Lecture 3 MIT 25/09/2025

# Casework and Strong induction

## Proof Techniques (so far)

- existential by example/Construction
  $\exists x \cdot P(x) \rightarrow$ Construct $x^*$, prove $(x^*)$
- universal by Instantiation
  $\forall x : P(x)$ take arbitrary $x$, prove $P(x)$
- Implication by direct argument
  $\forall$ P Impies Q $\rightarrow$ Assume P, prove Q.
- Implication by contrapositive
  P impries Q, aka $\bar{Q}$ impires $\bar{P}$ $\rightarrow$ Assume $\bar{Q}$ prove $\bar{P}$. $\neg Q$ = not Q
  $\neg\neg Q \vee \neg P$
  $Q \vee \neg P$
- Contradiction
  P, aka $\bar{P}$ impries false $\rightarrow$ Assume $\bar{P}$ and prove false, i.e prove Q and $\bar{Q}$ for some preposition Q
- Universal over $\mathbb{N}$ by Induction.
  $\forall n \in \mathbb{N} . P(n) \rightarrow$ Prove $P(0)$ and $P(n)$ Implies $P(n+1)$ for every $n \geq 0$.

Today Proof by cases & Strong Induction.

## Proof by 2 cases.

Take any proposition C
- $C \vee \neg C$ is a tautology ← means always true!
- so P is equivalent to $\underbrace{(C \vee \neg C)}_{\text{true}} \rightarrow P$. $\in (C \rightarrow P) \wedge (\neg C \rightarrow P)$
  ~~true~~ $\vee P$

prove P is true assuming C P. & prove P is true assumit $\neg C$

Example 2.

Mutual friends/strangers.

Template (for any ~~sta~~ proposition C)

Thm: P is true.

Proof: by cases on truth ~~table~~ value of C

Case 1: C is true (i.e, assume C true)
then P is true b/c $\boxed{?}$ ← use C

Case 2: C is false

then P is true b/c $\boxed{?}$ ← use $\neg C$.

C is either true or false

⇒ Cases are exhaustive. □ ← QED.

Example 1.

Thm. $(A \to B) \vee (B \to C)$
is a tautology.

$\forall A, B, C \in \{true, false\}: (A \to B) \vee (B \to C)$

Proof by cases: B is either true or false

Case 1: B is true

$A \to B$ is true   $A \to true$.

$\neg A \vee B$   $\neg A \vee true$.

$\Rightarrow (A \to B) \vee \ldots$ is true

Case 2: B is false

$B \to C$ is true   $false \to C$.

$\underbrace{\neg B}_{true} \vee C$

$\vee (B \to C)$ is true.

in either case, P is true □ ← QED

Example 2.

Mutual friends/strangers.

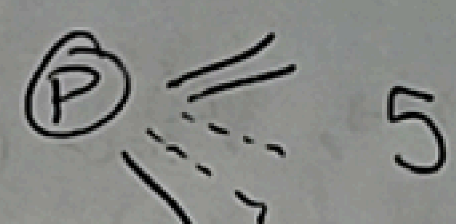
- 6 people
- every 2 people either friends XOR not friends

Strangers

Thm. Theres always either ① 3 mutual friends or ② 3 mutual strangers.

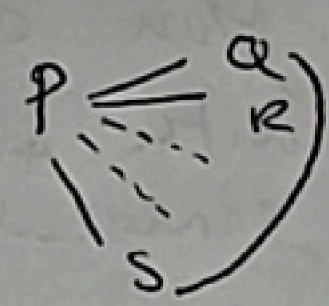
Proof: by cases

- Pick any person P.



Case 1. either P has $\geq 3$ friends or q, r, s.

Case 1a Some 2 of q, r, s are friends.



Then P + those 2 ppl are 3 mutual friends. ①

~~Case~~ 2.

Case 1b. ~~Q, R, S~~. no 2 of q, r, s are friends.

ie. mutual strangers ②

Case 2. P ~~has $\geq$ strangers~~.

P has $\leq 2$ friends.

i.e. P has $\geq 3$ Strangers.

