CS 1010 Discrete Structures Lecture 7: Mathematical Induction

Maria Francis

January 7,2021

Mistaken Proofs

- Mathematical Induction is a powerful technique but if not done correctly then you can end up proving ridiculous statements like $n = n + 1$.
- How? Assume $k = k + 1$, then $k + 1 = k + 1 + 1$ (by I.H.) and looks like we have proved it! (NO!!)
- Usually error is in the basis step which many times we think is easy to show and we ignore it.
- False Claim: Every set of lines in the plane, no two of which are parallel meet in a common point.
  - $P(n)$: every set of n lines in a plane, no two of which are parallel meet in a common point.
  - We will attempt to prove that $P(n)$ is true for all $n \in \mathbb{N}$, $n \geq 2$.

Mistaken Proofs

- Basis Step: $P(2)$ is true since any two lines in the plane that are not parallel meet in a common point by defintion.
- Inductive Step: Assume $P(k)$: every set of k lines in the plane, no two of which are parallel, meet in a common point, is true.
- T.S.T. every set of $k+1$ lines in the plane, no two of which are parallel, meet in a common point.
- Consider a set of$k+1$ distinct lines in the plane. The first k of these lines meet in a common point p_1 (I.H.).
- Also, by I.H. the last k of these lines meet in a common point p_2.

Mistaken Proofs

- Both the points have to be the same, else *the lines that contain* p_1 *and* p_2 *will have to be on the same line since* p_1 *and* p_2 *determine a line.* But our assumption says all the lines are distinct.
- Thus $p_1 = p_2$ lies on all $k + 1$ lines.
- Done with basis step and inductive step. So we can conclude that $P(n)$ is true for all n.
- No! Where is the error? In the inductive step we needed $k \geq 3$ because $P(2)$ does not imply $P(3)$.
- When $k = 3$ we need to show three distinct lines meet in a common point. We use the same argument: First two lines meet in a common point p_1 and last two lines in p_2.
- But here the lines that contain p_1 and p_2 is just one line, the second line, so p_1 and p_2 can be different.

Guidelines for Mathematical Induction Proofs

- Write down what is $P(n)$ clearly as well as from which value onwards it is true.
- Write down basis step and inductive step clearly.
- State the inductive hypothesis $P(k)$ clearly.
- State $P(k+1)$ clearly.
- Prove the statement $P(k+1)$ making use the assumption $P(k)$. Be sure that your proof is valid for all integers k with $k \geq b$, taking care that the proof works for small values of k, including the base case $k = b$.

Strong Induction

- Used when typically one cannot prove a result using mathematical induction.
- What is the difference? The basis step is the same. Inductive step is different.
- In strong induction we show that if $P(j)$ is true for all positive integers not exceeding k then $P(k+1)$ is true.
- That is, $P(j)$ is true for $j = 1, 2, \ldots, k$.
- Validity of strong induction can be shown using well-ordering principle. Mathematical induction, strong induction and well-ordering are all equivalent principles.
- Sometimes it maybe easier to see the proof using one of the principles but it can be equivalently proved by the other two as well.

Strong Induction

- T.P.T. $P(n)$ is true for all positive integers n, $P(n)$ is a propositional function, we complete two steps:
  - Basis Step: We verify that $P(1)$ is true.
  - Inductive Step: We show that $[P(1) \wedge P(2) \wedge \cdots \wedge P(k)] \rightarrow P(k+1)$ is true for all positive integers k.
- It is a more flexible proof technique because we have more statements $P(1), P(2), \ldots, P(k)$ to prove $P(k+1)$.
- Also called second principle of mathematical induction or complete induction.

07/01/2021

Example

- S.T. if n is an integer greater than 1 then n can be written as the product of primes. Fundamental Theorem of Arithmetic says it can be uniquely written as a product of primes.
- $P(n)$: n can be written as product of primes.
- Basis Step: $P(2)$ is true, because 2 can be written as the product of one prime itself.
- Inductive Step: Assume $P(j)$ is true for all integers j with $2 \leq j \leq k$. T.S.T. that $P(k+1)$ is true under this assumption, i.e. $k+1$ is the product of primes.
- Case i: $k+1$ is prime. Then $P(k+1)$ is true.
- Case ii: $k+1$ is composite, $k+1 = ab$, $2 \leq a \leq b \lneq k+1$.
  - Since a, b are at least 2 and not exceeding k, we can use the I.H. to write a and b as a product of primes.
  - Then $k+1$ can be written as a product of primes, namely the primes in the factorization of a and b.

