10/18/2024

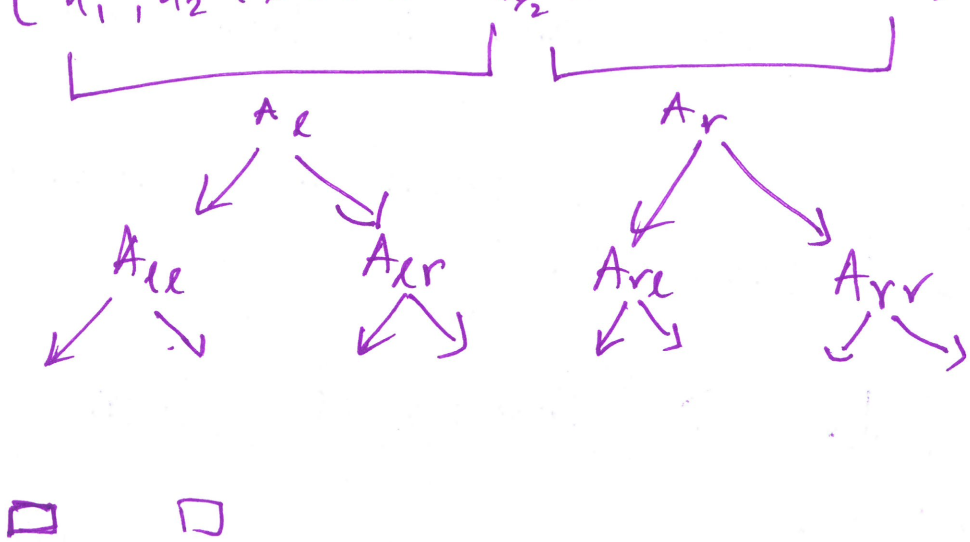
Induction

- Recursion is a common strategy for problems in CS
  - Take a problem instance
  - split it into smaller subproblems
  - Once the subproblems are small enough that they tractable, you solve them.

Ex: Binary Search

- Given a sorted array, we want to find an element.

$$A = [a_1, a_2 \ldots\ldots a_{n/2} \ldots\ldots a_n]$$



Mathematical induction is a proof technique that is analogous to recursion.

(This is only used to prove claims dealing with natural numbers)

$$\forall n \in \mathbb{N} : \underbrace{0 + 1 + 2 + 3 + \cdots + n}_{\sum_{i=0}^{n} i} = \frac{n \cdot (n+1)}{2}$$

In induction, we do something like this

step 1: We prove this claim for the smallest possible case.

(we prove for $n=0$)

Step 2: We assume that this claim holds for $n (\geq 1)$ and we show that the claim is true for $n+1$

Let p be a predicate concerning integers greater than 0.

$P : \mathbb{N} \longrightarrow \{\text{True}, \text{False}\}$, define $P(n)$ as follows.

$$P(n) \text{ says } \sum_{i=0}^{n} i = \frac{n \cdot (n+1)}{2}$$

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

I assume that I have already prove 2 things.

1. The base case (smallest case possible)
2. Inductive case: $\forall n \geq 1 : P(n-1) \Rightarrow P(n)$.

Let's see why these two steps helps us to prove $[\forall n \in \mathbb{N} : P(n)]$

Why?

Statement	reason
$\boxed{P(0)}$	we showed that the base case is true.
$\boxed{P(0) \Rightarrow P(1)}$	using inductive case and plugging ~~n=1~~ $n=1$.
$P(1)$ is true	"Modus ponens" Since we already know $P(0) \Rightarrow P(1)$, therefore if $P(0)$ is true, $P(1)$ has to be true. "the way that affirms by affirming"

p	q	$p \Rightarrow q$
0	0	1
0	1	1
1	0	0
1	1	[illegible]

if $P(0) \Rightarrow P(1) \land P(0)$, then $P(1)$ has to be true

(this is the only case to consider if $[P(0) \Rightarrow P(1)$ is true$]$)

→ "continued on Monday".

$P(1) \Rightarrow P(2)$ using the inductive case & plugging $n=2$

$P(2)$ using modus ponens.

$P(2) \Rightarrow P(3)$ using inductive case & plugging $n=3$

$P(3)$ modus ponens.

⋮

$P(n)$ modus ponens.

