Non-regular Languages and the Pumping Lemma

CMPU 240 • Language Theory and Computation • Spring 2019

Previously:
Introduced regular languages and three equivalent models of computation for recognizing them (DFAs, NFAs, NFA-εs, REs)

Today:
Use the Pumping Lemma to show some languages are not regular

Upcoming:
Quiz on Thursday
Exam 1, next week

Not every language is a regular language.

To prove a language is regular, we can construct a DFA, NFA, or regular expression for it.

Or use other closure properties! More on that next class.

To prove a language is not regular, we need to show that it’s not possible to construct a DFA, NFA, or regular expression for it.

This kind of argument is challenging – how can we show that we wouldn’t be able to devise a finite automaton for it if we tried harder?

A simple language

Let $\Sigma = \{a, b\}$ and consider this language:

$L = \{a^ib^i \mid i \in \mathbb{N}\}$

$L$ is the language of all strings of $i$ as followed by $i$ bs:

$\{\varepsilon, ab, aabb, aaabbb, \ldots\}$

Is this language regular?
How many states are needed to recognize \( \{ \text{a} \text{b} \} \)?

This language is not regular!

Intuitive explanation:
Imagine a finite automaton to accept this language. The number of as must be equal to the number of bs, so it must have some way to remember how many as were seen, and accept if the rest of the string contains the same number of bs.

Intuition
The automaton below has \( n \) states and no loops. Expressed in terms of \( n \), what’s the longest string this automaton can accept?

Since the model of computation has a finite description, for sufficiently long strings, it must repeat some step of the computation, i.e., there must be a cycle in the path from the start state to an accept state.

Because the cycle may be repeated any number of times, the language of any FA with a loop is infinite.
If we have a string \( w \in L \) that is “sufficiently long”, then we can split the string into three pieces and “pump” the middle.

We can write \( w = xyz \) such that \( xy^0z, xy^1z, xy^2z, \ldots \) are all in \( L \).

**Proof of the Pumping Lemma**

If we claim \( L \) is regular, there must be a DFA \( A \) such that \( L = L(A) \).

Let \( A \) have \( n \) states; choose this \( n \) for use as the constant in the Pumping Lemma.

Let \( w \) be a string in \( L \) of length \( \geq n \); say \( w = a_1a_2 \ldots a_m \), where \( m \geq n \).

Let \( q_i \) be the state \( A \) is in after reading the first \( i \) symbols of \( w \).

\( q_0 = \text{start state}, \)
\( q_1 = \delta(q_0, a_1), \)
\( q_2 = \delta(q_0, a_1a_2), \) etc.

**The Pumping Lemma for regular languages**

For every regular language \( L \), there exists an integer \( n \) such that for every string \( w \in L \) of length \( n \) or more, there exist strings \( x, y, \) and \( z \) such that

\[
\begin{align*}
    w &= xyz \\
    |xy| &\leq n \\
    y &\neq \epsilon \\
    xy^iz &\in L \text{ for all } i \geq 0
\end{align*}
\]

Since there are only \( n \) different states, two of \( q_0, q_1, \ldots, q_n \) must be the same; say \( q_i = q_j \), where \( 0 \leq i < j \leq n \).

Let \( x = a_1 \ldots a_i \)
\( y = a_{i+1} \ldots a_j \)
\( z = a_{j+1} \ldots a_m \)

Then by repeating the loop from \( q_i \) to \( q_j \) with label \( a_{i+1} \ldots a_j \) zero or more times, we can show that \( xy^iz \) is accepted by \( A \).
Example

DFA with 4 states that accepts an infinite language:

Any string of length 4 or more contains a circuit.

\[
\begin{align*}
\text{aabb} \\
\text{aaaba} \\
\text{...}
\end{align*}
\]

Rationale for requirements in the Pumping Lemma

0 < |y|

Because y labels the loop, and it has to consist of at least one symbol.

|xy| ≤ n

Because xy is what you get when you take the loop once.

For all \(i \geq 0\), \(xy^iz\) is also in \(L\)

Because y can be pumped zero or more times.

Note that the string \(xz\) is in the language, because it is \(xy^iz\).
The Pumping Lemma gets its name because the repeated string is “pumped”.

Note that because of the nature of FAs, we cannot control the number of times it is pumped.
So, a regular language with strings of length \( \geq n \) is always infinite!

The Pumping Lemma is only interesting for infinite languages.

But it works for finite languages, which are always regular.
For finite languages, \( n \) is larger than the longest string, so nothing can be pumped.

Use of the Pumping Lemma

We use it to show a language \( L \) is not regular.