Strong Induction

- It is difficult in the previous case to use only Mathematical Induction and prove the result.
- We can modify strong induction to start form a different base:

$$[P(b) \wedge P(b+1) \wedge \cdots \wedge P(k)] \to P(k+1)$$

  is true for every integer $k \geq b+j$.
- Some cases we can use either of the mathematical induction results (We will see that as in problem solving session).

Proof using Well-ordering Property

- Every nonempty set of nonnegative integers has a least element.
- Division algorithm: If a is an integer and d is a positive integer, then there are unique integers q and r with $0 \leq r \lneq d$ and $a = dq + r$.
- S be the set of nonnegative integers of the form $a - dq$, where q is an integer. The set is non-empty because we can choose q to be any negative integer with large absolute value.
- By well-ordering property S has a least element $r = a - dq_0$ for some q_0.
- r is nonnegative, Also $r < d$.

Proof using Well-ordering Property

- If it is not the case, there would be a smaller nonnegative element in S, $a - d(q_0 + 1)$. How?
- $r \geq d$, Since $a = dq_0 + r$, we have, $a - d(q_0 + 1) = (a - dq_0) - d = r - d \geq 0$.
- So we have integers q and r with $0 \leq r < d$. Remaining: Uniqueness.

Proof using Well-ordering Property

- In a round-robin tournament every player plays every other player exactly once and each match has a winner and a loser.
- Players $p_1, p_2, \ldots, p_m$ are said to form a cycle if p_1 beats p_2, p_2 beats p_3, $\ldots p_{m-1}$ beats p_m and p_m beats p_1.
- T.S.T if there is a cycle of length m $(m \geq 3)$ among the players *there must be a cycle of* 3 *of these players*.
- We assume that there is no cycle of three players. Because there is at least one cycle, the set of all positive integers n for which there is a cycle of length n is nonempty.
- By the well-ordering property, this set has a least element k, $k \gneq 3$. I.e. there is a cycle of players $p_1, p_2, p_3, \ldots, p_k$ and no shorter cycle exists.

Proof using Well-ordering Property

- Consider p_1, p_2, p_3. What are the outcomes of the match between p_1 and p_3? If p_3 beats p_1 then it follows that p_1, p_2, p_3 forms a cycle of length 3, so cannot happen.
- So p_1 beats p_3 but then that means if we omit p_2 from the cycle we get $p_1, p_3, p_4, \ldots, p_k$ of length $k-1$, a smaller cycle.
- Again a contradiction which means there must be a cycle of length 3.

Recursive Definitions

- Sometimes its easier to define an object in terms of itself. Recursive defintions.
- We already saw how sequences can defined using recursive definitions.
- The sequence $a_n = 2^n$ for $n = 0, 1, 2, \ldots$ can be defined recursively as $a_{n+1} = 2a_n$ and $a_0 = 1$.
- Defining a set recursively means : specifying initial elements (Basis step)and then providing a rule for constructing new elements from those (Recursive step).
- Proving results about recursively defined sets : structural induction.

Recursive/Inductive Definitions for $\mathbb{N}$ domains

- Basis step: Specify the value of the function at zero.
- Recursive step:Give a rule for finding its value at an integer from its values at smaller integers.
- A real-valued sequence $a_0, a_1, a_2, \ldots$ where $a_i \in \mathbb{R}$ is the same as a function from $\mathbb{N}$ to $\mathbb{R}$.
- Recursively defined functions are well-defined, for every positive integer in the domain there is one and only one mapping.
- Use mathematical induction to prove that a function F defined by specifying $F(0)$ and a rule for obtaining$F(n+1)$ from $F(n)$ is well-defined.

Well-defined comes in when we choose a different representative for the same element, then we should not get a different mapping.

Proof of Well-definedness

- To prove : Suppose $F(0) = G(0)$ and $F(n+1) = h(F(n))$ and $G(n+1) = h(G(n))$, where h is a function that gives the rule that relates $F(n+1)$ with $F(n)$s and the same rule for $G(i)$s.
- T.P.T. $F(n) = G(n)$ for all $n \in \mathbb{N}$, i.e. if $F(k) = G(k)$ then $F(k+1) = G(k+1)$.
- We have $F(0) = G(0)$ so basis step is proved.
- Inductive step: if $F(k) = G(k)$, that implies $h(F(k)) = h(G(k))$ which implies $F(k+1) = G(k+1)$.
- With both basis step and inductive step proved to be true, from principle of M.I. we have that $F(n) = G(n)$ for all n and thus F is well-defined.

