Principle of Mathematical Induction

It is a powerful proof technique to prove proposition \( P(n) \) is true for all positive integers. It consists of 2 steps:

**Basic step:** Verify that \( P(1) \) is true.

**Inductive step:** Show that \( P(k) \Rightarrow P(k+1) \) is true for all positive integers \( k \).

As a rule of reference, it is written as

\[
[ P(1) \land \forall k \, (P(k) \Rightarrow P(k+1))] \Rightarrow \forall n \, P(n)
\]

The assumption that \( P(k) \) is true is called inductive hypothesis.

**Example:** Show that \( \sum_{i=1}^{n} i = \frac{n(n+1)}{2} \)

**Proof:**

**Basic step:** \( P(1) \) is true because \( 1 = \frac{1(1+1)}{2} \).
Inductive step:

Assume the claim is true for an arbitrary positive integer \( k \), i.e.
\[
\sum_{i=1}^{k} i = \frac{k(k+1)}{2}. \quad \text{(note: } P(k) \text{)}
\]

Then, we must show the claim is true for \( k+1 \) under this assumption (hypothesis).

\[
\sum_{i=1}^{k+1} i = (k+1) + \sum_{i=1}^{k} i = \frac{2(k+1) + k(k+1)}{2} = \frac{(k+1)(k+2)}{2}. \quad \text{(note: } P(k+1) \text{)}
\]

Both base step and inductive step are completed. By mathematical induction,

\[
\sum_{i=1}^{n} i = 1 + 2 + \ldots + n = \frac{n(n+1)}{2}
\]
Example: Show \( \sum_{j=0}^{n} ar^j = \frac{ar^{n+1}-a}{r-1} \) when \( r \neq 1 \).

Proof: Let \( P(n) \) be the statement that \( \sum_{j=0}^{n} ar^j = \frac{ar^{n+1}-a}{r-1} \) is correct.

**Basis Step:** \( P(0) \) is true, because

\[
\frac{ar^{0+1}-a}{r-1} = \frac{a(r-1)}{r-1} = a
\]

**Inductive Step:**
Suppose \( P(k) \) is true, where \( k \geq 0 \), and consider \( P(k+1) \). Then,

\[
\left( \sum_{j=0}^{k} ar^j \right) + ar^{k+1} = \frac{ar^{k+1}-a}{r-1} + ar^{k+1}
\]

\[
= \frac{ar^{k+1}-a}{r-1} + \frac{ar^{k+2}-ar^{k+1}}{r-1}
\]

This gives that \( P(k+1) \) is also true.
This completes the inductive argument.
Example: \( n^3 - n \) is divisible by 3 whenever \( n \) is a positive integer.

Proof:

Basis Step: \( P(1) \) is true because \( 1^3 - 1 = 0 \) is divisible by 3.

Inductive Step: Assume \( P(k) \) is true.

Inductive hypothesis.

Consider \( P(k+1) \):

\[
(k+1)^3 - (k+1) = (k^3 + 3k^2 + 3k + 1) - (k+1)
\]

\[
= (k^3 - k) + 3(k^2 + k)
\]

divisible by 3 (by the hypothesis).

Because both terms are divisible by 3, the sum is also divisible by 3.
Creative uses of mathematical induction: Mathematical induction can often be used in unexpected ways.

Example: An odd number of people stand in a yard at mutually distinct distances. Each person throws a pie at his/her nearest neighbor, hitting this person. Show that there is at least one survivor (a person that is not hit by a pie).

Proof: P(n): There is a survivor for 2n+1 people playing this game.

Base Step: When n=1, there are 2n+1=3 people. Suppose the closest pair are A and B.

Then A and B throw pies at each other, and C is a survivor. Thus, P(1) is true.
Inductive Step: Suppose $P(k)$ is true, and consider $P(k+1)$. There are $2(k+1)+1 = 2k+3$ people.

Let $A$ and $B$ be the closest pair among these $2k+3$ people. Then, $A$ and $B$ throw pies at each other. There are 2 cases.

Case 1: Someone else throws a pie at either $A$ or $B$.

