

The pumping lemma

We have learned four types of devices for defining formal languages, and we have proved that all of them are able to describe exactly same languages, called **regular languages**.

We have procedures for converting any device into an equivalent device of any other type. The conversion procedures are mechanical algorithms in the sense that one can write a computer program that performs any conversion.

We say the devices are **effectively** equivalent.

It is usually fairly straightforward to prove that some language L is regular: one only needs to design a finite automaton (DFA, NFA or ε -NFA) or a regular expression, and to show that it defines language L .

But what about the **opposite problem**: How does one show that some language L is not regular? One needs to show that there does not exist any device that defines L , but one cannot try all DFA one-by-one because there are infinitely many of them.

(In general, proving that something cannot be done is harder than proving that it can be done, and that makes the negative question more interesting.)

Example. The language

$$L = \{a^n b^n \mid n \geq 0\}$$

contains all words that begin with any number of a 's, followed by equally many b 's. Let us show that there does not exist any DFA that recognizes L .

Example. The language

$$L = \{a^n b^n \mid n \geq 0\}$$

contains all words that begin with any number of a 's, followed by equally many b 's. Let us show that there does not exist any DFA that recognizes L .

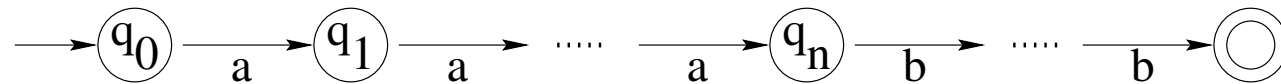
Informally: A DFA that accepts L should first count the number of a 's in the beginning of the word. But a DFA has only finitely many states, so it will get confused: Some a^i and a^j take the DFA to the same state, and then it can no longer remember whether it saw i or j letters a . Since the DFA accepts input word $a^i b^i$, it also accepts input word $a^j b^i$, which is not in the language. So the DFA works incorrectly.

Example. $L = \{a^n b^n \mid n \geq 0\}$

More precisely: Assume that there exists a DFA

$$A = (Q, \Sigma, \delta, q_0, F)$$

such that $L = L(A)$. Let n be the number of states in Q . Consider the accepting path for the input word $a^n b^n$.

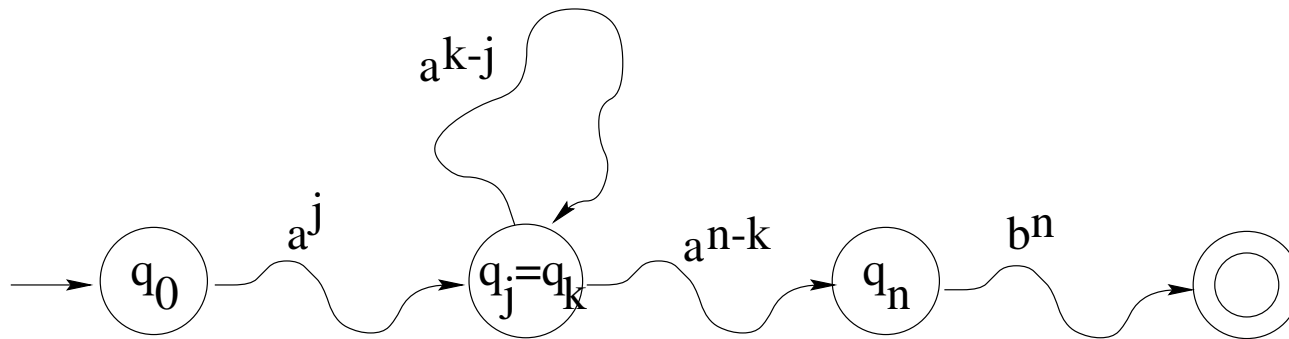


Let q_j be the state of A after reading the first j input letters a .

