Non-regular languages

$$A = \{ a^n b^n \mid n \geq 0 \}$$

$$B = \{ ww \mid w \in \Sigma^* \}$$

$$C = \{ 1^{n^2} \mid n \geq 0 \}$$

Pumping Lemma

All regular languages have some special property. If a language doesn't have this property then it isn't regular.

All strings in a language can be "pumped" if they are at least as long as a certain value, called the <u>pumping length</u>.

some substring in the language can be repeated any number of times with the resulting string remaining in the language.

Pumping Lemma

If A is a regular language, then there exists a number p (the pumping length).

$\forall s \in A$ such that $|s| \geq p$

s may be divided into 3 pieces

$s = xyz$ that satisfies

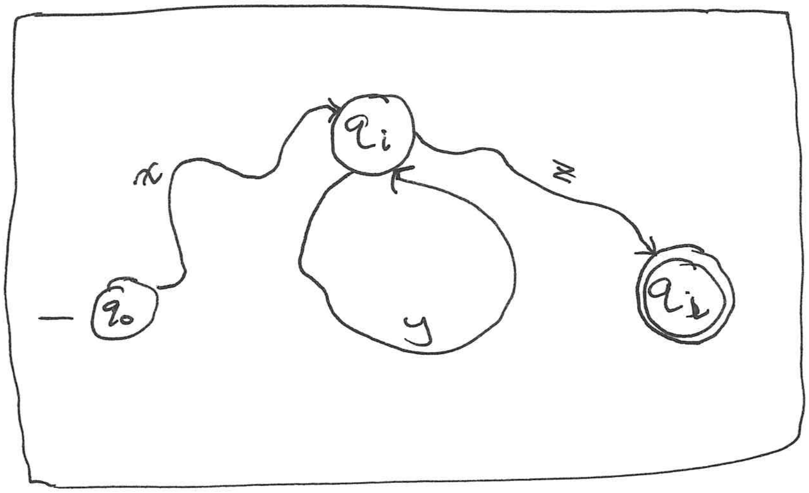
1. $\forall i \geq 0$, $xy^iz \in A$
2. $|y| > 0$
3. $|xy| \leq p$

What is p?

p is the number of states.

If the length of s is greater than the number of states then some state or sequence of states must be visited ~~twice~~ more than once.

Pumping Lemma Intuition



if a loop exists, then I can take the loop again by just adding another repetition of the sequence of characters consumed by the loop. I can also choose not to take the loop at all.

Theorem

The language $A = \{a^n b^n \mid n \geq 0\}$ is not regular

Proof by Contradiction

- Assume that A is regular
- If A is regular then the pumping lemma hold for some pumping length p
- Choose $s = a^p b^p$
- Because $s \in A$ and $|s| > p$, s can be split into three pieces, $s = xyz$, where
  $\forall i \geq 0, \; xy^iz \in A$
- Because $|xy| \leq p$ and $|y| > 0$, y must be some number of ~~as~~ a's and can't contain any bs
- let $r = |y|$
- The string $xyyz = a^{p+r} b^p$ has more as than bs
- Therefore the pumping lemma does not hold and we have reached a contradiction.

The language $A = \{a^n b^n \mid n \geq 0\}$ is not regular

Theorem

The language $A = \{ 1^{n^2} \mid n \geq 0 \}$ is not regular.

Proof by Contradiction

Assume A is regular. If A is regular then the pumping lemma holds for some pumping length p.

Choose $s = 1^{p^2}$. s can be divided into three pieces

$s = xyz$ such that $|y| > 0$ $|xy| \leq p$

let $r = |y|$ $\quad 0 < r \leq p$

$xyyz = 1^{p^2+r}$

$$1^{p^2} < 1^{p^2+r} < 1^{(p+1)^2}$$

$(p+1)^2 = p^2 + 2p + 1$

since $r \leq p$

$r < 2p$

Since $xyyz \notin A$ the pumping lemma does not hold

Therefore

The language $A = \{ 1^{n^2} \mid n \geq 0 \}$ is not regular

$A = \{ w \mid w$ contains a 1 in the second to last position $\}$ $(0 \cup 1)^* \cdot 1 \cdot (0 \cup 1)$

$B = \{ 0^n 1^n \mid n \geq 1 \}$

<u>Def</u> Distinguishable Strings

Let L be a language over an alphabet Σ. We say that two strings x and y are <u>distinguishable</u> with respect to L if there is a string z such that $xz \in L$ and $yz \notin L$ or vice versa.