Then, altogether at least 3 pies are thrown at $A$ and $B$, and at most $(2k+3)-3 = 2k$ people pies are thrown at the remaining $2k+1$ people. This guarantees that at least 1 person is a survivor.

Case 2: No one else throws a pie at either $A$ or $B$. Because the distance between pairs of these $2k+1$ people are distinct, there is at least one survivor among these $2k+1$ people by $P(k)$. This completes Case 2 and proof.
Example:

Show that every $2^n \times 2^n$ checkerboard (n is a positive integer) with one square removed can be tiled using right triominoes.

Proof:

Basis Step: For $n=1$, the statement is obviously true.

Inductive Step:

Suppose the statement is true for $n=k$. Now consider $n=k+1$, i.e., a $2^{k+1} \times 2^{k+1}$ checkerboard with 1 square removed. Split this board into 4 $2^k \times 2^k$ boards. No square is removed from 3 of them, but the fourth has one square removed. By the hypothesis, this fourth board can be tiled. "Remove" one square from each of the other 3 around the center of the $2^{k+1} \times 2^{k+1}$ board using 1 right triominoes. By the hypothesis, these 3 $2^k \times 2^k$ boards can also be tiled.
Strong Induction

To prove that \( P(n) \) is true for all positive integers \( n \), we complete two steps:

**Basis Step:** Verify \( P(1) \) is true

**Inductive Step:** Show that

\[
[P(1) \land P(2) \land \cdots \land P(k)] \implies P(k+1)
\]

is true for all positive integers \( k \).

**Example:** Show that if \( n \) is an integer greater than 1, then \( n \) can be written as the product of primes.

**Proof:** \( P(n) \): \( n \) can be written as the product of primes.

**Basis Step:** \( P(2) \) is true (2 is the product of one prime, itself)

**Inductive Step:** Assume \( P(j) \) is true for all positive integers \( j \) with \( j \leq k \), and consider \( P(k+1) \). There are two cases.
Case 1: $k+1$ is prime.

Clearly $P(k+1)$ is true.

Case 2: $k+1$ is composite.

Then $k+1 = a \cdot b$, where $2 \leq a \leq b < k+1$.

By the inductive hypothesis, both $a$ and $b$ can be written as the product of primes. Thus, $P(k+1)$ is also true.

**Example:** Postage of 12 cents or more can be formed using 4-cent and 5-cent stamps.

**Proof:**

**Basis Step:**

- $12 = 4 + 4 + 4$ (P(12))
- $13 = 4 + 4 + 5$ (P(13))
- $14 = 4 + 5 + 5$ (P(14))
- $15 = 5 + 5 + 5$ (P(15))

**Inductive Step:** Assume that $P(j)$ is true for $12 \leq j \leq k$, where $k \geq 15$. Then $P(k-3)$ is true.

Consider $P(k+1)$. Since $k+1 = (k-3) + 4$ to form postage of $k+1$ cents, we only need to add 1 4-cent stamp to the stamps used for $k-3$ cents.
Recursively Defined Functions

A recursive function with the set of nonnegative integers as its domain is defined as follows:

**Basis Step**: Specify the value of the function at 0

**Recursive Step**: Give a rule for finding its value at an integer from its values at smaller integers.

**Example**:

\[ f(0) = 3 \]
\[ f(n+1) = 2f(n) + 3 \]

From this definition, we have

\[ f(1) = 2f(0) + 3 = 2 \times 3 + 3 = 9 \]
\[ f(2) = 2f(1) + 3 = 2 \times 9 + 3 = 21 \]
\[ f(3) = 2f(2) + 3 = 2 \times 21 + 3 = 45 \]
\[ \vdots \]
Example

\[ f(0) = 1 \]
\[ f(n+1) = (n+1)f(n) \]

Clearly, \( f(n) = n! \)

Example (Fibonacci numbers)

\[
\begin{align*}
  f_0 &= 0 \\
  f_1 &= 1 \quad \text{\textit{Basis}} \\
  f_n &= f_{n-1} + f_{n-2} \quad \text{\textit{Recursion}}
\end{align*}
\]

Many properties of recursive functions can be proved using mathematical induction.