Example. $L = \{a^n b^n \mid n \geq 0\}$

There are $n + 1$ states $q_0, q_1, \dots, q_n$ but the DFA has only n different states, so two of the states must be identical. (This is the **pigeon hole principle**: if you have more pigeons than holes then two or more pigeons have to share a hole.)

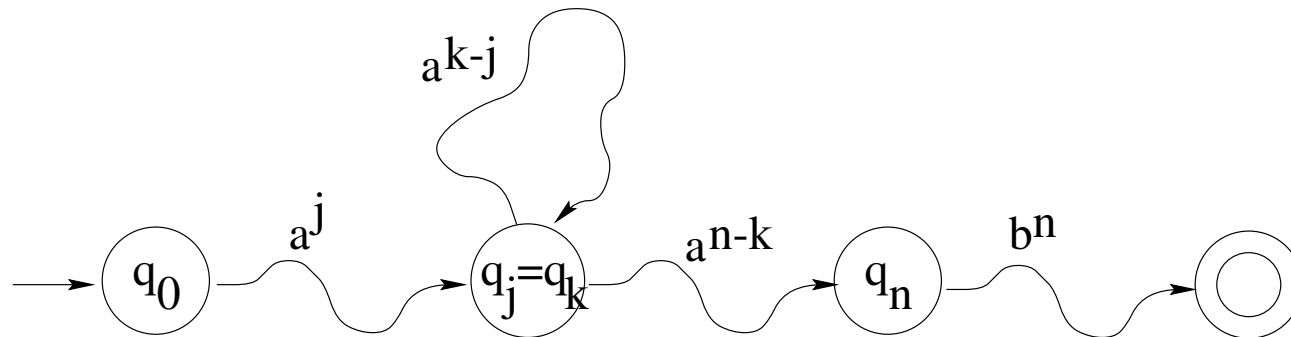
Let q_j and q_k be two identical states, where $j < k$:



Example. $L = \{a^n b^n \mid n \geq 0\}$

There are $n + 1$ states $q_0, q_1, \dots, q_n$ but the DFA has only n different states, so two of the states must be identical. (This is the **pigeon hole principle**: if you have more pigeons than holes then two or more pigeons have to share a hole.)

Let q_j and q_k be two identical states, where $j < k$:



The word a^{k-j} **loops** the DFA at state $q_j = q_k$. The loop can be repeated arbitrarily many times, always getting an accepting computation. For example, if we repeat the loop 0 times we have accepting computation for the word

$$a^j a^{n-k} b^n = a^{n-(k-j)} b^n.$$

This word is not in language L because $k - j \neq 0$. Therefore the DFA A is **not correct**: the language it recognizes is not L .

Similar **”confused in counting”** argument works with many other languages as well. Instead of always repeating the argument like above, we formulate the argumentation as a theorem known as the **pumping lemma**. (Word ”pumping” refers to the fact that the loop can be repeated, or ”pumped”, arbitrarily many times.)

Pumping lemma states a property that every regular language has. If a language does not have that property then the language is not regular.

Pumping lemma. Let L be a regular language. Then there exists some positive constant n such that every word z of length n or greater that belongs to language L can be divided into three segments

$$z = uvw$$

in such a way that

$$\begin{cases} |uv| \leq n, \text{ and} \\ v \neq \varepsilon, \end{cases} \quad (*)$$

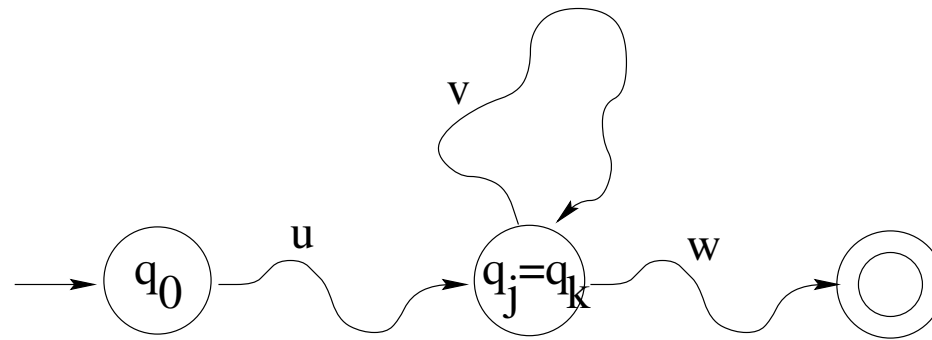
and for all $i \geq 0$ the word $uv^i w$ is in language L .

