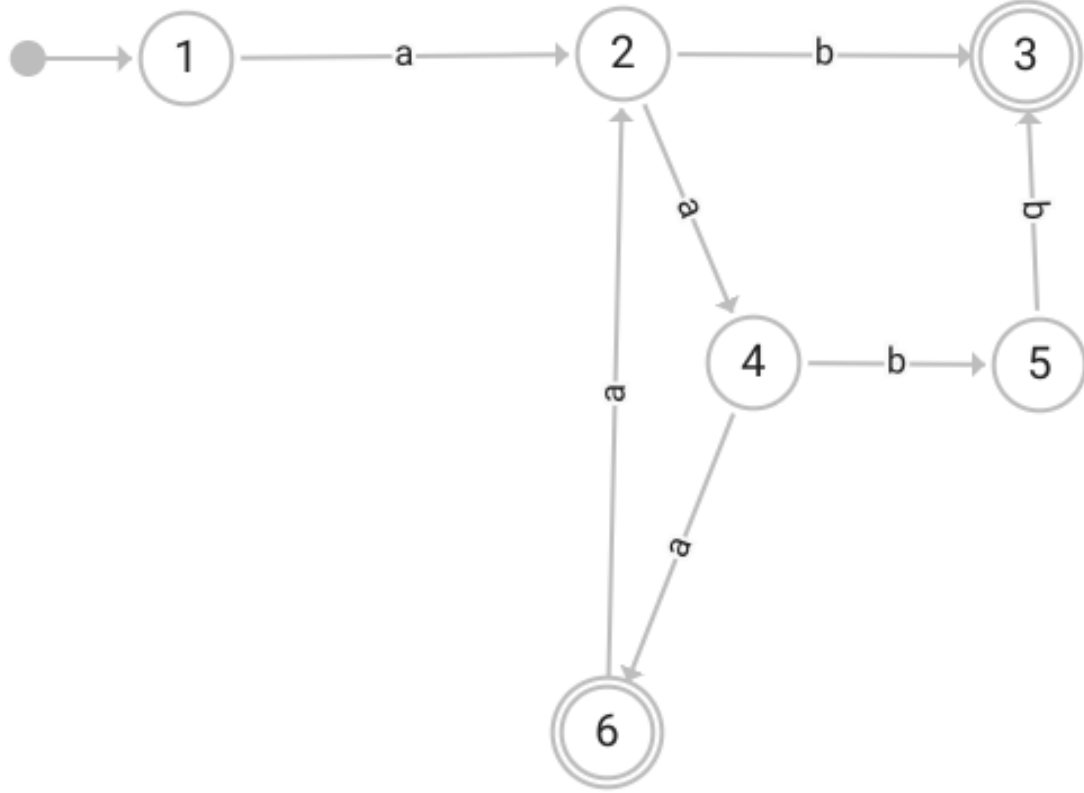
We notice that L_r gives us the structure of the language L_1 —that is, on the left are always symbols a , and on the right are always symbols b . The next and last restriction that L_r makes is that the minimal string is a . Otherwise, it can contain any number of additional symbols a on the left and any number of symbols b on the right.

If we now apply the restriction from L_l to L_r , we see that on the right, we can have 0, 1, or 2 symbols b (depending on the number of symbols a on the left).

We can therefore simplify the language L_1 as:

$$L_1 = \{a^m b^n : m \in \mathbb{N}, n \in \mathbb{N}_0 \wedge |w|_a \bmod 3 = |w|_b\}$$

To prove that this language is regular, we could utilize closure properties for intersection. That is, if we could prove that L_l and L_r are regular, then their intersection would also be regular. But now that we have the simplified description L_1 , which is equivalent, it will be much easier to prove the regularity of this simplified description—that is, we need to describe it using one of the formalisms for regular languages. And we will do that using this finite automaton:



That is, there exists a FA (in this case, a DFA) generating this language; therefore, this language is regular.

- (2) Formally, we will verify whether this language has the pumping property (we already know that it should have it; otherwise, this language could not be regular).

We want to verify that there exists a constant $p \in \mathbb{N}$ such that for every string $w \in L_1$, where if $|w| \geq p$, then there exists some decomposition of the string $w = xyz$ such that $|y| \geq 1$ and also $|xy| \leq p$ and also $\forall k \in \mathbb{N}_0$ it holds that $xy^kz \in L_1$.

Formally:

$$(\exists p \geq 1)(\forall w \in L_1)[|w| \geq p \Rightarrow (\exists x, y, z \in \Sigma^*)(w = xyz \wedge |xy| \leq p \wedge |y| \geq 1 \wedge (\forall k \geq 0)(xy^kz \in L_1))]$$

At the same time, we need to find this constant p minimal.

Let's try $p = 1$:

Take the string $w = a$, then there exists exactly one decomposition w :

$$\begin{aligned}x &= \epsilon \\y &= a \\z &= \epsilon\end{aligned}$$

Then for $k = 0$, we get $xy^0z = \epsilon a^0 \epsilon = \epsilon \notin L_1$. Thus, we cannot ensure the pumping property for all strings of length at least 1; we must take a larger constant p .

For $p = 2$:

As a counterexample, we take the string $w = ab$. It has two different decompositions, but for each, we will certainly find some k for which the resulting string will not be in the language because the number of symbols a and b will not match.

For example:

$$\begin{aligned}x &= \epsilon \\y &= a \\z &= b\end{aligned}$$

Then for $k = 0$, we get $xy^0z = \epsilon a^0 b = b \notin L_1$.

If we choose the other remaining decomposition, again for $k = 0$, we get the empty string, which does not belong to the language.