Inequality based on Fibonacci Sequence

- $f_0, f_1, f_2, \ldots,$ defined as $f_0 = 0, f_1 = 1$ and $f_n = f_{n-1} + f_{n-2}$, for $n = 2, 3, 4, \ldots$.
- $P(n) : f_n > \alpha^{n-2}$ where $\alpha = (1 + \sqrt{5})/2$. S.T. for $n \geq 3$, $P(n)$ is true, $n \geq 3$.
- Basis Step: $\alpha < 2 = f_3$ and $\alpha^2 = (3 + \sqrt{5})/2 < 3 = f_4$ so $P(3)$ and $P(4)$ are true.
- Inductive Step: If $P(j)$ is true, i.e. $f_j > \alpha^{j-2}$ for all j $3 \leq j \leq k$, T.S.T $P(k+1)$ is true, $f_{k+1} > \alpha^{k-1}$.
- α is the solution of $x^2 - x - 1 = 0$ which means $\alpha^2 = \alpha + 1$.

$$\alpha^{k-1} = \alpha^2 \cdot \alpha^{k-3} = (\alpha + 1)\alpha^{k-3} = \alpha^{k-2} + \alpha^{k-3}.$$

Inequality based on Fibonacci Sequence

- Since $k \geq 4$, by I.H. we have $f_{k-1} > \alpha^{k-3}$, $f_k > \alpha^{k-2}$.
- $f_{k+1} = f_k + f_{k-1} > \alpha^{k-2} + \alpha^{k-3} = \alpha^{k-1}$. Thus $P(k+1)$ is true.
- Inductive step shows that whenever $k \geq 4$, $P(k+1)$ follows from the assumption that $P(j)$ is true for $3 \leq j \leq k$.
- Inductive step does not $P(3) \to P(4)$. So we had two things to show in the basis step.

Euclidean Algorithm

- Euclidean Algorithm: To compute gcd of two positive integers a and b, $a \geq b$.
- The following are the sequence of equations where $a = r_0$ and $b = r_1$. The $gcd(a, b) = r_n$.

$$
\begin{aligned}
r_0 &= r_1 q_1 + r_2 && 0 \leq r_2 < r_1, \\
r_1 &= r_2 q_2 + r_3 && 0 \leq r_3 < r_2, \\
&\ \ \vdots \\
r_{n-2} &= r_{n-1} q_{n-1} + r_n && 0 \leq r_n < r_{n-1}, \\
r_{n-1} &= r_n q_n.
\end{aligned}
$$

- Eventually a remainder of 0 is assured, n divisions are done, quotients $q_1, q_2, \ldots, q_{n-1}$ are all at least 1.
- We will see more when we learn modular arithmetic.

Euclidean Algorithm – An Example

$gcd(662, 414)$:

$$
\begin{aligned}
662 &= 414 \cdot 1 + 248 \\
414 &= 248 \cdot 1 + 166 \\
248 &= 166 \cdot 1 + 82 \\
166 &= 82 \cdot 2 + 2 \\
82 &= 2 \cdot 41.
\end{aligned}
$$

GCD is 2.

Analysis of Euclidean Algorithm

- **Lamé's Theorem:** Let a and b be positive integers with $a \geq b$. The number of divisions by the Euclidean algorithm to find $gcd(a, b)$ is less than or equal to five times the number of decimal digits in b.
  - Note that $q_n \geq 2$ since $r_n < r_{n-1}$.

$$
\begin{aligned}
r_n &\geq 1 = f_2 \\
r_{n-1} &\geq 2r_n \geq 2f_2 = f_3 \\
r_{n-2} &\geq r_{n-1} + r_n \geq f_3 + f_2 = f_4 \\
&\vdots \\
r_2 &\geq r_3 + r_4 \geq f_{n-1} + f_{n-2} = f_n \\
b = r_1 &\geq r_2 + r_3 \geq f_n + f_{n-1} = f_{n+1}.
\end{aligned}
$$

Analysis of Euclidean Algorithm

- If n divisions are used by the algorithm to find $gcd(a, b)$ with $a \geq b$ then $b \geq f_{n+1}$.
- We know from previous analysis $f_{n+1} > \alpha^{n-1}$ for $n > 2$ where $\alpha = (1 + \sqrt{5})/2$.
- So we have $b > \alpha^{n-1}$. $\log_{10}\alpha \approx 0.208 > 1/5$.

$$\log_{10} b > (n-1)\log_{10}\alpha > (n-1)/5.$$

- Thus $(n-1) < 5 \cdot \log_{10} b$.
- If b has k decimal digits then $b < 10^k$ and $\log_{10} b < k$.
- It follows $n - 1 < 5k$ and since k is an integer $n \leq 5k$. Done!
- Number of decimal digits in b is $\lfloor \log_{10} b \rfloor + 1 \leq \log_{10} b + 1$. $5(\log_{10} b + 1)$ is $\mathcal{O}(\log b)$ and so $\mathcal{O}(\log b)$ divisions are needed for the algorithm.

