Non-regular Languages

Assignments:
Assignment 2 due today.
Assignment 3 out Thursday.

Previously:
Introduced the Pumping Lemma to show some languages are not regular.

Today:
Other approaches to show that languages are regular or non-regular.

The Pumping Lemma

If $L$ is a regular language, then there exists a constant $n$ such that every string $w$ in $L$, of length $n$ or more, can be written as $w = xyz$, where:

- $0 < |y|$
- $|xy| \leq n$
For all $i \geq 0$, $xy^iz$ is also in $L$

Note: $y^i = y$ repeated $i$ times; $y^0 = \varepsilon$.

Proving a language non-regular without the pumping lemma

The pumping lemma isn’t the only way we can prove a language is non-regular.

Other techniques:
Show that the desired DFA would require infinite states to model the intended language
Use closure properties to relate to other non-RL languages
DFA Method

Consider the language \( \{a^ib^i \mid i \geq 0\} \) and a DFA to recognize it.

For any \( i \), let \( a_i \) be the state entered after processing \( a^i \), i.e.,
\[
\hat{\delta}(q_0, a^i) = a_i
\]

Consider any \( i \) and \( j \) such that \( i \neq j \)

\[
\hat{\delta}(q_0, ab^i) \neq \hat{\delta}(q_0, a^ib^i), \text{ since the former is accepting, and the latter is rejecting}
\]

\[
\hat{\delta}(q_0, a^ib^i) = \hat{\delta}(\hat{\delta}(q_0, a^i), b^i) = \hat{\delta}(a_i, b^i), \text{ by definition of } \hat{\delta} \text{ and definition of } a_i, \text{ respectively}
\]

\[
\hat{\delta}(q_0, ab^i) = \hat{\delta}(\hat{\delta}(q_0, a), b^i) = \hat{\delta}(a, b^i), \text{ by the same reasoning}
\]

Closure properties

Certain operations on regular languages are guaranteed to produce regular languages.

Closure properties can also be used to prove a language is non-regular (or regular).

Regular languages are closed under common set operations

**Union:** \( L_1 \cup L_2 \)

**Intersection:** \( L_1 \cap L_2 \)

**Concatenation:** \( L_1L_2 \)

**Complementation:** \( \overline{L_1} \)

**Star-closure:** \( L_1^* \)
Other closures

**Difference:** If $L_1$ and $L_2$ are regular, then $L_1 - L_2$ is also regular.

*Proof:*
Set difference is defined as $L_1 - L_2 = L_1 \cap \overline{L_2}$

We know that if $L_2$ is regular, so is $\overline{L_2}$. We also know regular languages are closed under intersection. Therefore, we know that $L_1 \cap \overline{L_2}$ is regular.

*Difference is sometimes notated as $L_1 \setminus L_2$*

---

Other closures

**Reversal:** If $L_1$ is regular, then $L^R$ is also regular.

*Proof:*
Suppose $L$ is a regular language. We can therefore construct an NFA with a single final state that accepts $L$. We can then make the start state of this NFA the final state, make the final state the start state, and reverse the direction of all arcs in the NFA. The modified NFA accepts a string $w^R$ if and only if the original NFA accepts $w$.

Therefore the modified NFA accepts $L^R$.

---

Using regular language closure properties

**Show a language is regular**

Show that by using two or more known regular languages and one or more of the operations over which regular languages are closed, you can produce that language.

**Basic template**

$L_{\text{REG}_1} \ [\text{OP}] \ L_{\text{REG}_2} = L_{\text{REG}_3}$

where OP is one of the operations over which regular languages are closed

$L_{\text{REG}_3}$ is the language we need to prove is regular

$L_{\text{REG}_1}$ and $L_{\text{REG}_2}$ are known regular languages

If the two languages on the left side of the operator are regular, then so too must be the one on the right side.

**NB:** Cannot assume that if the language on the right is regular, so too must be both languages on the left!
**Example**

If $L$ is a regular language, is $L_1 = \{uv \mid u \in L, \vert v \vert = 2\}$ also regular?

We know $L$ is regular.

Every string in $L_1$ consists of a string from $L$ concatenated to a string of length 2.

The set of strings of length 2 (call it $L_2$) over any alphabet is finite, and therefore this is a regular language since all finite languages are regular.

