Induction and Recursion

- **Principle of Mathematical Induction**: To prove that $P(n)$ is true for all positive integers n, where $P(n)$ is a propositional function, we complete two steps.

Basic Step: We verify that $P(1)$ is true.

Inductive Step: We show ~~that~~ the conditional statement $P(k) \rightarrow P(k+1)$ is true for all positive integers k.

- To complete the inductive ~~proof~~ step of a proof using the principle of mathematical induction, we assume that $P(k)$ is true for an arbitary positive integer k and show that under this assumption, $P(k+1)$ must also be true. The assumption that $P(k)$ is true is called the inductive hypothesis. Once we complete both steps in a proof by mathematical induction, we have shown that ~~$\forall n P(n)$ is true~~ $P(n)$ is true for all positive integers, that is, we have shown that $\forall n P(n)$ is true where the quantification is over the set of positive integers. In the inductive step, we show that $\forall k (P(k) \rightarrow P(k+1))$ is true, where again, the domain is the set of positive integers.
- Expressed as a rule of inference, this proof technique can be stated as

$$[P(1) \wedge \forall k (P(k) \rightarrow P(k+1))] \rightarrow \forall n P(n)$$

when domain is the set of positive integers.

Ex. Example. ~~Show~~ Show that if n is a positive integer, then.

$$1+2+\cdots+n = \frac{n(n+1)}{2}$$

Solution: Let $P(n)$ be the proposition that the sum of the first n positive integers is $\frac{n(n+1)}{2}$. We must do two things to prove that $P(n)$ is true for $n = 1, 2, 3 \ldots$ Namely, we must show that $P(1)$ is true and that the conditional statement $P(k)$ implies $P(k+1)$ is true for $k = 1, 2, 3 \ldots$.

Basic step: $P(1)$ is true, because $1 = \frac{1(1+1)}{2}$

Inductive step: For the inductive hypothesis we assume that $P(k)$ holds for an arbitary positive integer k. That is, we assume that

$$1+2+\cdots+k = \frac{k(k+1)}{2}$$

Under this assumption, it must be shown that $P(k+1)$ is true, namely that

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

is also true.

When we add $k+1$ to both sides of equation in $P(k)$, we obtain

$$1+2+3\cdots+k+(k+1) = \frac{k(k+1)}{2}+(k+1)$$

$$= \frac{(k+1)(k+2)}{2}$$

The last ~~step~~ equation shows that $P(k+1)$ is true under the assumption ~~to~~ that $P(k)$ is true. This completes the inductive step.

<u>Example</u> Use mathematical induction to prove that n^3-n is divisible by 3 whenever n is a positive integer.

<u>Solution:</u>

Let $P(n)$ denotes the proposition: n^3-n is divisible by 3.

Basic step: Let $P(n)$ is true for $n=1$

i.e $P(1)$: $1^3-1=0$, is divisible by three.

Inductive step: Let $P(n)$ is true for $n=k$

i.e $P(k)$: k^3-k is divisible by 3.

Using inductive hypothesis, we try to show that $P(n)$ is true for $n=k+1$

$$P(k+1) = (k+1)^3-(k+1)$$

$$= k^3+3k^2+3k+\cancel{1}-k-\cancel{1}$$

$$P(k+1) = (k^3-k)+3(k^2+k)$$

Since, both terms in this sum are divisible by 3. It follows that $(k+1)^3-(k+1)$ is divisible by 3. Thus by principle of mathematical induction, n^3-n is divisible by 3, whenever n is a positive integer.

Example: Prove by mathematical induction that

$$1+3+5+\cdots+(2n-1)=n^2$$

[The sum of first n positive odd integers is n^2].

Solution:

Let $P(n) = 1+3+5+\cdots+(2n-1)=n^2$

1. Basic step: For $n=1$, we have

$P(1) = 1 = (1)^2$. Hence, $P(1)$ is true.

2. Induction Hypothesis:

Assume $P(k)$ is true i.e.

$P(k) = 1+3+5+\cdots+(2k-1)=k^2$.