Symmetric to case 1, swapping friends vs strangers.

① $\leftrightarrow$ ②.

This is a case of Ramsey Theory.

$R(3,3) \leq 6$

$R(t,s)$

$R(4,4) = 18$

$R(5,5) = \in \{43, 48\}$

What if we want more than 1/2 cases

Proof by $k$ cases: $C_1 \vee C_2 \vee \dots \vee C_k$ is a tautology.

Template theorem: $P$ is true.

Proof by cases

Case $i$: $C_i$ is true (assume $C_i$)

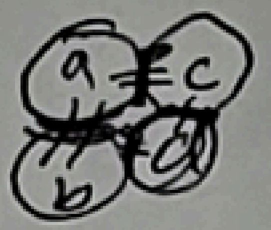
then $P$ is true b/c $\boxed{\dots}$.

Cases are exhaustive because $\boxed{\dots}$

$C_1 \vee C_2 \vee \dots \vee C_k$ is a tautology.

Famous Proof By Cases, Four Colour Theorem.

De Morgan
- Inverted term Mathematical Induction.




. you can colour regions with 4 colours to get no consecutive/touching colours.

Kempe*
- proved the theorem.
  but, it got disproved
  ~~Too many bugs~~ -> Continuously not exhaustive by other theorists.

- Appel & Haken ← one of the first mathematical proof using a computer.

2005 -> COq -> entire proof in this language. = trusted scrutiny.
still 100s cases.

## Induction axiom.

for a predictate $P(n)$ over $n \in \mathbb{N}$

if ① $P(0)$ is true

and if ② $P(n) \Rightarrow P(n+1)\ \forall n \in \mathbb{N}$

then $P(n) \cdot \forall n \in \mathbb{N}$

## Block diagrams:

- true props $\rightarrow \boxed{\text{block}} \rightarrow$ true prop.
- given ① $\boxed{\text{base case}} \rightarrow P(0)$

  ② $P(n) \rightarrow \boxed{\text{Step}} \rightarrow P(n+1)\ \forall n \in \mathbb{N}$
- Chain base case argument

$$\boxed{\text{basec}} \rightarrow P(0) \rightarrow \boxed{\text{Step}} \rightarrow P(1) \rightarrow \boxed{\text{step}} \rightarrow P(2) \rightarrow \boxed{\text{Step}} \cdots$$

base $\rightarrow P(0)$ — true

$P(n) \rightarrow \boxed{\text{Step}} \rightarrow P(n+1)$ — true

$\Rightarrow$ Induction $\Rightarrow \forall n \in \mathbb{N}$ $P(n)$ is true.

---

Equivalently: ~~Ind~~ induction

if ① $P(0)$ is true

and ② $\forall n \geq 1 : P(n-1) \rightarrow P(n)$

then $\forall n \in \mathbb{N}: P(n)$

# Strong Induction

- Just gives you more to work with.
- If $\forall n \in \mathbb{N}: [\underbrace{P(0) \wedge \cdots \wedge P(n-1)}] \to P(n)$ then $\forall n \in \mathbb{N}: P(n)$

true for $n=0$.

$\boxed{P(i)\ \forall i < n.}$

* Proof. by induction (that strong induction is true).

Induction hypothesis $IH(n) = P(0) \wedge \cdots \wedge P(n-1) = \forall i < n: P(i)$

Base case: $IH(0) =$ true by definition

Step: Assume $IH(n)$ WTS $IH(n+1)$

$$IH(n+1) = P(0) \wedge \cdots \wedge P(n-1) \wedge P(n) = IH(n) \vee P(n)$$

true.

By induction axiom

$\forall n \in \mathbb{N}: IH(n)$

$\forall n \in \mathbb{N}: IH(n+1)$

$\forall n \in \mathbb{N}: P(n) \wedge P(n)$

<u>Template</u> strong induction

Thm $\forall n \in \mathbb{N}: P(n)$

Proof: by strong induction - assume $P(k)$. $\forall k < n$

WTS $P(n)$,

base case $P(0), (P(1)), \ldots P(b)$ b/c $\boxed{?}$

Ind step assume $n > b$

Then $P(n)$ b/c $\boxed{?}$

Unstacking game:

- Start with 1 stack of $n$.
- **move** splits stack of $S>1$ items into a stack with $a, b$ items.