Start by assuming \( L \) is regular.
Then there must be some \( n \) that serves as the PL constant
\( n \) is the number of states in the automaton
We may not know what \( n \) is, but we can work the rest of the proof with \( n \) as a parameter
We choose some \( w \) known to be in \( L \), with length \( \geq n \)
Typically, \( w \) depends on \( n \), so that we know that its length is \( \geq n \).

The Pumping Lemma “‘game’”

You win the Pumping Lemma game by finding a contradiction of the Pumping Lemma for the given language.

You get to choose the string and the number of times to pump that string up or down to produce a string that isn’t in the language.

Your opponent tries to foil you by selecting decompositions of your string into \( x, y, \) and \( z \) that would let it be pumped.

Applying the Pumping Lemma, we know \( w \) can be broken into \( xyz \), satisfying the properties in the definition of the lemma.
We may not know how to decompose \( w \), so we use \( x, y, \) and \( z \) as parameters.
We derive a contradiction by picking \( i \) – which might depend on \( n, x, y, \) and/or \( z \) – such that \( xy^iz \) is not in \( L \).
Gameplay Magazine described the rules as “punishingly intricate”.

**Four steps**

1. The number of states in the automaton is \( n \). Note that we don’t have to know what \( n \) is; we’ll use it as a variable.
2. Given \( n \), we pick a string \( w \in L \) of length \( \geq n \).
   We usually define the string in terms of \( n \).
3. Our opponent chooses the decomposition of \( w \) into \( xyz \), subject to \( |xy| \leq n \), \( |y| \geq 1 \).
4. We try to pick \( i \) (the power factor in \( xy^iz \)) in such a way that the pumped string \( w^i \) is not in \( L \).
   If we can do so, we win the game!

**Critical point**

It’s necessary to show there is *no segmentation* of the chosen string that won’t lead to a contradiction.

This means considering *every possible* mapping of \( xy \) onto the first \( n \) symbols in the chosen string.

We chose our string to make this easy, since every possible segmentation consists of *as* only.

Pumping therefore disrupts the equivalence of the number of *as* and *bs*.

**Critical point**

We only need to show that there’s *one string* in the language for which the Pumping Lemma doesn’t work.

For some strings in \( L \), it may work perfectly well!
Example: $L = \{a^ib^i\}$ is not regular

Suppose $L$ is regular.

Then there is a constant $n$ satisfying the Pumping Lemma conditions.

Consider the string $w = a^nb^n$.

$|w|$ is obviously $\geq n$, so the PL applies.

Then we can break this string into $w = xyz$, where $|xy| \leq n$ and $y \neq \varepsilon$, and for any $i \geq 0$, the string $xyz$ is also in $L$.

Because $|xy| \leq n$ and $|y| > 0$, the string $y$ has to consist only of $a$s.

So, no matter what segment of the string $xy$ covers, pumping $y$ adds to the number of $a$s, hence there are more $a$s than $b$s.

There is no way to segment $w$ into $xyz$ that can’t be pumped to produce a string that isn’t in the language.

Contradiction! Therefore, $L$ is not regular.

Example: $L = \{ww^R\}$ is not regular

We use $w^R$ to denote $w$ reversed. Let $\Sigma = \{a, b\}$.

Whatever $n$ is, we can always choose a $w$ as follows:

$|n n n n| - a...ab...bb...ba...a$

$\underline{x y z}$

Because of this choice and the requirement that $|xy| \leq n$, the opponent is forced to choose a $y$ that consists entirely of $a$s.

In Step 4, we use $i = 2$; the string $xy^2z$ has more $a$s on the left than on the right, so it cannot be of form $ww^R$.

Therefore $L$ is not regular!

Example

Consider the alphabet $\Sigma = \{0, 1\}$ and the language

$BALANCE = \{w \mid w \text{ has an equal number of 1s and 0s}\}$

E.g.,

$01 \in BALANCE$

$110010 \in BALANCE$

$11011 \notin BALANCE$

Is $BALANCE$ a regular language?

An incorrect proof

Theorem: $BALANCE$ is regular.

Proof: We show that $BALANCE$ satisfies the condition of the Pumping Lemma. Let $n = 2$ and consider any string $w \in BALANCE$ such that $|w| \geq 2$.

Then we can write $w = xyz$ such that $x = z = \varepsilon$ and $y = w$, so $y \neq \varepsilon$. Then for any natural number $i, xy^iz = w^i$, which has the same number of 0s and 1s.

Since $BALANCE$ passes the conditions of the Pumping Lemma, $BALANCE$ is regular.
For every regular language \( L \),
there exists an integer \( n \) such that
for every string \( w \in L \) of length \( n \) or more,
there exist strings \( x, y, \) and \( z \) such that
\[
w = xyz
\]
\[
|xy| \leq n
\]
\[
y \neq \varepsilon
\]
\[
xyz \in L \text{ for all } i \geq 0
\]

Caution with the Pumping Lemma

The Pumping Lemma describes a necessary condition of regular languages.