[Note, that k^{th} positive integer is $(2k-1)$, $k=1,2,3\ldots$]

3. Inductive. Step:

Now we wish to show $P(k+1)$ is true. So adding $(2k+1)$ on both sides of $P(n)$ then

$$1+3+5+\cdots+(2k-1)+(2k+1)=k^2+(2k+1)$$
$$=(k+1)^2$$

$\therefore$ $P(k+1)=(k+1)^2$, is true.

Thus, by mathematical induction $P(n)$ is true for all n.

Example Prove, for $n \geq 1$ and $a \neq 1$, that

$$1 + a + a^2 + \cdots + a^n = \frac{a^{n+1} - 1}{a-1}$$

Solution:

Let $P(n) = 1 + a + a^2 + \cdots + a^n = \frac{a^{n+1} - 1}{a-1}$

1. Basic Step: For $n=1$, we have

$$P(1) = 1 + a \stackrel{?}{=} \frac{a^{1+1} - 1}{a-1} = \frac{a^2-1}{a-1} = \frac{(a-1)(a+1)}{a-1}$$

$$\therefore 1 + a = a + 1$$

So, $P(1)$ is true.

Induction Hypothesis: Suppose $P(k)$ is true.

$$P(k) = 1 + a + a^2 + \cdots + a^k = \frac{a^{k+1} - 1}{a-1}$$

Inductive step:

Now we wish to show $P(k+1)$ is true, for this we add a^{k+1} to both sides of $P(n)$, then

$$1 + a + a^2 + a^3 + \cdots + a^k + a^{k+1} = \frac{a^{k+1} - 1}{a-1} + a^{k+1}$$

$$= \frac{a^{k+1} - 1 + a^{k+1}(a-1)}{a-1}$$

$$= \frac{a^{k+1} - 1 + a^{k+2} - a^{k+1}}{a-1}$$

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

$P(k+1) = \frac{a^{(k+1)+1} - 1}{a-1}$ is true.

Strong Induction:

To prove that $P(n)$ is true for all positive integers n, where $P(n)$ is a propositional function, we complete two steps:

Basic step: We verify that the proposition $P(1)$ is true.

Inductive step: We show that conditional statement $[P(1) \wedge P(2) \wedge \cdots \wedge P(k)] \rightarrow P(k+1)$ 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.

Solution: Let $P(n)$ be the proposition than n can be written as product of primes.

Basic Step: $P(2)$ is true, because 2 can be written as the product of one prime, itself.

Inductive step: The inductive hypothesis is the assumption that $P(j)$ is true for all positive integers j with $j \leq k$, that is, the assumption that j can be written as the product of primes whenever j is a positive integer at least 2 and not exceeding k. To complete the inductive step, it must be shown that $P(k+1)$ is true under this assumption, that is, that $k+1$ is the product of primes.

There are two cases to consider, namely, when k+1 is prime and when k+1 is composite. If k+1 is ~~composite~~ is prime, we ~~cannot~~ immediately see that P(k+1) is true.

Otherwise, k+1 is composite and can be written as product of two prime integers a and b with $2 \leq a \leq b < k+1$. By the inductive hypothesis, both a and b can be written as product of primes.

Thus, if k+1 is composite, it can be written as the product of primes, namely, those primes in the factorization of a and b.

Example: Every amount of postage stamp of 12 cents or more can be made with just 4 and 5 cents stamps.

Solution

Follow the steps.

- State base step
- Prove base step
- State Inductive step
- Prove Inductive step
- Invoke the Principle of strong mathematical induction.

Base step:

$12 = a \times 4 + b \times 5$

~~$13 = a \times 4 + b \times$~~

$13 = c \times 4 + d \times 5$

$14 = e4 + f5$

$15 = g \cdot 4 + h \cdot 5$

a, b, c, d, e, f, g and h are ~~positive~~ integers. (non-negative)

· Proof of the Base step.

· $12 = 3 \times 4 + 0 \times 5$

· $13 = 2 \times 4 + 1 \times 5$

· $14 = 1 \times 4 + 2 \times 5$

· $15 = 0 \times 4 + 3 \times 5$

· Statement of the inductive step.