Example: For \( n \geq 3 \), \( f_n > \alpha^{n-2} \), where \( \alpha = \frac{1 + \sqrt{5}}{2} \)

\[ P(n) \]

Proof:

\textit{Basis Step:} \( f_3 = 3 > \alpha \)
\[ f_4 = 3 > (3 + \sqrt{5})/2 = \alpha^2 \]

So \( P(3) \) and \( P(4) \) are true.

\textit{Inductive Step:} Assume \( P(j) \) is true for all \( j \) with \( 3 \leq j \leq k, k \geq 4 \)

(Strong induction)
Now consider $P(k+1)$; i.e. $f_{k+1} > \alpha^{k-1}$.

Since $\alpha^2 = \left( \frac{1 + \sqrt{5}}{2} \right)^2 = \frac{5 + \sqrt{5}}{2} = \alpha + 1$,

we have

\[
\alpha^{k+1} = \alpha^2 \cdot \alpha^{k-2} = \alpha \cdot \alpha^{k-2} + 1 \cdot \alpha^{k-2} = \alpha^k + \alpha^{k-2}.
\]

By the inductive hypothesis,

\[
f_{k-1} > \alpha^{k-3}, \quad f_k > \alpha^{k-2},
\]

we have

\[
f_{k+1} = f_k + f_{k-1} > \alpha^{k-2} + \alpha^{k-3} = \alpha^{k-2} (1 + \alpha^{-1}) = \alpha^{k-2} \cdot \alpha = \alpha^{k-1}
\]

Therefore, $P(k+1)$ is true. (1 + $\alpha^{-1} = \alpha$ because $\alpha^2 = \alpha + 1$)
Recursively Defined Sets and Structures

Example: A subset $S$ of the set of integers is defined by

**Basis Step:** $3 \in S$.

**Recursive Step:** If $x \in S$ and $y \in S$, then $x + y \in S$.

$S$ is the set of all positive multiples of 3.

Example: The set $\Sigma^*$ of strings over the alphabet $\Sigma$ is defined by

**Basis Step:** $\lambda \in \Sigma^*$ ( $\lambda$ is the empty string containing no symbols of $\Sigma$).

**Recursive Step:**

If $w \in \Sigma^*$ and $x \in \Sigma$, then $wx \in \Sigma^*$.
Recursive definition of rooted trees:

**Basic Step:** A single vertex \( r \) is a rooted tree.

**Recursive Step:** Let \( T_1, T_2, \ldots, T_n \) be disjoint rooted trees with roots \( r_1, r_2, \ldots, r_n \) respectively. Then the graph formed by introducing a new vertex \( r \) and adding an edge to each of \( r_1, r_2, \ldots, r_n \) is also a rooted tree.

Recursive definition of extended binary trees:

**Basic Step:** An empty set is an extended binary tree.

**Recursive Step:** If \( T_1 \) and \( T_2 \) are disjoint extended binary trees, there is an extended binary tree, denoted by \( T_1 \cdot T_2 \), consisting of a root \( r \) with edges to the root of the left (right) subtree \( T_1 \) (\( T_2 \)) if it is nonempty.
Recursive definition of full binary trees:

Basis Step: A single vertex \( r \) is a full binary tree.

Recursive Step: Let \( T_1 \) and \( T_2 \) be disjoint full binary trees. There is a full binary tree, denoted by \( T = T_1 \cdot T_2 \), consisting of a root \( r \) with edges connecting \( r \) to the left (respectively, right) subtree \( T_1 \) (respectively, \( T_2 \)).

(Note: Every full binary tree is "extended", but the converse is not true.)
**Structural Induction**

(for proving results about recursively defined sets and structures)

**Basis Step:**
Show that the result holds for all elements specified in the basis step of the recursive definition.

**Recursive Step:**
Show that if 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 those new elements.

**Example:**

$h(T)$: height of a full binary tree $T$

- $h(T) = 0$, if $T$ has only 1 vertex
- $h(T) = \max \{ h(T_1), h(T_2) \} + 1$, if $T = T_1 \cdot T_2$

$n(T)$: number of vertices in a full binary tree $T$

- $n(T) = 1$, if $T$ has only 1 vertex
- $n(T) = n(T_1) + n(T_2) + 1$, if $T = T_1 \cdot T_2$
Show that $n(T) \leq 2^{h(T)+1} - 1$, if $T$ is a full binary tree.

