Unit 6

Non Regular Languages
The Pumping Lemma

Reading: Sipser, chapter 1
Are all languages regular?

• No!
  – Most of the languages are not regular!
• Why?
  – A finite automaton has limited memory.
• How can we show that a language is regular?
• How can we show that a language is not-regular?
Finite Languages

- **Theorem**: If $L$ is finite then $L$ is regular.

- **Proof idea**: 
  - Build an automaton for every word $w$ in $L$: $A(w)$.
  - Add $q_{\text{start}}$ with $\varepsilon$-transitions to every initial state of $A(w)$
  - The resulting automaton recognizes $L$. 
What About Infinite Languages?

• Infinite language could be regular or non-regular

• Example of an infinite regular language:
  \[ L = \{0^n1^m \mid n,m > 0\} \]

• Example of a non-regular language:
  \[ L = \{0^n1^m \mid m > n\} \]
Example of an infinite non-regular language

The language $L = \{0^n1^m \mid m > n\}$ is not regular.

- It is *not regular because* when we reach substring of 1’s we have to remember the length of the 0’s substring.
- The length is remembered by states - one state for each letter.
- The length of the 0's prefix can be any number.
- A finite automaton has a finite number of states. So it cannot remember the length of 0’s prefix.
The Pumping Lemma

- The central theorem concerning regular languages.
- It determines the basic nature of regular languages.
- Any regular language satisfies the Lemma.
- A language that does not satisfy the Lemma must be non-regular.
The Pumping Idea

- Assume an automaton with n states.
- Any word w, |w| \( \geq n \) must include a ‘loop’.
- Denote the part before the loop by x.
- The loop is y.
- The part after the loop is z.
- The z part ends with an accepting state as the word is in L.
The Pumping Idea (cont.)

- The loop can be repeated any number of times and still go through z to an accepting state.
- Hence the y part can be pumped.
- We can repeat the loop any number of times and still proceed through z to accepting state.
The Pumping Lemma

**Lemma**: If \( L \) is a regular language then **there is a** number \( n \) such that **any** string \( w \in L \), where \(|w| \geq n\), **can** be divided into three parts \( w=xyz \) such that:

1. \(|y|>0\)
2. \(|xy| \leq n\)
3. For each \( i \geq 0 \) \( xy^i z \in L \)
Informally

• For each regular language there is a constant number $n$, such that any string of length more than $n$ can be pumped.

• The pumped part $(y)$ should not be of length 0 (at least one letter).

• The parts before and after the "pumpable" part $(x,z)$ could be $\varepsilon$.

• The part we may pump is inside a prefix $(xy)$ of length at most $n$. 
Formal Proof

• Let $M$ be a DFA that recognizes $L$. Let $n$ be the number of states of $M$.
• Let $w$ be a string in $L$ of length at least $n$.
• It has a prefix of length $n$. The sequence of states in automaton $M$ that we go through while processing the prefix is $q_0$ and one state for each letter read.
• After reading this prefix we have a sequence of $n+1$ states.
Formal Proof (cont.)

- As there are only $n$ different states in automaton $M$ at least one state has to be repeated while reading the prefix.
- Denote one of them $r$.
- Let $x$ be the prefix of $w$ read before first appearance of $r$.
- Let $y$ be the substring of $y$ read between first and second appearance of $r$.
- And let $z$ be the substring of $w$ read after second appearance of $r$ till the end $w$. 
Formal Proof (cont.)

• $z$ ends with an accepting state $f$ because the string is in $L$.

• $y$ is a loop in the automaton.

• $y$ has at least length 1 because we move from state to state in the DFA only upon reading a letter.
Formal Proof (cont.)

• As there is a path from \( r \) to an accepting state \( f \) there is a direct path \( q_0 \rightarrow r \rightarrow f \) without going through the \( r \) loop even once (\( xy^0z=xz \)).

• On the other hand we can go through the loop more than once and still get out from \( r \) to \( f \) (\( xy^i z \)).

• There is no limit on a number of times we can go through the loop .

Q.E.D.
Example of an infinite languages automaton

$L = ab^*c$

Each word in $L$ consists of a path $q_0 \to q_1$ followed by any number of loops on $q_1$, followed by a path $q_1 \to q_2$.

