Lecture 3

Proof by Induction

Example: Show that for every palindrome w, we have $w = w^R$. ($\Sigma = \{a, b\}$)

Proof: by induction on the form of w.

Basis: $w = \lambda \Rightarrow w^R = \lambda$ ✓

$w = a \Rightarrow w^R = a$ ✓

$w = b \Rightarrow w^R = b$ ✓

Ind. Hyp. Assume a palindrome w satisfies $w = w^R$.

Show: For $m = awa$, we have $m = m^R$

$$m^R = (awa)^R$$
$$= aw^Ra$$
$$= awa \quad \text{by ind. hyp we have } w = w^R$$
$$= m.$$

(Similarly, for $m = bwb$ we have $m = m^R$)

Example: Prove that $3^n > 3n$ for $n \geq 2$.

Proof: by induction

Basis: $n = 2 \Rightarrow 3^2 > 3 \cdot 2$ ✓

Ind. Hyp: Assume $3^k > 3k$ for $k < n$.

Show: $3^n > 3n$ is true.

$$3^n = 3 \cdot 3^{n-1}$$

$$> 3 \cdot 3(n-1) \quad \text{by ind. hyp.}$$

$$= 9n - 9$$

$$= 3n + 6n - 9$$

$$> 3n \quad \text{if } n \geq 2.$$

$\Rightarrow 3^n > 3n$ //

Proof by Contradiction

Example: Let $a, b, c > 0$. Prove: if $\underbrace{ab = c}_{p}$, then $\underbrace{a \leq \sqrt{c} \text{ or } b \leq \sqrt{c}}_{q}$. $(p \Rightarrow q)$

Proof: by contradiction

Assume the opposite.

- Assume p is true
- Assume q is false

$\Rightarrow a > \sqrt{c}$ and $b > \sqrt{c}$

$\Rightarrow a \cdot b > \sqrt{c} \cdot \sqrt{c} = c$

Contradiction.

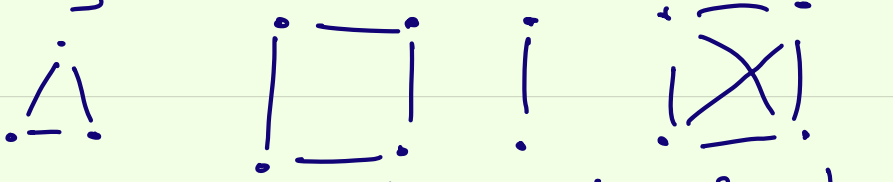
Exercise: - Prove that $\sqrt{2}$ is irrational.

- Prove that there are infinitely many primes.
- Prove that there exists no set of all sets. (Bertrand Russell)

Proof by Construction

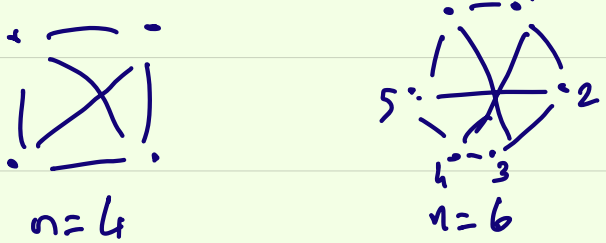
Example: A k-regular graph is a graph in which every node has degree k.

e.g.



2-regular 2-regular 1-regular 3-regular

Claim: For every even number $n \geq 4$, there exist 3-regular graphs with n nodes.



$n=4$ $n=6$ $n=8$

In general we place every node into a cycle and connect the opposite nodes.

nodes .... $0, 1, \dots, n-1$

$G = (V, E)$ where

$V = \{0, 1, \dots, n-1\}$

$E = \{(0,1), (1,2), (2,3), \dots, (n-1, 0)\}$

$\cup \{(0, \frac{n}{2}), (1, \frac{n}{2}+1), \dots, (\frac{n}{2}-1, n-1)\}$

Some more defs. on languages.

- Let Σ be an alphabet
- Let $L_1, L_2 \subseteq \Sigma^*$ be two languages.

Concatenation of L_1 and L_2:

$L_1 \cdot L_2 = \{u \cdot v \mid u \in L_1 \text{ and } v \in L_2\}$

e.g. $\Sigma = \{a, b\}$ $L_1 = \{a, b\}$, $L_2 = \{aa, bb\}$

$L_1 \cdot L_2 = \{aaa, abb, baa, bbb\}$

$L_2 \cdot L_1 = ?$

- For a string w and $n \geq 0$

$$w^n = \begin{cases} \lambda & \text{if } n = 0 \\ w^{n-1} \cdot w & \text{if } n \geq 1 \end{cases}$$

e.g. $w = 01 \Rightarrow w^3 = w^2 \cdot w = w^2 \cdot 01$
$= w \cdot w \cdot 01$
$= 01 \cdot 01 \cdot 01$

- For a language L and $n \geq 0$

$$L^n = \begin{cases} \{\lambda\} & \text{if } n = 0 \\ L^{n-1} \cdot L & \text{if } n \geq 1. \end{cases}$$

- Kleene closure of a language L:

$$L^* = L^0 \cup L \cup L^2 \cup \ldots = \bigcup_{i \geq 0} L^i$$

- Positive closure of a language L:

$$L^+ = L \cup L^2 \cup L^3 \cup \ldots = \bigcup_{i > 0} L^i$$

Note: $L^* = L^+ \cup \{\lambda\}$