Regular and Nonregular Languages

\[ a^* b^* \] is regular.
\[ \{a^n b^n : n \geq 0\} \] is not.

\[ \{w \in \{a, b\}^* : \text{every } a \text{ is immediately followed by } b\} \] is regular.

\[ \{w \in \{a, b\}^* : \text{every } a \text{ has a matching } b \text{ somewhere}\} \] is not

Questions:

- Showing that a language is regular.
- Showing that a language is not regular.
Showing that a Language is Regular

**Theorem:** Every finite language is regular.

**Proof:** If $L$ is the empty set, then it is defined by the regular expression $\emptyset$ and so is regular. If it is any finite language composed of the strings $s_1, s_2, \ldots, s_n$ for some positive integer $n$, then it is defined by the regular expression:

$$s_1 \cup s_2 \cup \ldots \cup s_n$$

So it too is regular.
Showing that a Language is Regular

Example:

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

- \( L_1 = \{ a^n b^n, n \geq 0 \} \), and
- \( L_2 = \{ b^n a^n, n \geq 0 \} \)

\( L_1 \) and \( L_2 \) are infinite however

\( L = \{ \varepsilon \} \) is regular
Showing that a Language is Regular

1. Show that \( L \) is finite.

2. Exhibit an FSM for \( L \).

3. Exhibit a regular expression for \( L \).

4. Show that the number of equivalence classes of \( \equiv_L \) is finite.

5. Exhibit a regular grammar for \( L \).

6. Exploit the closure theorems
Closure Properties of Regular Languages

- Union
- Concatenation
- Kleene star
- Complement
- Intersection
- Difference
- Reverse
- Letter substitution
Letter Substitution

- Let $\Sigma_1$ and $\Sigma_2$ be alphabets.

- Let $sub$ be any function from $\Sigma_1$ to $\Sigma_2^*$.

Example:

Let: $\Sigma_1 = \{a, b\}$,
$\Sigma_2 = \{0, 1\}$,

$\text{sub}(a) = 0$, and
$\text{sub}(b) = 11$. 
Letter Substitution

- *letsub* is a letter substitution function iff:

\[
\text{letsub}(L_1) = \{ w \in \Sigma_2^* : \exists y \in L_1 \text{ and } w = y \text{ except that: }
\]
\[
\text{every character } c \text{ of } y \text{ is replaced by } \text{sub}(c) \}. 
\]

Example:

\[
\text{sub}(a) = 0, \text{ and } \quad \text{sub}(b) = 11.
\]

Then \( \text{letsub}(\{a^n b^n, n \geq 0\}) = 0^n 1^{2n} \)
Showing that a Language is Not Regular

Every regular language can be accepted by some FSM.

It can only use a finite amount of memory to record essential properties.

Example:
\[ a^n b^n, \ n \geq 0 \] is not regular
Showing that a Language is Not Regular

The only way to generate/accept an infinite language with a finite description is to use:

- Kleene star (in regular expressions), or
- cycles (in automata).

This forces some kind of simple repetitive cycle within the strings.

Example:

\[ ab^*a \text{ generates } \text{aba, abba, abbbba, abbbba, etc.} \]

Example:

\[ \{a^n : n \geq 1 \text{ is a prime number}\} \text{ is not regular.} \]
Exploiting the Repetitive Property

If an FSM with \( n \) states accepts any string of length \( \geq n \), how many strings does it accept?

\[ L = \text{bab}^*\text{ab} \]

\[
\begin{array}{cccccccc}
\text{b} & \text{a} & \text{b} & \text{b} & \text{b} & \text{b} & \text{a} & \text{b} \\
\text{x} & \text{y} & \text{z}
\end{array}
\]

\( xy^*z \) must be in \( L \).

So \( L \) includes: \( \text{baab}, \text{babab}, \text{babbab}, \text{babbbbbbbbbbab} \)
Theorem – Long Strings

**Theorem:** Let $M = (K, \Sigma, \delta, s, A)$ be any DFSM. If $M$ accepts any string of length $|K|$ or greater, then that string will force $M$ to visit some state more than once (thus traversing at least one loop).

**Proof:** $M$ must start in one of its states. Each time it reads an input character, it visits some state. So, in processing a string of length $n$, $M$ creates a total of $n + 1$ state visits. If $n + 1 > |K|$, then, by the pigeonhole principle, some state must get more than one visit. So, if $n \geq |K|$, then $M$ must visit at least one state more than once.
The Pumping Theorem for Regular Languages

If \( L \) is regular, then every long string in \( L \) is pumpable.

So, \( \exists k \geq 1 \)

\[
(\forall \text{ strings } w \in L, \text{ where } |w| \geq k

(\exists x, y, z \ (w = xyz,
\quad |xy| \leq k,
\quad y \neq \varepsilon, \text{ and}
\quad \forall q \geq 0 \ (xy^qz \text{ is in } L))).
\]
Example: \( \{a^mb^n : n \geq 0\} \) is not Regular

If \( L \) were regular, then there would exist some \( k \) such that any string \( w \) where \( |w| \geq k \) must satisfy the conditions of the theorem. Let \( w = a^{\lceil k/2 \rceil}b^{\lceil k/2 \rceil} \). Since \( |w| \geq k \), \( w \) must satisfy the conditions of the pumping theorem. So, for some \( x, y, \) and \( z \), \( w = xyz \), \( |xy| \leq k \), \( y \neq \varepsilon \), and \( \forall q \geq 0, xy^qz \) is in \( L \). We show that no such \( x, y, \) and \( z \) exist.