Denote the path from $q_0$ to $q_1$ by $x$

Denote the path from $q_1$ to $q_1$ by $y$

Denote the path from $q_1$ to $q_2$ by $z$

We can say that each word $w \in L$ is $xy^iz$. 
Another example

All the strings over $\Sigma^*$

In this case $x$ and $z$ can be $\varepsilon$.

For any word of length $\geq 1$, $y$ is not $\varepsilon$. 
Finite Languages

Let’s consider the following automaton:

![Automaton Diagram]

A has 4 states. \( L(A) \) is all words of length 2 or less. \( L(A) \) is regular, however, does the pumping lemma work here???

**Theorem:** For a DFA A, if \( L(A) \) is finite, and the number of states in A is \( n \), then the longest word in \( L \) has length < \( n \)

**Proof:** if \( \exists (w \in L \text{ and } |w| \geq n) \) then \( \exists (w=xyz) \) s.t. \( 1+2+3 \) exists, thus \( L \) is infinite.
Usage of the Pumping Lemma

The main usage of pumping lemma is to prove that some language is not regular.

The technique:

1. Assume that the language is regular.
2. Assume some n, and find a string w that has a length greater than n.
3. Show that for any partition of w into x,y,z such that properties 1 and 2 of pumping lemma hold, there exists an i such that xy^iz is not in the language.
Caveats

- The Lemma:

  If $L$ is regular, then there exists $n$, s.t. $\forall (w \in L \text{ and } |w| \geq n)$, there exists $(w=xyz)$ s.t. $1+2+3$ exists.

- $\text{regular}(L) \implies \text{lemma}(L)$.

- $\text{lemma}(L) \nRightarrow \text{regular}(L)$.

- $\sim\text{lemma}(L) \implies \sim\text{regular}(L)$.

- In order to show that $L$ is non-regular, one has to show that $\forall n$, there exists $(w \in L \text{ and } |w| \geq n)$ s.t. $\forall (w=xyz)$, $\sim1$ or $\sim2$ or $\sim3$. 
Caveats

• The Lemma:

  If L is regular \( \exists n, \text{ s.t. } \forall (w \in L \text{ and } |w| \geq n) \exists (w=xyz) \text{ s.t. } 1+2+3 \) exists.

• regular(L) \( \Rightarrow \) lemma (L).

• lemma (L) \( \not\Rightarrow \) regular(L).

• \( \sim\)lemma(L) \( \Rightarrow \) \( \sim\)regular(L).

• In order to show that L is non-regular, one has to show that \( \forall n, \exists (w \in L \text{ and } |w| \geq n) \text{ s.t. } \forall (w=xyz), \sim 1 \text{ or } \sim 2 \text{ or } \sim 3 \)
Common Mistakes

• \( \exists w \in L \text{ s.t. } \text{lemma}(w) \Rightarrow \text{regular}(L) \)

• \( \forall w \in L, \text{lemma}(w) \Rightarrow \text{regular}(L) \)

• \( \exists w \in L \text{ and } \exists (w=xyz) \text{ s.t. } xy^iz \notin L \Rightarrow \sim \text{regular}(L) \)
  (you must show for all possible decompositions)

• \( \exists w \in L \text{ and } \forall (w=xyz) \text{ xy}^iz \notin L \Rightarrow \sim \text{regular}(L) \)
  (|w| might be smaller than n)
L = \{0^m1^m \mid m>0\}

• We will use the pumping lemma to show that \(L = \{0^m1^m \mid m>0\}\) is not regular.

• Assume to the contrary that \(L\) is regular.

• Let \(n\) be the number promised by the pumping lemma.

• Consider \(w = 0^n1^n\).

• \(w\) is in \(L\), \(|w|>n\) so by the pumping lemma \(w\) can be divided into three parts \(w=xyz\).
L = \{0^m1^m \mid m > 0\} \text{ (cont.)}

- x and y are both in prefix of length n and so are consist of 0's.
- Denote the length of x by s and the length of y by t.
- Then:
  - \(x = 0^s\), \(s \geq 0\)
  - \(y = 0^t\), \(t \geq 1, s + t \leq n\)
  - \(z = 0^{n-s-t}1^n\)
- By the pumping lemma if the language is regular then \(x y^i z\) is in L for any i.
\( L = \{0^m1^m \mid m > 0\} \) (cont.)