(So every long enough word in L contains a non-empty subword that can be pumped arbitrarily many times, and the result always belongs to L .)

Proof of the pumping lemma. Let L be a regular language. Then it is accepted by some DFA A . Let n be the number of states in A .

Consider an arbitrary word $z \in L$ such that $|z| \geq n$. We have to show how to divide z into three segments in such a way that $(*)$ is satisfied.

We argue as follows: Let q_j be the state of the machine after the first j input letters of z have been read. The machine has n states so the pigeon hole principle tells that there must exist two identical states among $q_0, q_1, \dots, q_n$, say $q_j = q_k$ for some $j < k \leq n$:



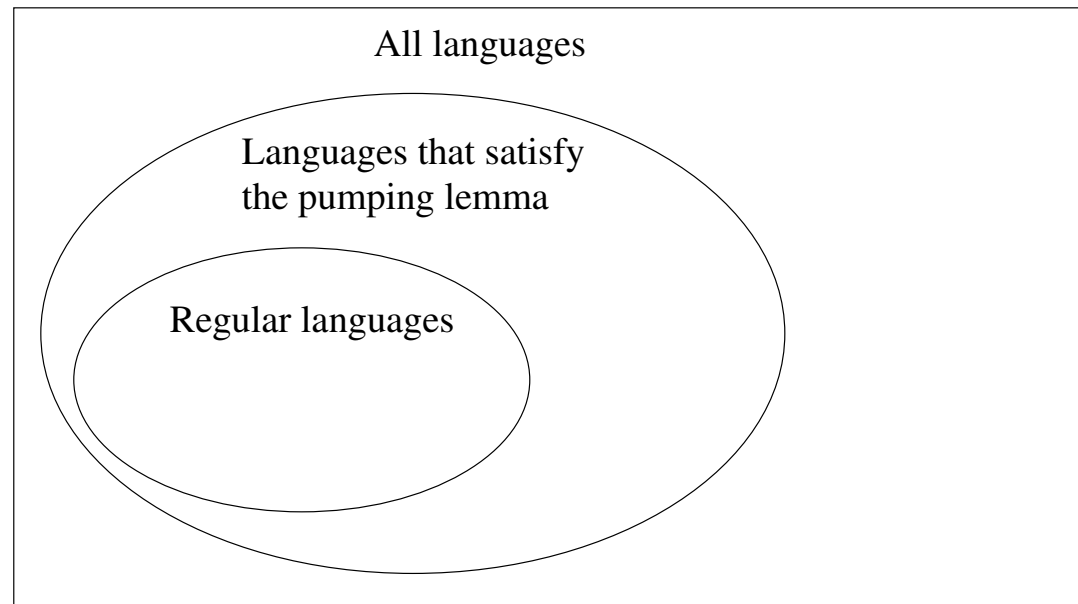
We divide the input word z into three segments in a natural way: u consists of the first j letters, v of the following $k - j$ letters, and w is the remaining tail. This division satisfies $(*)$:

$$|uv| = k \leq n, \text{ and} \\ v \neq \varepsilon \text{ because } |v| = k - j > 0.$$

The loop can be repeated i times for any i , so $uv^i w \in L$ for all $i \geq 0$.

The pumping lemma gives a property satisfied by every regular language. Therefore, if language L does not satisfy the pumping lemma then L is not regular. We can use the pumping lemma to show that certain languages are not regular.

We can **not** use the pumping lemma to prove that a language L is regular. The property is not "if and only if": There are some non-regular languages that satisfy the pumping lemma.