$\underbrace{P \Rightarrow q}_{\text{True}}$ P is false what can you say about q?

$$\forall n \geq 0: \sum_{i=0}^{n} i = \frac{n \cdot (n+1)}{2}$$

step 01: Define the predicate $p(n)$

$P: \mathbb{N} \longrightarrow \{T, F\}$, $P(n)$ is defined as follows:

$$P(n) = \begin{cases} T, & \text{if } \sum_{i=0}^{n} i = \frac{n \cdot (n+1)}{2} \\ F, & \text{otherwise} \end{cases}$$

step 02: state the variable that you are performing induction

- we perform induction over $n \in \mathbb{N}$

we want to show: $[\forall n \in \mathbb{N} : P(n)]$

step 03: State the base case

$n=0$ is the base case.

Step 04: Prove the base case

$$\text{WTS}: \underbrace{\sum_{i=0}^{0} i}_{LHS} = \underbrace{\frac{0 \times (0+1)}{2}}_{RHS}$$

$$LHS = \sum_{i=0}^{0} i = 0$$

$$RHS = \frac{0 \times 1}{2} = 0$$

$$LHS = RHS$$

$\therefore$ base case is proven.

Step 05: State the inductive case

$\forall n \geq 1 : P(n-1) \Rightarrow P(n)$

This is okay → $\boxed{\forall n \geq 0 : P(n) \Rightarrow P(n+1)}$

$\forall n \geq 1 : \left(\sum_{i=0}^{n-1} i = \frac{(n-1)n}{2} \right) \Rightarrow \underbrace{\sum_{i=0}^{n} i}_{LHS} = \underbrace{\frac{n(n+1)}{2}}_{RHS}$

Step 06: Prove the inductive case

Let $n \geq 1$

Assume $P(n-1)$ is true ← Inductive assumption.

$LHS = \sum_{i=0}^{n} i$ (def of summation)

$= \sum_{i=0}^{n-1} i + n$ (def of summation)

$= \frac{(n-1) \cdot n}{2} + n$ substituting inductive hypothesis.

$LHS = \frac{n(n-1) + 2n}{2} = n\frac{(n-1+2)}{2} = \underbrace{\frac{n \cdot (n+1)}{2}}_{RHS}$ by algebra

$LHS = RHS$

We have proven $P(n)$ is true, when $P(n-1)$ is true.

Step 07: We have shown that $P(0)$ (base case) is true, and $\forall n \geq 1 : P(n-1) \Rightarrow P(n)$.

$\therefore$ using the principle of mathematical inductio[n]

$[\forall n \geq 0 : P(n)]$ is true //

claim: $\forall n \geq 0, \sum_{i=0}^{n} 2^i = 2^{n+1} - 1$

Sum of first $(n+1)$ terms of powers of 2

$$\sum_{i=0}^{n} 2^i = \underbrace{2^0 + 2^1 + 2^2 + \dots + 2^{n-1} + 2^n}_{(n+1) \text{ terms}}$$

Step 01: Define the predicate $P(n)$

$P: \mathbb{N} \longrightarrow \{T, F\}$ $P(n)$ is defined as follows:

$$P(n) = \begin{cases} T, & \text{if } \sum_{i=0}^{n} 2^i = 2^{n+1} - 1 \\ F, & \text{otherwise} \end{cases}$$

Step 02: Define the variable that you perform induction.

Variable $n \in \mathbb{N}$

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

Step 03: ~~Prove~~ state the base case

$n = 0$

Step 04: prove the base case.

WTS: $\underbrace{\sum_{i=0}^{0} 2^i}_{LHS} = \underbrace{2^{0+1} - 1}_{RHS}$

$LHS = \sum_{i=0}^{0} 2^i = 2^0 = 1$ $RHS = 2^1 - 1 = 1$ $LHS = RHS$ //

Step 05: state the inductive case:

$$\forall n \geq 1 : P(n-1) \Rightarrow P(n)$$

$$\left[\forall n \geq 1 . \left(\sum_{i=0}^{n-1} 2^i = 2^n - 1\right) \Rightarrow \left(\underbrace{\sum_{i=0}^{n} 2^i}_{LHS} = \underbrace{2^{n+1}}_{RHS} - 1\right)\right]$$

Step 06: Prove the inductive case

Let $n \geq 1$

Assume $P(n-1)$ is true — Inductive hypothesis

$LHS = \sum_{i=0}^{n} 2^i$ — def of summation

$LHS = \sum_{i=0}^{n-1} 2^i + 2^n$ — def of summation

$LHS = 2^n - 1 + 2^n$ — substituting inductive hypothesis.

$LHS = 2 \cdot 2^n - 1 = \underbrace{2^{n+1}}_{RHS} - 1$ — by algebra

$LHS = RHS$

Step 07: We have proven $P(n)$ is true, when $P(n-1)$ is true.

Since we have shown that the base case $P(0)$ is true & inductive case is true, we can conclude that $[\forall n \geq 0 : P(n)]$ is true using the principle of mathematical induction