Recursively Defined Sets

- Not just functions, sets can have recursive definitions.
- Implicitly we assume exclusion rule, i.e. a recursively defined set contains nothing other than those elements specified in the basis step or generated by the recursive step.
- Recursive sets are commonly seen when we study strings which in turn is very important when we study formal languages.
- We define the alphabet Σ and the set of strings (a finite sequence of elements from Σ) over Σ as Σ^* which is defined recursively:
  - Basis Step: the empty string, $\lambda \in \Sigma^*$.
  - Recursive Step: if $w \in \Sigma^*$ and $x \in \Sigma$ then $wx \in \Sigma^*$.

Strings

- If $\Sigma = \{0, 1\}$ then Σ^* is the set of all bit strings which include λ (basis step),$0, 1$ formed by applying the first recursive step , $00, 01, 10, 11$ by applying recursive step second time and so on.
- Concatenation of two strings : Basis step: If $w \in \Sigma^*$ then $w \cdot \lambda = w$ and Recursive step: If $w_1, w_2 \in \Sigma^*$ and $x \in \Sigma$, then $w_1 \cdot (w_2 x) = (w_1 \cdot w_2)x$.
- Length of a string$l(w)$: $l(\lambda) = 0$, $l(wx) = l(w) + 1$ if $w \in \Sigma^*$ and $x \in \Sigma$.

Well Formed Logic Formulae

- Basis Step: T, F, p where p is a propositional variable are well-formed formulae.
- Recursive Step: If E and F are well formed then $(\neg E)$, $(E \wedge F)$, $(E \vee F)$, $(E \to F)$ and $(E \leftrightarrow F)$ are well-formed.
- Using this definition we can see for eg: $pq\wedge$, $p\neg \wedge g$ are not well-formed.
- Similar such a definition can be extended to well-formed arithmetic formulae.

Structural Induction

- To prove results about recursively defined sets it makes sense to use some form of mathematical induction: Structural Induction
- Basis Step Show that the result holds for all elements specified in the basis step of the recursive definition.
- Recursive Step Show thatif the statement is true for each of the elements used to construct new elements in the recursive step of the definition, the result holds for these new elements.
- Validity: Follows from mathematical induction. (Verify!)

Structural Induction – Examples

- S.T. every well-formed formula for compound propositions contains an equal number of left and right parentheses.
- Basis Step Each of the formula T, F, and p contains no parentheses, so they have an equal number of left and right parentheses.
- Recursive Step: Assume p and q are well-formed formulae each containing an equal number of left and right parentheses.
- I.e. number of left parentheses in p : l_p) and q: (l_q) and right parenthesis: r_p and r_q. We have $l_p = r_p$ and $l_q = r_q$.

Structural Induction – Examples

- We need to show that $(\neg p)$, $(p \wedge q)$, $(p \vee q)$, $(p \rightarrow q)$, $(p \leftrightarrow q)$ also contain equal number of left and right parentheses.
- The number of left parentheses in the first compound propositions equals $l_p + 1$ and others equals l$l_p + l_q + 1$. Similarly for r_p.
- Because $l_p = r_p$ and $l_q = r_q$ we have the result.

Structural Induction – Examples

- Use structural induction to show that $l(xy) = l(x) + l(y)$ where $x, y \in \Sigma^*$.
- Basis Step: To show that $P(\lambda)$ is true. T.S.T. $l(x\lambda) = l(x) + l(\lambda)$ for all $x \in \Sigma^*$.
- $l(x\lambda) = l(x) = l(x) + 0 = l(x) + l(\lambda)$ for every string x we have $P(\lambda)$ is true.
- Recursive Step: Assume $P(y)$ is true and S.T. this implies $P(ya)$ is true. I.e. T.S.T. $l(xya) = l(x) + l(ya)$, for every $a \in \Sigma$.
- $l(xya) = l(xy) + 1$ and $l(ya) = l(y) + 1$. (by definition of length)
- By I.H. $l(xy) = l(x) + l(y)$, so $l(xya) = l(x) + l(y) + 1 = l(x) + l(ya)$. Done!