• Take \( i = 2 \).
• The resulting word is \( xy^2z = 0^s0^{2t}0^{n-s-t}1^n = 0^{n+t}1^n \).
• It is not in \( L \) as \( t \geq 1 \) and the number of 0 is greater than the number of 1.
• Thus we have a contradiction to the pumping lemma.
• That means: our assumption that \( L \) is regular is wrong.
• \( L \) is not a regular language. Q.E.D.
L=$\{a^k! \mid k \geq 0\}$

- Is $L=$\{a$^k$! $\mid k \geq 0\}$ regular?
- We will use the pumping lemma to show that $L$ is not regular.
- Assume to the contrary that $L$ is regular.
- Let $n$ be the number promised by the pumping lemma.
- We choose $w= a^{n!}$ where $n=\max(n,2)$ .
- $w$ is in $L$, $|w| \geq n$ so by the pumping lemma $w$ can be divided into three parts $w=xyz$. 
\[ L = \{ a^k \mid k \geq 0 \} \]

- x and y consist of a's.
- Denote the length of x by s and the length of y by t.
- Then:
  - \( x = a^s, \quad s \geq 0 \)
  - \( y = a^t, \quad t > 0, \quad s + t \leq n \)
  - \( z = a^{n!-t-s} \)
- By the pumping lemma if the language is regular then \( xy^iz \) is in L for any i.
\[ L=\{a^k \mid k \geq 0\} \]

• Take \( i=2 \).
• The resulting word is \( xy^2z = a^{n!+t} \).
• We know that \( t>0 \) and \( t \leq n \), which means
  \[
n!+t \leq n!+n < n!+(n+1) < n!^*(n+1) = (n+1)!\]
• Thus we have a contradiction to the pumping lemma, contradicting our assumption that \( L \) is regular.
\[ L = \{0^m1^n \mid m > n\} \]

- Assume \( L \) is regular.
- Let \( p \) be the number promised by the Lemma.
- Let \( w = 0^{p+1}1^p \)
- \( w \) can be split \( w \) into \( xyz \).
- Since \( |xy| \leq p \), \( y \) consists of only 0s, \( |y| > 0 \).
- \( \implies \#_0(xy^0z) \leq \#_1(xy^0z) \)
- \( \implies xy^0z \notin L \) contradicting the pumping lemma.
- \( \implies L \) is non-regular.
\[ L = \{ w \mid \#_1(w) = \#_0(w) \} \]

• Is L regular? (no!)

• **Proof idea:**
  – Assume to the contrary that L is regular.
  – By the pumping lemma we have \( n \) s.t. …
  – Choose \( w=0^n1^n \in L \)
  – Pump \( w \) towards \( w' \notin L \)

• **Problem:** if we choose \( y=0^n1^n \) then \( w' \in L \) ??
\[ L = \{w \mid \#_1(w) = \#_0(w)\} \]

- Is \( L \) regular? (no!)

- **Proof**:
  - Regular languages are close under intersection.
  - Assume to the contrary that \( L \) is regular.
  - It gives \( L' = (0^*1^* \cap L) \) is regular.
  - But \( L' = 0^n1^n \) is not regular!
  - \( L \) is non-regular
L = \{ww \mid w \in \{0,1\}^* \}

• Is L regular? (no.)

• **Proof idea:**
  – Assume to the contrary that L is regular.
  – By the pumping lemma we have $n$ s.t. …
  – Choose $w=0^n10^n1 \in L$
  – Pump $w$ towards $w' \notin L$
A language is NOT regular if it does not satisfy the Pumping Lemma

A language is regular if it has a DFA/NFA/RE recognizing it
Regular Languages: Recap

- Finite Automata \((Q, \Sigma, \delta, q_0, F)\)
- Regular languages
- Closures (union, intersection, complements, minus)
- Non deterministic automaton: \(\delta: Q \times \Sigma \times \epsilon \rightarrow 2^Q\)
- Equivalence \(DFA \leftrightarrow NFA\)
- Closures (concatenation, Kleene star).
- Regular Expressions
- Equivalence \(DFA \leftrightarrow RE\)
- Minimal DFA
- Pumping Lemma.