Therefore we have

$$L \ [\text{concatenation}] \ L_2 = L_1$$

$$\left[ L_{REG_1} \right] \ [\text{OP}] \ [ L_{REG_2} ] = \ [ L_{REG_3} ]$$

Since $L_1$ is the concatenation of two regular languages, $L_1$ must also be regular.

---

**Example**

Prove the language $\{a^n b^m \mid n > 3 \text{ or } m > 3\}$ is regular.

Show that this language can be produced using regular language closure properties on known regular languages $L_1 = \{a^n b^n\}$, $L_2 = \{\varepsilon, a, aa, aaa\}$, $L_3 = \{\varepsilon, b, bb, bbb\}$ as follows:

- **concatenation:**
  $$L_{4} = L_2 \ L_3$$

- **complementation:**
  $$L_5 = L_4$$

- **intersection:**
  $$L_6 = L_5 \cap L_1$$

$$= \{a^n b^m \mid n > 3 \text{ or } m > 3\}$$

However:

the language in question is plugged into the template in the position of $L_{REG_1}$

want to use a known regular language for $L_{REG_2}$

If we can show that $L_{REG_3}$ is not regular, then it must be the case that $L_{REG_1}$ is not regular.

---

**Example**

Prove $\{a^n b^m \mid n > 3 \text{ and } m > 3\}$ is regular.

Show that this language can be produced using regular language closure properties on known regular languages $L_1 = \{a^n b^n\}$, $L_2 = \{\varepsilon, a, aa, aaa\}$, $L_3 = \{\varepsilon, b, bb, bbb\}$ as follows:

- **concatenation:**
  $$L_{a} = a^* L_3$$

- **concatenation:**
  $$L_{b} = L_2 b^*$$

- **union:**
  $$L_4 = L_{a} \cup L_{b}$$

- **complementation:**
  $$L_{5} = L_4$$

- **intersection:**
  $$L_{6} = L_{5} \cap L_1$$

$$= \{a^n b^m \mid n > 3 \text{ and } m > 3\}$$

---

**Using regular language closure properties**

**Show a language is not regular**

Use the same template$$L_{REG_1} \ [\text{OP}] \ L_{REG_2} = L_{REG_3}$$

However:

the language in question is plugged into the template in the position of $L_{REG_1}$

want to use a known regular language for $L_{REG_2}$

If we can show that $L_{REG_3}$ is not regular, then it must be the case that $L_{REG_1}$ is not regular.
Example

Show \( L = \{w \mid w \text{ in } \{a,b\}^* \mid \text{w has equal number of } a \text{ and } b\} \) is non-regular.

Use the template \( L \cap a^*b^* = \{a^n b^n \mid n \geq 0\} \)

If both languages on the left side of the “=” are regular, the language on the right side is regular (closure of regular languages over intersection).

\( \{a^n b^n \mid n \geq 0\} \) is easily proved non-regular using the Pumping Lemma

We know \( a^*b^* \) is regular.

Therefore \( L \) must be non-regular.

Example

Use same strategy as previous example:

Given
\[ L_1 \text{ is regular} \]
\[ L_1 \cap L_2 \text{ is regular} \]
\[ L_2 \text{ is non-regular} \]

Is \( L_1 \cup L_2 \) regular?

Fill the template:

Use one of our known regular languages on the left side.

Use the known non-regular language on the right side.

So we have
\[ (L_1 \cup L_2) \text{ OP} [\text{known reg. lang.}] = L_2 \]

Need to put in an operation and a known regular language that we know yields \( L_2 \)

To do this, we need to get \( L_2 \) isolated from \( L_1 \cap L_2 \)

Can’t “extract” \( L_2 \) from \( L_1 \cup L_2 \) using only \( L_1 \) or \( L_1 \cap L_2 \), since taking the difference of \( L_1 \cup L_2 \) and \( L_1 \) gives us only what is left of \( L_2 \) that is not in \( L_1 \).

Have to “remove” anything that is in \( L_1 \cap L_2 \) from \( L_1 \), then subtract the result (everything in \( L_1 \) that is not also in \( L_2 \)) from \( L_1 \cup L_2 \)

Difference and union are closed for regular languages

After doing this we know we still have a regular language to subtract from \( L_1 \cup L_2 \).