If you show that L satisfies the pumping lemma then language L can still be non-regular. But if you show that L does not satisfy the pumping lemma then L is guaranteed not to be regular.

So we are more interested in the negation of the pumping lemma. The pumping lemma says that if L is regular then

$$(\exists n)(\forall z \dots)(\exists u, v, w \dots)(\forall i)uv^i w \in L.$$

Therefore if L satisfies the opposite statement

$$(\forall n)(\exists z \dots)(\forall u, v, w \dots)(\exists i)uv^i w \notin L$$

then L is not regular.

So we are more interested in the negation of the pumping lemma. The pumping lemma says that if L is regular then

$$(\exists n)(\forall z \dots)(\exists u, v, w \dots)(\forall i)uv^i w \in L.$$

Therefore if L satisfies the opposite statement

$$(\forall n)(\exists z \dots)(\forall u, v, w \dots)(\exists i)uv^i w \notin L$$

then L is not regular.

This is what you do to show that L is not regular:

- (1) For **every** n , select a word $z \in L$, $|z| \geq n$.
- (2) Show that for **every** division $z = uvw$ where $|uv| \leq n$ and $v \neq \varepsilon$ **there exists** a number i such that

$$uv^i w$$

is **not** in language L .

If you can do (1) and (2) then L does not satisfy the pumping lemma, and L is not regular.

Example. Consider again $L = \{a^n b^n \mid n \geq 0\}$.

(1) For any given n select $z = a^n b^n$. The choice is good since $z \in L$ and $|z| \geq n$.

(2) Consider an arbitrary division of z into three part $z = uvw$ where

$$|uv| \leq n \text{ and } v \neq \varepsilon.$$

Then necessarily $u = a^j$ and $uv = a^k$ for some j and k , and $j < k$.

Choosing $i = 0$ gives

$$uv^i w = uw = a^{n-(k-j)} b^n$$

which does not belong to L .

Example 2. Let us prove that

$$L = \{a^{m^2} \mid m \geq 1\} = \{a, a^4, a^9, a^{16}, \dots\}$$

is not regular.

Example 2. Let us prove that

$$L = \{a^{m^2} \mid m \geq 1\} = \{a, a^4, a^9, a^{16}, \dots\}$$

is not regular.

(1) For given n let us choose

$$z = a^{n^2}.$$

Example 2. Let us prove that

$$L = \{a^{m^2} \mid m \geq 1\} = \{a, a^4, a^9, a^{16}, \dots\}$$

is not regular.

(1) For given n let us choose

$$z = a^{n^2}.$$

(2) Consider an arbitrary division $z = uvw$ that satisfies

$$|uv| \leq n \text{ and } v \neq \varepsilon.$$

Let k denote the length of v . It follows from that $0 < k \leq n$.

Pump v with $i =$

Example 2. Let us prove that

$$L = \{a^{m^2} \mid m \geq 1\} = \{a, a^4, a^9, a^{16}, \dots\}$$

is not regular.

(1) For given n let us choose

$$z = a^{n^2}.$$

(2) Consider an arbitrary division $z = uvw$ that satisfies

$$|uv| \leq n \text{ and } v \neq \varepsilon.$$

Let k denote the length of v . It follows from that $0 < k \leq n$.

Pump v with $i = 2$. Then

$$uv^i w = uvvw = a^{n^2+k}$$

Because

$$n^2 = n^2 + 0 < n^2 + k \leq n^2 + n < (n+1)^2$$

number $n^2 + k$ is not a square of any integer. (It falls between two consecutive squares.) Therefore uv^2w does not belong to L .

Some more examples:

1. Is the language $L = \{ww \mid w \in \{a, b\}^*\}$ regular ?

2. Is the language $L = \{ww \mid w \in \{a\}^*\}$ regular ?

3. Is the language $L = \{a^p \mid p \text{ is a prime number}\}$ regular ?