COMP 250

Lecture 9

mathematical induction

Sept. 26, 2016
1 + 2 + 3 + \ldots + (n - 1) + n = \frac{n(n + 1)}{2}

How to prove such a statement?

(By “proof”, we mean a formal logical argument that convincingly shows the statement is true.)
\[ 1 + 2 + \ldots + (n - 1) + n \]

Write sum backwards:

\[ n + (n - 1) + \ldots + 2 + 1 \]

Adding up \( n \) pairs gives \( n \times (n + 1) \).

Dividing by 2 gives the result.

You should be 100% convinced by this proof.
Mathematical Induction

Consider statements of the form:

“For all $n \geq n_0$, $P(n)$” where $P(n)$ is either true or false for each $n$, and $n_0$ is a constant.

Mathematical induction is a general technique for proving such statements.
Mathematical induction requires proving two things:

**Base case:**

“\( P(n_0) \) is true.”

**Induction step:**

“For any \( k \geq n_0 \), if \( P(k) \) is true, then \( P(k+1) \) is also true.”
Base case:

“P(n0) is true.”

Induction step:

“For any k >= n0, if P(k) is true, then P(k+1) is also true.”

“P(k) is true” is called the “induction hypothesis”.
For any $k \geq n_0$, if $P(k)$ then $P(k+1)$.

**Base case:**
$P(n_0)$ is true.

**Induction step:**
For any $k \geq n_0$, if $P(k)$ then $P(k+1)$.

Thus,

For any $n \geq n_0$, $P(n)$ is true.
Statement: For all \( n \geq 1 \),

\[
1 + 2 + 3 + \ldots + (n - 1) + n = \frac{n(n + 1)}{2}
\]

Proof (base case, \( n=1 \)):

\[
1 = \frac{1(1+1)}{2} \quad \text{(true)}
\]
Proof of Induction Step:

\[(1 + 2 + 3 + \ldots + k) + k + 1\]

\[= \frac{k(k+1)}{2} + k + 1\]

Why? Because of the induction hypothesis

\[(1 + 2 + 3 + \ldots + k) = \frac{k(k + 1)}{2}\]
Proof of Induction Step:

\[
(1 + 2 + 3 + \ldots + k) + k + 1
\]

by induction hypothesis

\[
= \frac{k(k+1)}{2} + k + 1
\]

\[
= \left(\frac{k}{2} + 1\right)(k + 1)
\]

\[
= \frac{1}{2}(k + 2)(k + 1)
\]

Thus, \(P(k)\) is true implies \(P(k + 1)\) is true.
Example 2

Statement: For all $n \geq 3$, $2n + 1 < 2^n$. 
Statement: For all $n \geq 3$, $2n + 1 < 2^n$.

Note: $P(n)$ is false for $n = 1, 2$.

Proof (base case, $n = 3$):

$$2 \times 3 + 1 < 8$$

(true)
Proof of Induction Step:

We want to show that $P(k)$ implies $P(k+1)$.

\[ 2(k + 1) + 1 = 2k + 2 + 1 \]

\[ < 2^k + 2 \]

\[ < 2^k + 2^k, \quad \text{for } k \geq 2 \]

\[ = 2^{k+1} \]

This condition is stronger than we need.
For any $k \geq n_0$, if $P(k)$ then $P(k+1)$.

**Base case:**

$P(n_0)$ is true.

**Induction step:**

For any $k \geq n_0$, if $P(k)$ then $P(k+1)$.

Thus,

For any $n \geq n_0$, $P(n)$ is true.
Example 3

Statement: For all $n \geq 5$, $n^2 < 2^n$. 
Statement: For all $n \geq 5$, $n^2 < 2^n$.

Base case (n = 5):

$$25 < 32$$

Induction step:

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

by induction hypothesis

$$< 2^k + 2k + 1$$

by Example 2

$$< 2^k + 2^k$$

$$= 2^{k+1}$$
Example 4: Fibonacci Sequence

0, 1, 2, 3, 5, 8, 13, 21, 34, 55, ....

\[ F(0) = 0 \]
\[ F(1) = 1 \]
\[ F(n + 2) = F(n + 1) + F(n), \text{ for } n \geq 0. \]

Statement: For all \( n \geq 0 \), \( F(n) < 2^n \)
$F(0) = 0$
$F(1) = 1$
$F(n + 2) = F(n + 1) + F(n)$, for $n \geq 0$.

Base case(s):

$n = 0$: $0 < 2^0$ is true.

$n = 1$: $1 < 2^1$ is true.
\[ F(0) = 0 \]
\[ F(1) = 1 \]
\[ F(n + 2) = F(n + 1) + F(n), \quad \text{for} \quad n \geq 0. \]

**Induction step:**

\[ F(k + 1) = F(k) + F(k - 1) \]
\[ \quad < 2^k + 2^{k-1} \]
\[ \quad < 2^k + 2^k \]
\[ \quad = 2^{k+1} \]

by induction hypothesis
How to use mathematical induction to prove an algorithm is correct? e.g. insertion sort

```cpp
insertion_sort( list ){
    for k = 1 to n - 1 {
        elementK = list[k]
        i = k
        while (i > 0) and (list[i - 1] > elementK ){
            list[i] = list[i - 1]
            i = i -1
        }
        list[i] = elementK
    }
}
```
\[ P(k) = \]

"At the start of pass \( k \) through the for loop, list \([0, \ldots, k-1]\) contains the same elements as the original list, and these \( k \) elements are now sorted."

Q: What is the base case?

Q: What is the induction step?
Base case:

\[ P(1) = \]

"At the start of pass 1 through the for loop, list[0] contains the same element as the original list, and these elements are now sorted."
Induction step

if

"At the start of pass $k$ through the for loop, list[0, ... k-1] contains the same $k$ elements as the original list, and these elements are now sorted."

then

"At the start of pass $k+1$ through the for loop (i.e. at the end of pass $k$ through the for loop) list[0, ... k] contains the same elements as the original list, and these elements are now sorted."
You may have already been convinced that this algorithm was correct.

I would argue, however, that your proof (or intuition) uses mathematical induction. If not, then what *does* it use?