Result:
\[ (L_1 \cup L_2) \setminus (L_1 \setminus (L_1 \cap L_2)) = L_2 \]
Template

\[ L_{\text{REG}1} [\text{OP}] L_{\text{REG}2} = L_{\text{REG}3} \]
\[(L_1 \cup L_2) - (L_1 - (L_1 \cap L_2)) = L_2 \]

We know

\( L_{\text{REG}2} \) is regular
produced by applying the closure operations on two known regular languages \( L_1, \) and \( L_1 \cap L_2, \)
if \( L_{\text{REG}1} \) is regular, so is \( L_{\text{REG}3} \)

But we were given the fact that \( L_2 \) is non-regular

We can conclude that \( L_{\text{REG}1} = L_1 \cup L_2 \) is non-regular as well.

Using closure properties with Pumping Lemma

Show that \( L = \{w \mid w \text{ has more instances of substring } aa \text{ than substring } bb\} \) is non-regular.

Intersect with \( b^m a^p \) where \( p > m \).

Consider \( b^m a^{m+1} \).

If \( L \) is regular, the conditions of the Pumping Lemma hold:
\( xy \) spans at most the first \( n \) bs and \( y \) consists of 1 to \( n \) bs.

However, regardless of how long \( y \) is, when pumped it adds at least one \( b \) and the resulting string contains at least as many \( bs \) as as so is not in \( L \).

Divide and Conquer

\( L = \{w \in \{a,b\}^* \mid w \text{ contains an even number of } a \text{ s and an odd number of } b \text{ s and all } a \text{ s come in runs of three}\}\)

Regular or non-regular?

Regular: \( L = L_1 \cap L_2, \) where

\( L_1 = \{w \in \{a,b\}^* \mid w \text{ contains an even number of } a \text{ s and an odd number of } b \text{ s}\} \) and

\( L_2 = \{w \in \{a,b\}^* \mid \text{all } a \text{ s come in runs of three}\} \)

To prove it, build an FSA for each

Easier than FSA for the original language
What the Closure Theorem for Union Does Not Say

Closure theorem for union says: If $L_1$ and $L_2$ are regular, then $L = L_1 \cup L_2$ is regular.

What happens if (for example) $L$ is regular? Does that mean that $L_1$ and $L_2$ are also?

**Maybe.**

Example

We know $a^+$ is regular

Consider two cases for $L_1$ and $L_2$:

\[
\begin{align*}
\sigma^+ &= \{\sigma^n \mid n > 0 \text{ and } n \text{ is prime} \} \cup \\
&\quad \{\sigma^n \mid n > 0 \text{ and } n \text{ is not prime} \}
\end{align*}
\]

Neither $L_1$ nor $L_2$ is regular!

\[
\begin{align*}
\sigma^+ &= \{\sigma^n \mid n > 0 \text{ and } n \text{ is even} \} \cup \\
&\quad \{\sigma^n \mid n > 0 \text{ and } n \text{ is odd} \}
\end{align*}
\]

Both $L_1$ and $L_2$ are regular!

What the Closure Theorem for Concatenation Does Not Say

Closure Theorem for Concatenation says: If $L_1$ and $L_2$ are regular, then $L = L_1L_2$ is regular.

What happens (for example) if $L_2$ is not regular? Does that mean that $L$ isn’t regular?

**Maybe.**

Consider two examples:

\[
\begin{align*}
\{\sigma a \sigma \sigma^n \mid n \geq 0\} &= \{ab\} \{\sigma^n \mid n \geq 0\} \\
L &= L_1L_2
\end{align*}
\]

Neither $L$ nor $L_2$ is regular!

\[
\begin{align*}
\{\sigma a a^* \} &= \{a^+\} \{\sigma^n \mid n \text{ is prime} \} \\
L &= L_1L_2
\end{align*}
\]

$L_2$ is not regular, but $L$ is!
True or False?

If \( L_1 \subseteq L_2 \) and \( L_1 \) is not regular, then \( L_2 \) is not regular.

**False!** \( \{a,b\}^* \) is regular, and it has a non-regular subset \( \{a^n b^n \mid n \geq 0\} \)

If \( L_1 \subseteq L_2 \) and \( L_2 \) is not regular, then \( L_1 \) is not regular.

**False!** Non-regular languages have finite subsets, and finite languages are regular

Hints

When you need a known regular language in a proof, remember that \( \Sigma^*, \varepsilon, a^*, a^*b^* \), etc. are regular.

When you need a known non-regular language, use \( a^n b^n \) or any language with a similar dependency.