If \( L \) is regular, \( L \) passes the conditions of the Pumping Lemma.

The Pumping Lemma is not a sufficient condition to be a regular language.

If \( L \) is not regular, it still might pass the conditions of the Pumping Lemma!

Example: \( L = \{w \mid w \text{ has an equal number of } 1\text{s and } 0\text{s}\} \) is not regular

Given \( n \), we choose the string \((01)^n\).

We need to show splitting this string into \( xyz \) where \( xyz \) is in \( L \) is impossible…

But it is possible!

If \( x = \varepsilon, y = 01, \) and \( z = (01)^{n-1}, \)
\( xyz \) is in \( L \) for every value of \( i \)

Are we out of luck?

When using the Pumping Lemma:

If your string does not succeed, try another!
Let's try $1^n0^n$.

Again, we need to show splitting this string into $xyz$ where $xyz$ is in $L$ is impossible...

**But it is possible!**

If $x$ and $z$ are $\varepsilon$ and $y$ is $1^n0^n$, then $xy'z$ always has an equal number of $0$s and $1$s.

Are we still in trouble?

---

**Not this time…**

The Pumping Lemma says that our string has to be divided so that $|xy| \leq n$ and $|y| > 0$.

If $|xy| \leq n$, then $y$ must consist only of $1$s, so $xyyz \not\in L$.

**Contradiction! We win!**

---

**Example:**

$L = \{ww \mid w \in \Sigma^*\}$ is not regular

We choose the string $a^nb^na^nb$, where $n$ is the number of states in the FA.

We now show that there is no decomposition of this string into $xyz$ where for any $i \geq 0$, $xy^iz$ is in $L$.

Since $|xy| \leq n$, it’s easy to show that the Pumping Lemma won’t hold for $L$ because $y$ must consist only of $a$s, so $xyyz$ is not in $L$.

In the previous example, as before, the choice of string is critical.

Had we chosen $a^na^n$ (which is a member of $L$) instead of $a^nb^na^nb$, it wouldn’t work because it can be pumped and still satisfy the Pumping Lemma.

**Moral:** Choose your strings wisely.
Example:
$L = \{0^i1^j \mid i > j\}$ is not regular

Given $n$, choose $s = 0^n1^n$

If $L$ is regular, we can split $s$ into $xyz$ where for any $i \geq 0$, $xy^iz$ is in $L$, $|xy| \leq n$, and $|y| > 0$

Because $|xy| \leq n$, $y$ consists only of 0s

Is $xyz$ in $L$?

The Pumping Lemma states that $xy^iz$ is in $L$ even when $i = 0$

So, consider the string $xy^0z$

Removing string $y$ decreases the number of 0s in $s$

$s$ has only one more 0 than 1

Therefore, $xz$ cannot have more 0s than 1s, and is not a member of $L$

Contradiction!

This strategy is called “pumping down”

Example:
$L = \{a^i \mid i \text{ is prime}\}$ is not regular

Let $n$ be the Pumping Lemma value and let $k$ be a prime greater than $n$

If $L$ is regular, the Pumping Lemma states that $a^k$ can be decomposed into $xyz$, $|y| > 0$, such that $xy^iz$ is in $L$ for all $i \geq 0$

Assume such a decomposition exists

Then if $w$ is in $L$, the length of $w = xy^{k+1}z$ must be a prime. However,

$$\text{length}(xy^{k+1}z) = \text{length}(xyz)$$

$$= k + k(\text{length}(y))$$

The length of $xy^{k+1}z$ is therefore not prime, since it is the product of two numbers other than 1. So $xy^{k+1}z$ is not in $L$.

Contradiction!

Example:
$L = \{a^i \mid i \text{ is prime}\}$ is not regular

Let $n$ be the Pumping Lemma constant, and let $k$ be a prime greater than $n$.

If $L$ is regular, the Pumping Lemma states that $a^k$ can be decomposed into $xyz$, $|y| > 0$, such that $xyz$ is in $L$ for all $i \geq 0$. Assume such a decomposition exists.

Then if $w$ is in $L$, the length of $w = xy^kz$ must be a prime. However,

$$\text{length}(xy^kz) = \text{length}(xyz)$$

$$= k + k(\text{length}(y))$$

The length of $xy^kz$ is therefore not prime, since it is the product of two numbers other than 1. So $xy^kz$ is not in $L$.

Contradiction!
Remember

You only need to find one string for which the Pumping Lemma does not hold to prove a language is not regular.

But you must show that for any decomposition of that string into $xyz$ the Pumping Lemma holds.

This sometimes means considering several different cases.

Acknowledgments

This lecture incorporates material from:

- Nancy Ide
- Keith Schwarz
- Michael Sipser
- Jeffrey Ullman