Where $a+b=S$, $a \geq 1$, $b \geq 1$

choose as you want.

$S \to a, b$ $\boxed{\text{get } a \cdot b \text{ points}}$

Thm: $\forall n \in \mathbb{N}$: Every ~~strategy~~ Strategy for this game with 1 stack of items scores $\frac{n(n-1)}{2}$.

Proof: Strong induction. $P(n)$

- assume $P(k)$ is $\forall k<n$ "is true for all smaller $n$"
- Consider first move $\underset{a \cdot b \text{ points.}}{n \to a, b}$ $a+b=n$ $a, b \geq 1$
- left with 2 piles (games $a, b$)
- $a, b < n$ because $a+b=n$, $a, b \geq 1$.
- by strong induction we know, $P(a)$ & $P(b)$ are true.

$\Rightarrow$ get $\frac{a(a-1)}{2}$ points from $a$

$\frac{b(b-1)}{2}$ points from $b$

$a \cdot b$ points from first move

$$\text{Total} = a \cdot b + \frac{a(a-1)}{2} + \frac{b(b-1)}{2}$$

$$= \frac{2a \cdot b + a^2 - a + b^2 - b}{2} = \frac{\overset{a^2+2ab+b^2-a-b}{(a+b)^2 - (a+b)}}{2}$$

$$= \frac{(a+b)(a+b-1)}{2}$$

$$= \frac{n(n-1)^2}{2}$$

Video FPS → biary, digits 25/09/2025

Sound, how to represent, analouge as digital, each number corresponds to a frequency, volume, length,
pitch, volume, duration like the RGB of colour, but for sound.

Input → ☐ → output.

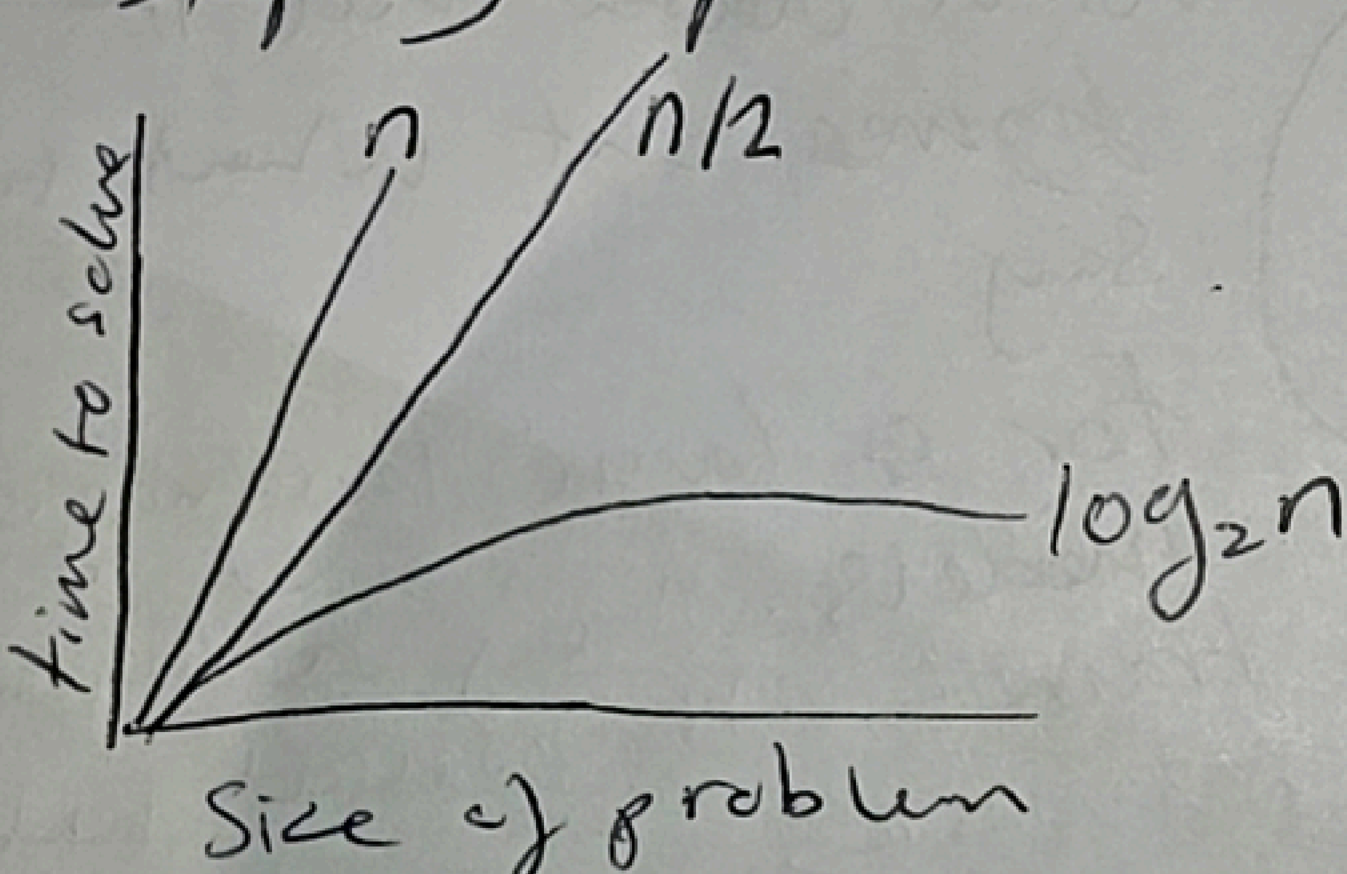
How would a computer know 74 73 33 is a colour message or sound?