And we have no other decompositions; for this $p = 2$, we cannot prove the pumping property for the language.

For $p = 3$:

We choose again the minimal string of length 3:

$$w = aaa$$

.

If we choose $y = aaa$, then $\forall k \geq 1$ the pumping property holds.

But for $k = 0$, we get the empty string, which does not belong to the language. If we try the remaining two decompositions, we will certainly find such k where the number of symbols a and b will not match.

Observation: We see that if we pump the substring aaa , we never violate the condition on the number of symbols a and b . That is, generally, we look for decompositions where $y = aaa$, but at the same time, we do not get the empty string.

For $p = 4$:

The minimal string will be $w = aabb$. Similarly, for each decomposition, we will certainly find such k that the resulting string will not belong to the language due to violating the correct number of symbols a and b .

For $p = 5$:

The minimal string will be $w = aaaab$. We also notice that for every string $w \geq 5$, we have that string in the form $a^3 a^m b^{m \bmod 3}$, $\forall m \in \mathbb{N}$.

We always choose the decomposition:

$$\begin{aligned}x &= \epsilon \\y &= a^3 \\z &= a^m b^{m \bmod 3}\end{aligned}$$

And for every such string in this form and for every $k \in \mathbb{N}_0$, the pumping property holds.

So for $p = 5$, the pumping property holds for the language L_1 , which is a necessary condition (though not sufficient) for the regularity of the language L_1 . At the same time, the found constant $p = 5$ is minimal because for all smaller values, we have refuted the pumping properties for certain strings and constants k .

□

(0.5 point) Decide whether the following language L_2 is regular or not. If the language is not regular, formally prove this claim. If the language is regular, describe it using one of the formalisms for regular languages (i.e., FA, RG, or RE). Also, find the smallest pumping lemma constant p (whose existence is guaranteed by the pumping lemma for regular languages) and explain why your constant is correct and minimal.

$$L_2 = \{2^m 1^n 2^j : m, n, j \in \mathbb{N} \wedge m > 1 \wedge n, j \geq 1 \wedge j \neq m + n\}$$

Solution:

We see that the number of symbols 2 on the right must not be equal to the sum of all other symbols on the left.

Examples of strings contained in this language:

$$L_2 = \{2212, 22122, 2212222, 22112, \dots\}$$

Observation: The minimal string is 2212.

So far, we might have the intuition that this language does not have the pumping property due to the condition $j \neq m + n$. So we will try to refute the pumping property, i.e., that for the given language L_2 , the pumping property does not hold, and therefore the necessary condition for the regularity of this language is not met (i.e., we use the contrapositive of the implication in the pumping lemma):

If L_2 does not have the pumping property, then it is not regular.

We need to prove the negation of the pumping property:

$$\begin{aligned} \neg(\exists p \geq 1)(\forall w \in L_2)[|w| \geq p \Rightarrow (\exists x, y, z \in \Sigma^*)(w = xyz \wedge |xy| \leq p \wedge |y| \geq 1 \wedge (\forall k \geq 0)(xy^k z \in L_2))] \equiv \\ \equiv (\forall p \geq 1)(\exists w \in L_2)[|w| \geq p \wedge (\forall x, y, z \in \Sigma^*)((w = xyz \wedge |xy| \leq p \wedge |y| \geq 1) \Rightarrow (\exists k \geq 0)(xy^k z \notin L_2))] \end{aligned}$$

If we can prove this negation, then it will follow that L_2 is not regular.

For the proof of irregularity, we will prove the irregularity of the complement of L_2 . The reason is that perhaps the proof for L_2 would lead to complex calculations. It will be simpler to refute the pumping property for the complement of the language L_2 .

We know from the properties of regular languages that regular languages are closed under the complement operation. That is, it holds that:

$$L \text{ is regular} \Leftrightarrow \Sigma^* \setminus L = \bar{L} \text{ is regular}$$

That is, we will refute the pumping property for the language $\bar{L}_2 = \Sigma^* \setminus L_2$.

Specifically, we take some string where it holds that $j = m + n$. Any such string is not in L_2 , but certainly is in \bar{L}_2 . For example:

$$\begin{aligned} w &= 2^p 12^{p+2} \\ w &= 2_1 2_2 \dots 2_{p-1} 2_p 12_1 2_2 \dots 2_{p+2} \\ \text{Maximum } xy &= 2_1 2_2 \dots 2_{p-1} \\ \text{Minimum } xy &= 2 \\ \text{Minimum } z &= 2_p 12_1 2_2 \dots 2_{p+2} \end{aligned}$$

Then every possible decomposition for such w can be represented as:

$$\begin{aligned} x &= 2^r & r &\geq 0 \\ y &= 2^s & s &\geq 1 \\ z &= 2^t 12^{p+2} & t &\geq 1 \end{aligned}$$

where at the same time it holds:

$$r + s + t + 1 = p + (p + 2) \Leftrightarrow r + t = 2p + 1 - s$$

Then always when choosing $k = 0$, we get:

$$xy^0z = 2^r(2^s)^02^t12^{p+2} = 2^r2^t12^{p+2},$$

For this string to be in the language $\overline{L_2}$, it must hold that:

$$r + t + 1 = 2p + 2 - s \Rightarrow 2p + 1 - s + t + 1 = 2p + 2 \Rightarrow s = 0,$$

which is a contradiction with $s \geq 1$

This means that for no decomposition and for $k = 0$, the resulting string will ever be in the language $\overline{L_2}$. That is, the pumping property does not hold for this language, and thus this language is not regular, and therefore L_2 is not regular! \square