If for every i, $12 \leq i \leq k$ (where $k \geq 15$) there is a and b so that

$$i = a \times 4 + b \times 5.$$

then there is c and d such that

$k+1 = c \times 4 + d \times 5$, is true.

· Proof of inductive step.

· Want $k+1 = c \times 4 + d \times 5$

· $k+1 = 4 + (k-3)$ but since $k+1$ is atleast 15,

$\therefore$ $k-3$ is atleast 12.

Thus, $k-3 = a \times 4 + b \times 5$ by inductive hypothesis.

· $k+1 = 4 + (k-3) = 4 + a \times 4 + b \times 5 = (a+1) \times 4 + b \times 5$.

Thus, $k+1$ can be ~~ofe~~ made using 4 and 5 cent stamps.

- Invoke the principle of strong induction.
  - Since the base step and inductive step are both true by the principle of strong induction all amounts of postage stamps $n \geq 12$ can be obtained using four and five cent stamps.

Recursively Defined Functions:

We use two steps to define a function with the set of non-negative integers as its domain.

Basic step: Specify the value of function at zero.

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

- Such a definition is called a recursive or inductive definition.

Example: Suppose that f is defined recursively by

$f(0) = 3$

$f(n+1) = 2f(n) + 3$

find, $f(1)$, $f(2)$, $f(3)$ and $f(4)$

Example:

0 1 1 2 3 5 8 13 21 .....

$$f_n = f_{n-1} + f_{n-2} \qquad n \geq 2$$

[Fibonacci numbers

$f_0 = 0$

$f_1 = 1$

Example 3: Find an explicit formula for the Fibonacci numbers, the recursion relation

$$f_{n-1} + f_{n-2} = f_n$$

and $f_0 = 0$

$f_1 = 1$

Example 4: Give an recursive definition for the factorial function $F(n) = n!$.

Solution: We can define the factorial function by specifying the initial value of this function namely, $F(0) = 1$, and giving the rule for finding $F(n+1)$ from $F(n)$. This is obtained by noting that $(n+1)!$ is computed from $n!$ by multiplying by $n+1$. Hence, the desired rule is.

$$F(n+1) = (n+1)F(n).$$

To determine a value of the factorial function, such as $F(5) = 5!$, from the recursive definition it is necessary to use the rule that shows how to express $F(n+1)$ in terms of $F(n)$ several times.

$$F(5) = F(4)\cdot 5 = 5\cdot 4\, F(3) = 5\cdot 4\cdot 3\cdot F(2) = 5\cdot 4\cdot 3\cdot 2\cdot F(1)$$
$$= 5\cdot 4\cdot 3\cdot 2\cdot 1\cdot F(0) = 5\cdot 4\cdot 3\cdot 2\cdot 1\cdot 1 = 120.$$

- Recursively defined functions are well defined. That is, for every positive integer, the value of the function at this integer is determined in an unambiguous way.

Example Give a recursive definition of a^n, where a is a non-zero real number and n is a non-negative integer.

Solution. The recursive definition contains two parts.

First $a^0 = 1$. Then the ~~rule~~ rule for finding a^{n+1} from a^n, namely, $a^{n+1} = a\cdot a^n$, for $n = 0, 1, 2, 3 \ldots$ is given. These two equations uniquely define a^n for all non-negative integers n.

Recursively defined Sets and Structures:

- Recursive definitions of sets have two parts, a basic step and a recursive step.

In the basic step, an initial condition of elements is ~~spect~~ specified.

In the recursive step, rules of forming new elements in the step from those already known to be in the set are provided.

Recursive definitions may also include an exclusion rule, which specifies that a recursively defined set contains nothing other than those elements specified in the basic step or generated by applications of the recursive step.

Example: Consider the subset S of the set of integers defined by.

Basic step: $3 \in S$

Recursive step: If $x \in S$ and $y \in S$, then $x+y \in S$

The new elements found to be in S are 3 by the ~~the~~ basic step, $3+3=6$. at the first application of the recursive step, $3+6=6+3=9$, and $6+6=12$, at the second application of the recursive step and so on.