Example of distinguishable strings

0 and 00 with respect to B

take $z = 1$ $\quad 01 \in B$ and $001 \notin B$

00 and 01 with respect to A

take $z = 0$ $\quad 000 \notin A$ and $010 \in A$

Example of non-distinguishable strings

$x = 0110$ and $y = 10$

Lemma

Let L be a language, M be a DFA that decides L, and x and y be distinguishable strings with respect to L. Then the state reached by M on input x is different from the state reached by M on input y.

Proof by Contradiction

- Assume that M reaches the same state q on input x and input y.
- Let z be the string such that

  $xz \in L$ and $yz \notin L$ (or vice versa)

- Lets call q' the state reached by M on input xz.

  Note that q' is the state reached by M starting from q and given the string z.

- On input yz M must reach the same state q' because M reaches state q given y and then goes from q to q' given z
- Must either accept both xz and yz or reject both.

Def Distinguishable Set of Strings

Let L be a language. A set of strings $\{x_1, x_2, \ldots, x_k\}$ is distinguishable if for every two distinct strings x_i and x_j we have x_i is distinguishable from x_j.

Examples

A

$\{00, 01, 10, 11\}$

B

$\{0, 00, 000\}$

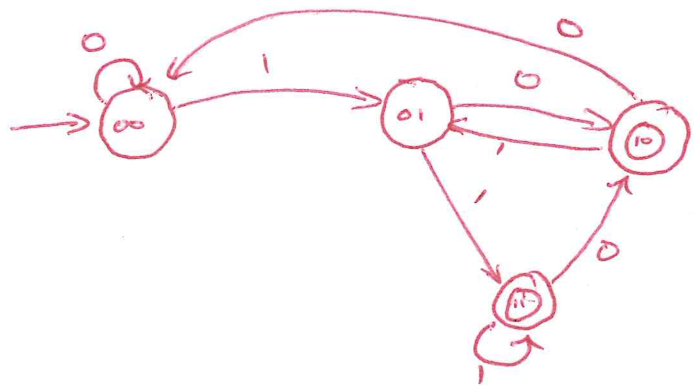
Let L be a language, and suppose there is a set of k distinguishable strings with respect to L. Then every DFA for L has at least k states.

Proof

- If L is not regular then there is no DFA for L
- If L is regular
  - Let M be ~~the~~ a DFA for L
  - Let $\{x_1, \ldots, x_k\}$ be the distinguishable strings
  - Let q_i be the state reached by M after reading x_i.
  - For every $i \neq j$ x_i and x_j are distinguishable so q_i and q_j must be different (previous lemma)
  - So we have k different states $q_1 \ldots q_k$ in M, so M has at least k states.

For language A

$\{00, 01, 10, 11\}$ are distinguishable with respect to A, so every DFA for A has at least 4 states.



Language B

For every $k \geq 1$ consider the set $\{0, 00, \ldots 0^k\}$ of strings made of k or fewer 0s. This set is distinguishable with respect to B.

This means a DFA for B would require an infinite number of states, and B is not regular.

If two strings x and y are indistinguishable over a language L we write $x \approx_L y$.

$x \approx_L y$ means that for every string z

$xz \in L$ iff $yz \in L$

$x \approx_L y$ iff $y \approx_L x$

if $x \approx_L y$ and $y \approx_L w$ then $x \approx_L w$

- The relation $\approx_L$ is an equivalence relation over Σ^*

- An equivalence ~~eal~~ class in Σ^* with respect to $\approx_L$ is a set of strings that are all indistinguishable from one another, and that are all distinguishable from all others not in the set

written as $[x]$ (the equivalence class containing the string x)

The previous lemma can be restated as

Every DFA for L must have at least as many states as the number of equivalence classes in Σ^* with respect to $\approx_L$

In fact there is always a DFA with exactly this number of states

Myhill-Nerode Theorem

Let L be a language over Σ. If Σ^* has infinitely many equivalence classes with respect to $\approx_L$, then L is not regular. Otherwise, L can be decided by a DFA whose number of states is equal to the number of equivalence classes in Σ^* with respect to $\approx_L$.

Proof

- Suppose there are a finite number of equivalence classes
- Define a state for each equivalence class. The start state is the class $[\varepsilon]$ and every state of the form $[x]$ for $x \in L$ is an accepting state.
- From state $[x]$, reading the character a, the DFA moves to state $[xa]$