— Depends on the context.

A better programmer can tell the code to read a certain way.

Input → $\boxed{\text{Algorithm}}$ → output.

step by step instructions.




like: in a phonebook - flip page by page
n →

n/2 → - flip every other page.

$\log_2 n$ - go to middle and half, half, half.

Implementing algorithms. → Code.

pseudocode - use english succinctly.

e.g
1. Pick up phone book
2. Open to a middle of phone book
3. look at page
4. If Person is on page
   call person
5. Else if person . . .
   . . . - . .
   Else . . .
   Quit . . .

If theres a bug not covering a certain scenario computer could just quit.

Functions. Action/Verb.

If, Else If - conditionals.

decide where to go off of questions, these questions are boolean expressions

Booleans have yes or no, black or white, 1 or 0 questions. The answers are simple on.off.

Going back to, looping.

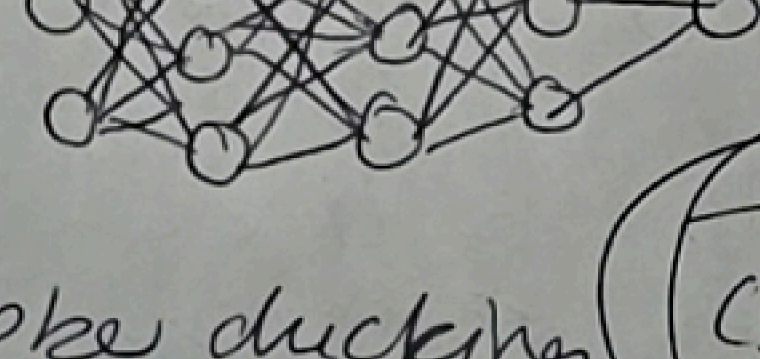
Artificial Intelligence.

If student says hello
say hello.
Else if student says goodbye
say goodbye.
etc..

Infinite no. of questions. Chatbot you train with data and it figures out what to say. For ~~&~~ large language models. They are typically trained with neural ~~network~~ networks.

Go Implementing (from biology) like a circle represents a neuron, a line represents a pathway, providing all internet books etc.

The CS50 duck is an ai representation. To talk through the problem. Rubber ducking.



I can USE THIS.

CS50.ai was their ai.