**Proof:**

**Basic Step:** The result is true for $T$ with a single vertex because $n(T) = 1$, $h(T) = 0$, and $n(T) = 1 \leq 2^{0+1} - 1 = 1$.

**Inductive Step:** Suppose the result holds for $T_1$ and $T_2$, i.e.

$n(T_1) \leq 2^{h(T_1)+1} - 1,$

$n(T_2) \leq 2^{h(T_2)+1} - 1$. Consider $T = T_1 \cdot T_2$.

We know $n(T) = 1 + n(T_1) + n(T_2)$

$h(T) = 1 + \max \{h(T_1), h(T_2)\}$.

Then

$n(T) = 1 + n(T_1) + n(T_2)$

$\leq \left(2^{h(T_1)+1} - 1\right) + \left(2^{h(T_2)+1} - 1\right) + 1$

$= 2^{h(T_1)+1} + 2^{h(T_2)+1} - 1$

$\leq 2 \cdot \max \{2^{h(T_1)+1}, 2^{h(T_2)+1}\} - 1$

$= 2 \cdot 2 \cdot \max\{h(T_1), h(T_2)\} + 1$

$= 2 \cdot 2^{h(T)+1} - 1$

$= 2^{h(T)+1} - 1.$
Recursive Algorithms

1. \[ n! = \begin{cases} 1, & n = 0 \\ n \cdot (n-1)!, & n > 0 \end{cases} \]

ALGORITHM 1 A Recursive Algorithm for Computing \( n! \).

procedure factorial(n: nonnegative integer)
if \( n = 0 \) then factorial(n) := 1
else factorial(n) := n \cdot factorial(n - 1)

2. \[ b^n \mod m = \begin{cases} 1, & n = 0 \\ (b^{\frac{n}{2}} \mod m)^2 \mod m, & n \text{ is even} \\ ((b^{\frac{n-1}{2}} \mod m)^2 \mod m \cdot b \mod m) \mod m, & n \text{ is odd} \end{cases} \]

ALGORITHM 3 Recursive Modular Exponentiation.

procedure mpower(b, n, m: integers with \( m \geq 2, n \geq 0 \))
if \( n = 0 \) then
mpower(b, n, m) := 1
else if \( n \) is even then
mpower(b, n, m) := mpower(b, n/2, m)^2 \mod m
else
mpower(b, n, m) := (mpower(b, \lfloor n/2 \rfloor, m)^2 \mod m \cdot b \mod m) \mod m
{mpower(b, n, m) = b^n \mod m}
\[
gcd(a, b) = \begin{cases} 
b, & \text{if } a = 0 \\
gcd(b \mod a, a) & \text{else}
\end{cases}
\]

**Algorithm 4** A Recursive Algorithm for Computing gcd(a, b).

```
procedure gcd(a, b: nonnegative integers with \(a < b\))
if \(a = 0\) then gcd(a, b) := b
else gcd(a, b) := gcd(b \mod a, a)
```

**Figure 2** The Merge Sort of 8, 2, 4, 6, 9, 7, 10, 1, 5, 3.
ALGORITHM 9 A Recursive Merge Sort.

procedure mergesort($L = a_1, \ldots, a_n$)
if $n > 1$ then
    $m := \lfloor n/2 \rfloor$
    $L_1 := a_1, a_2, \ldots, a_m$
    $L_2 := a_{m+1}, a_{m+2}, \ldots, a_n$
    $L := \text{merge}(\text{mergesort}(L_1), \text{mergesort}(L_2))$
{\small $L$ is now sorted into elements in nondecreasing order\}

ALGORITHM 10 Merging Two Lists.

procedure merge($L_1, L_2$: sorted lists)
$L := \text{empty list}$
while $L_1$ and $L_2$ are both nonempty
begin
    remove smaller of first element of $L_1$ and $L_2$ from the list it is
    in and put it at the right end of $L$
    if removal of this element makes one list empty then remove
    all elements from the other list and append them to $L$
end {\small $L$ is the merged list with elements in increasing order\}