There are 3 cases for where \( y \) could occur: We divide \( w \) into two regions:

\[
\begin{align*}
\text{aaaaa.....aaaaaa} & \quad | \quad \text{bbbbbb.....bbbbbb} \\
1 & \quad | \quad 2
\end{align*}
\]

So \( y \) can fall in:

- (1): \( y = a^p \) for some \( p \). Since \( y \neq \varepsilon \), \( p \) must be greater than 0. Let \( q = 2 \). The resulting string is \( a^{k+p}b^k \). But this string is not in \( L \), since it has more \( a \)'s than \( b \)'s.

- (2): \( y = b^p \) for some \( p \). Since \( y \neq \varepsilon \), \( p \) must be greater than 0. Let \( q = 2 \). The resulting string is \( a^k b^{k+p} \). But this string is not in \( L \), since it has more \( b \)'s than \( a \)'s.

- (1, 2): \( y = a^pb^r \) for some non-zero \( p \) and \( r \). Let \( q = 2 \). The resulting string will have interleaved \( a \)'s and \( b \)'s, and so is not in \( L \).

There exists one long string in \( L \) for which no \( x, y, z \) exist. So \( L \) is not regular.
Using the Pumping Theorem

If $L$ is regular, then every “long” string in $L$ is pumpable.

To show that $L$ is not regular, we find one that isn’t.

To use the Pumping Theorem to show that a language $L$ is not regular, we must:

1. Choose a string $w$ where $|w| \geq k$. Since we do not know what $k$ is, we must state $w$ in terms of $k$.
2. Divide the possibilities for $y$ into a set of equivalence classes that can be considered together.
3. For each such class of possible $y$ values where $|xy| \leq k$ and $y \neq \varepsilon$:
   
   Choose a value for $q$ such that $xy^qz$ is not in $L$.  

Example: \( L = \{a^n: n \text{ is prime} \} \)

Let \( w = a^j \), where \( j \) is the smallest prime number > \( k+1 \).

\[ y = a^p \text{ for some } p. \]

\( \forall q \geq 0 \) (\( a|x| + |z| + q\cdot|y| \) must be in \( L \)). So \( |x| + |z| + q\cdot|y| \) must be prime.

But suppose that \( q = |x| + |z| \). Then:

\[
|x| + |z| + q\cdot|y| = |x| + |z| + (|x| + |z|) \cdot y = (|x| + |z|) \cdot (1 + |y|),
\]

which is non-prime if both factors are greater than 1:

\[
(|x| + |z|) > 1 \text{ because } |w| > k+1 \text{ and } |y| \leq k. \\
(1 + |y|) > 1 \text{ because } |y| > 0.
\]
Using the Closure Properties

The two most useful ones are closure under:

Intersection

Complement
Using the Closure Properties

The two most useful ones are closure under: Intersection and Complement

Example:

\[ L = \{ w \in \{a, b\}^*: \#_a(w) = \#_b(w) \} \]

If \( L \) were regular, then:

\[ L' = L \cap a^*b^* \]

would also be regular. But it isn’t.
Using the Closure Properties

$L = \{a^i b^j: i, j \geq 0 \text{ and } i \neq j\}$

Try to use the Pumping Theorem by letting $w = a^{k!} b^{k+k!}$.

Then $y = a^p$ for some nonzero $p$.

Let $q = (k!/p) + 1$ (i.e., pump in $(k!/p)$ times).

Note that $(k!/p)$ must be an integer because $p < k$.

The number of $a$’s in the new string is $k + (k!/p)p = k + k!$.

So the new string is $a^{k+k!} b^{k+k!}$, which has equal numbers of $a$’s and $b$’s and so is not in $L$. 
Using the Closure Properties

\[ L = \{a^i b^j: i, j \geq 0 \text{ and } i \neq j\} \]

An easier way:

If \( L \) is regular then so is \( \overline{L} \). Is it?

\[ \overline{L} = a^n b^n \cup \{\text{out of order}\} \]

If \( \overline{L} \) is regular, then so is \( L' = \overline{L} \cap a^*b^* \)

\[ = a^n b^n \]
Using the Closure Properties

$L = \{a^i b^j c^k : \ i, \ j, \ k \geq 0 \text{ and } (i \neq 1 \text{ or } j = k)\}$

Every string in $L$ of length at least 1 is pumpable:

If $i = 0$ then: if $j \neq 0$, let $y$ be $b$; otherwise, let $y$ be $c$. Pump in or out. Then $i$ will still be 0 and thus not equal to 1, so the resulting string is in $L$.

If $i = 1$ then: let $y$ be $a$. Pump in or out. Then $i$ will no longer equal 1, so the resulting string is in $L$.

If $i = 2$ then: let $y$ be $aa$. Pump in or out. Then $i$ cannot equal 1, so the resulting string is in $L$.

If $i > 2$ then: let $y = a$. Pump out once or in any number of times. Then $i$ cannot equal 1, so the resulting string is in $L$. 
Using the Closure Properties

$L = \{a^i b^j c^k : i, j, k \geq 0 \text{ and } (i \neq 1 \text{ or } j = k)\}$

Suppose we guarantee that $i = 1$. If $L$ is regular, then so is:

$L' = L \cap a^* b^* c^*$.

$L' = \{a^i b^j c^k : j, k \geq 0 \text{ and } j = k\}$

Use Pumping to show that $L'$ is not regular:

OR

If $L$ is regular, then so is $L^R$:

$L^R = \{c^k b^j a^i : i, j, k \geq 0 \text{ and } (i \neq 1 \text{ or } j = k)\}$

Use Pumping to show that $L'$ is not regular: