Regular Languages (again!)
Ways of Proving a Language is Regular

- We have learnt many different ways to prove a language is regular
- Can you name some?
Finite Languages

- Are all finite languages regular?
What about Infinite Languages?

- So far, we have seen many regular languages that are infinite
- Examples?
- What do they all have in common?
Exercise

Consider \( L = \{ 0^n 1^n : n \geq 0 \} \).

- Try to design a DFA for it
- What is the difficulty?
Nonregular Languages
Infinite Regular Languages

A regular language is infinite if its corresponding regular expression contains a Kleene star. Kleene stars correspond to loops in finite automata.

\[(a \cup b)^*abba(a \cup b)^*\]

Both Kleene stars and loops give rise to simple repetitive patterns in the language.
The Pigeonhole Principle

- Consider a DFA $M$ with $p$ states for language $L$
- Any string $w$ accepted by $M$ of length at least $p$ must visit some state more than once
  - Number of states visited is the length of input string + 1
- The pigeonhole principle: *if $n$ pigeons are placed into fewer than $n$ holes, then some hole must have more than one pigeon in it*

Let $w = w_1w_2....w_p \in L$

Only $p$ states: not all of the $p+1$ $q_i$'s can be distinct. Some state must be visited more than once!
Machine Loops

Let $q_j$ be the first repeated state, that is, $q_j = q_{j+k}$ for some $k$, $1 \leq j < j+k \leq p$. 
Pumping Strings

- Let $L$ be a regular language. For any string $w \in L$ that is “sufficiently long” we can split the string into three pieces and pump the middle.
- Let $w = xyz$. Then, for all $i \geq 0$, $xy^iz \in L$, where $y^i$ denotes $i$ copies of $y$ (that is, $xz$, $xyz$, $xyyz$, $xxyyz$, $xyyyyz$, etc. must all be in $L$).
PUMP ALL THE STRINGS!
Pumping Lemma

**Theorem.** If $A$ is a regular language, then there exists a natural number $p$ (the pumping length) such that all strings $w \in L$ of length at least $p$ can be written as $w=xyz$ such that

1. $|y| > 0$
2. $|xy| \leq p$
3. for each $i \geq 0$, $xy^iz \in L$

Constraint that $y$ must appear among the first $p$ symbols

![Diagram](image)
Pumping Lemma in Action

- Used to prove that a language $L$ is not regular by contradiction
- First assume $L$ is regular
- Pumping lemma guarantees existence of a pumping length $p$ such that all strings of length at least $p$ can be pumped
- Find a string $w$ of length $p$ or more that “cannot be pumped”
  - Show that no matter how you divide $w$ into $xyz$ such that $|y|>0$ and $|xy| \leq p$, there always exists an $i \geq 0$, such that $xy^iz$ is not in $L$
  - In other words, check all possible ways to split $w$ and reach a contradiction for each
Deciding Regularity: Pumping Lemma

**Theorem.** The language \( L = \{ 0^n 1^n : n \geq 0 \} \) is not regular.

**Proof.** Assume \( L \) is regular.

Let \( p \) be the pumping length guaranteed by the pumping lemma.

Let \( w = 0^p 1^p \). Then \( w \in L \) and \( |w| = 2p \geq p \). Consider all possible ways to split \( w \) into \( xyz \) such that \( y \) is non-empty and \( |xy| \) is at most \( p \).

- String \( y \) must only consists of 0s!
Picking the Right Strings

Theorem. The language $C = \{ uu \mid u \in \{0,1\}^* \}$ is not regular.

Proof.
Final Questions

- Do all regular languages satisfy the pumping lemma?
- If a language satisfies the pumping lemma does that mean it is regular?