We say: Regular languages are closed under

- Union: $L_1 \cup L_2$
- Concatenation: $L_1L_2$
- Star: $L_1^*$
- Reversal: $L_1^R$
- Complement: $\overline{L_1}$
- Intersection: $L_1 \cap L_2$

Reverse

NFA for $L_1^R$

1. Reverse all transitions
2. Make initial state final state and vice versa
Example

\[ L_1 = \{a^n b\} \]

\[ L_1^R = \{ba^n\} \]

Intersection

DeMorgan's Law: \[ L_1 \cap L_2 = \overline{L_1} \cup \overline{L_2} \]

\[ L_1 \cap L_2 \] regular

\[ \overline{L_1} \cup \overline{L_2} \] regular

Standard Representations of Regular Languages

- DFAs
- NFAs
- Regular Expressions
- Regular Grammars

Example

\[ L_1 = \{a^n b\} \] regular

\[ L_2 = \{ab, ba\} \] regular

\[ L_1 \cap L_2 = \{ab\} \] regular
When we say: We are given a Regular Language $L$

We mean: Language $L$ is in a standard representation

Elementary Questions about Regular Languages

Membership Question

Question: Given regular language $L$ and string $w$, how can we check if $w \in L$?

Answer: Take the DFA that accepts $L$ and check if $w$ is accepted
Question: Given regular language $L$ how can we check if $L$ is empty: $(L = \emptyset)$?

Answer: Take the DFA that accepts $L$

Check if there is any path from the initial state to a final state

Question: Given regular language $L$ how can we check if $L$ is finite?

Answer: Take the DFA that accepts $L$

Check if there is a walk with cycle from the initial state to a final state

DFA

$L \neq \emptyset$

DFA

$L = \emptyset$

DFA

$L$ is infinite

DFA

$L$ is finite
Question: Given regular languages $L_1$ and $L_2$, how can we check if $L_1 = L_2$?

Answer: Find if $(L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2) = \emptyset$

When we say: We are given a Regular Language $L$

We mean: Language $L$ is in a standard representation
How can we prove that a language $L$ is not regular?

Prove that there is no DFA that accepts $L$.

**Problem:** this is not easy to prove

**Solution:** the Pumping Lemma !!!

---

The Pigeonhole Principle

- $a^n b^n : n \geq 0$
- $\{ vv^R : v \in \{a, b\}^* \}$

Regular languages

- $a^* b$
- $b^* c + a$
- $b + c(a + b)^*$
- etc...

Non-regular languages

- $\{a^n b^n : n \geq 0\}$
- $\{ vv^R : v \in \{a, b\}^* \}$
A pigeonhole must contain at least two pigeons.

The Pigeonhole Principle

There is a pigeonhole with at least 2 pigeons.

The Pigeonhole Principle

and

DFAs
DFA with 4 states

In walks of strings: 
- aabb
- bhaa
- abbbabh
- abbbabhbabbb...

If string $w$ has length $|w| \geq 4$:
- Then the transitions of string $w$ are more than the states of the DFA.
- Thus, a state must be repeated.
In general, for any DFA:

String $w$ has length $\geq$ number of states

A state $q$ must be repeated in the walk of $w$

In other words for a string $w$:

Transitions are pigeons

States are pigeonholes

The Pumping Lemma

Take an infinite regular language $L$

There exists a DFA that accepts $L$
Take string $w$ with $w \in L$.

There is a walk with label $w$:

If string $w$ has length $|w| \geq m$ (number of states of DFA)

then, from the pigeonhole principle:

a state is repeated in the walk $w$

Let $q$ be the first state repeated in the walk of $w$

Write $w = x y z$.
Observations:

\[
\begin{align*}
\text{length } |x y| &\leq m \\
\text{number of states of DFA} & \\
\text{length } |y| &\geq 1
\end{align*}
\]

Observation: The string \(x \ z\) is accepted.

Observation: The string \(x y y z\) is accepted.

Observation: The string \(x y y y z\) is accepted.

Observation: The string \(x y y y z\) is accepted.
In General: The string $x y^i z$ is accepted $i = 0, 1, 2, ...$

In other words, we described:

The Pumping Lemma !!!

The Pumping Lemma:

- Given a infinite regular language $L$
- there exists an integer $m$
- for any string $w \in L$ with length $|w| \geq m$
- we can write $w = x y z$
- with $|x y| \leq m$ and $|y| \geq 1$
- such that: $x y^i z \in L$ $i = 0, 1, 2, ...$

Language accepted by the DFA
Applications of the Pumping Lemma

**Theorem:** The language $L = \{a^n b^n : n \geq 0\}$ is not regular.

**Proof:** Use the Pumping Lemma

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

Assume for contradiction that $L$ is a regular language.

Since $L$ is infinite, we can apply the Pumping Lemma.

Let $m$ be the integer in the Pumping Lemma.

Pick a string $w$ such that: $w \in L, \: |w| \geq m$.

We pick $w = a^m b^m$.
Write: \( a^m b^m = x y z \)

From the Pumping Lemma it must be that length \(|x y| \leq m, \ |y| \geq 1\)

\[
x y z = a^m b^m = a \cdots a a a a a b \cdots b
\]

\( x y z = a^m b^m \quad y = a^k, \ k \geq 1 \)

From the Pumping Lemma: \( x y^i z \in L \)

\( i = 0, 1, 2, \ldots \)

Thus: \( x y^2 z \in L \)

Thus: \( y = a^k, \ k \geq 1 \)

\[
xy^2z = a \cdots a a a \cdots a a \cdots a b \cdots b \in L
\]

\( x y z = a^m b^m \quad y = a^k, \ k \geq 1 \)

From the Pumping Lemma: \( x y^i z \in L \)

\( i = 0, 1, 2, \ldots \)

Thus: \( x y^2 z \in L \)

\[
xy^2z = a \cdots a a a \cdots a a \cdots a b \cdots b \in L
\]

Thus: \( a^{m+k} b^m \in L \)

BUT: \( L = \{ a^n b^n : n \geq 0 \} \)

\( a^{m+k} b^m \notin L \)

CONTRADICTION!!!
Our assumption that $L$ is a regular language is not true.

**Conclusion:** $L$ is not a regular language

Non-regular languages $\{a^n b^n : n \geq 0\}$

Regular languages

---

**What’s Next**

- **Read**
  - Linz Chapter 1, 2.1, 2.2, 2.3, (skip 2.4) 3, and 4
  - JFLAP Startup, Chapter 1, 2.1, 3, 4, 6.1

- **Next Lecture Topics from Chapter 4.3**
  - More Pumping Lemma

- **Quiz 1 in Recitation on Wednesday 9/17**
  - Covers Linz 1.1, 1.2, 2.1, 2.2, 2.3 and JFLAP 1, 2.1
  - Closed book, but you may bring one sheet of 8.5 x 11 inch paper with any notes you like.
  - Quiz will take the full hour

- **Homework**
  - Homework Due Today
  - New Homework Assigned Friday Morning
  - New Homework Due Following Thursday