Structural Induction:

Basic Step: Show the result holds for all elements specified in the basis step of the recursive step.

~~Recur~~

Recursive Step: Show that if the statement is true for all elements ~~set~~ used to construct new elements in the recursive step of the definition, the results holds for these new elements.

Definition: We define the height $h(T)$ of a full binary tree T recursively.

Basic step: The height of full binary tree containing ~~con~~ only a root r is $h(T)=0$.

Recursive Step: If T_1 and T_2 are full binary trees, then the full binary tree $T=T_1 \cdot T_2$ has height

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

- If we let $n(T)$ denote the number of vertices in a full binary tree, we observe that $n(T)$ satisfies the following recursive formula:

Basic step: The number of vertices $n(T)$ of the full binary tree consisting only a root r is $h(T) = 0$.

Recursive step: If T_1 and T_2 are full binary trees, then the number of vertices of the full binary tree, $T = T_1 \cdot T_2$ is $n(T) = 1 + n(T_1) + n(T_2)$.

Theorem: If T is a full binary tree T, then $n(T) \leq 2^{h(T)+1} - 1$

Proof: We prove this inequality using structural induction.

Basic Step: For the full binary tree consisting of just the root r, the result is true because $n(T) = 1$ and $h(T) = 0$

so that, $n(T) = 1 \leq 2^{0+1} - 1 = 1$

i.e $1 \leq 1$ (True)

Inductive Step: For the inductive hypothesis we assume that $n(T_1) \leq 2h^{n(T_1)+1} - 1$

and $n(T_2) \leq 2h^{n(T_2)+1} - 1$

Whenever, T_1 and T_2 are full binary trees. By recursive formula for $n(T)$ and $h(T)$ we have

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

and $h(T) = 1 + \max(h(T_1), h(T_2))$

Algorithm 5: A Recursive Linear search Algorithm

```
Procedure (i, j, x: i, j, x integers, 1≤i≤n, 1≤j≤n)
if a_i = x then
        location := i
else if i=j then
        location := 0
else
        search (i+1, j, x)
```

Algorithm 6: A recursive Binary Search Algorithm.

```
Procedure binary_search (i, j, x: i, j, x integers,
                        1≤i≤n, 1≤j≤n)
m := ⌊(i+j)/2⌋
if x=a_m then
        location := m
else if (x<a_m and i<m) then
        binary_search (x, i, m-1)
else if (x>a_m and j>m) then
        binary_search (x, m+1, j)
else location := 0.
```

Algorithm 7: A Recursive Algorithm for Fibonacci Numbers.

procedure fibbonacci (n: nonnegative integer)
if n=0 then fibonacci (0):= 0.
else if n=1 the fibonacci (1):= 1
else fibonacci (n):= fibonacci (n-1) + fibonacci (n-2)

• Iteration:
Instead of successively reducing the computation to the evaluation of the function at small integers, we can start with a value of function at one or more integers, the base cases and successively apply the recursive definition to find the values of the function at successive large integers. Such procedure is called iterative.

~~Algorithm 8: An Iterative~~ A

Algorithm 8 An iterative Algorithm for Computing Fibonacci Numbers

```
Procedure iterative fibonacci (n: non negative integer)
 if n=0 then y=0
 else
 begin
        x=0
        y=1
        for i=1 to n-1
        begin
          z=x+y
          x=y
          y=z
      end
  end
 {y is the nth Fibonacci number}
```

The Merge Sort

Algorithm 9 A Recursive Merge Sort

```
 Procedure mergesort (L=a1,...,an)
 if n>1 then
   m= ⌊n/2⌋
   L1 = a1,a2,...,am
   L2 = am+1, am+2, ..., an
   L = merge (mergesort(L1), mergesort(L2))
   {L is now sorted into elements of nondecreasing
                                        order}
```

Algorithm 10: Merging Two lists

procedure merge (L_1, L_2: sorted lists)
L := empty list
while L_1 and L_2 are both non empty
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 { L is the merged list with elements in increasing orders}.

Book page: 325
figure 2: