# 4.1 MATHEMATICAL INDUCTION

Many theorems state that $P(n)$ is true for all positive integers $n$, where $P(n)$ is a propositional function, such as the statement that $1 + 2 + \cdots + n = n(n + 1)/2$ or the statement that $n \leq 2^n$. Mathematical induction is a technique for proving theorems of this kind. In other words, mathematical induction is used to prove propositions of the form $\forall n\ P(n)$, where the universe of discourse is the set of positive integers.

A proof by mathematical induction that $P(n)$ is true for every positive integer $n$ consists of two steps:

*BASIS STEP:* The proposition $P(1)$ is shown to be true.

*INDUCTIVE STEP:* The implication $P(k) \to P(k + 1)$ is shown to be true for every positive integer $k$.

Here, the statement $P(k)$ for a fixed positive integer $k$ is called the **inductive hypothesis.** When we complete both steps of a proof by mathematical induction, we have proved that $P(n)$ is true for all positive integers $n$; that is, we have shown that $\forall n P(n)$ is true.

Expressed as a rule of inference, this proof technique can be stated as

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

Since mathematical induction is such an important technique, it is worthwhile to explain in detail the steps of a proof using this technique. The first thing we do to prove that $P(n)$ is true for all positive integers $n$ is to show that $P(1)$ is true. This amounts to showing that the particular statement obtained when $n$ is replaced by 1 in $P(n)$ is true. Then we must show that $P(k) \to P(k + 1)$ is true for every positive integer $k$. To prove that this implication is true for every positive integer $k$, we need to show that $P(k + 1)$ cannot be false when $P(k)$ is true. This can be accomplished by assuming that $P(k)$ is true and showing that *under this hypothesis* $P(k + 1)$ must also be true.

***Remark:*** In a proof by mathematical induction it is *not* assumed that $P(k)$ is true for all positive integers! It is only shown that *if it is assumed* that $P(k)$ is true, then $P(k + 1)$ is also true. Thus, a proof by mathematical induction is not a case of begging the question, or circular reasoning.

When we use mathematical induction to prove a theorem, we first show that $P(1)$ is true. Then we know that $P(2)$ is true, since $P(1)$ implies $P(2)$. Further, we know that $P(3)$ is true, since $P(2)$ implies $P(3)$. Continuing along these lines, we see that $P(n)$ is true, for every positive integer $n$.

**Ex.** P(n): $2^n < n!$ for $n \geq 4$

**Proof:**

**Basis step:** P(4): $2^4 = 16 < 24 = 4!$ true

**Inductive step:** Assume P(k) is true. $(k \geq 4)$

[We want to show that $2^{k+1} < (k+1)!$ ]

**Multiply both sides by 2**

$$\begin{aligned} 2 \cdot 2^k &< 2 \cdot k! \\ &< (k+1) \cdot k! \\ &= (k+1)! \end{aligned}$$

q.e.d.

**Ex.** P(n): $1 + 2 + 2^2 + \ldots + 2^n = 2^{n+1} - 1$ $n \geq 0$

**Proof:**

**Basis step:** P(0): $2^0 = 1 = 1 = 2^{0+1} - 1$ true

**Inductive step:** Assume P(k) is true. $(k \geq 0)$

[We want to show that $1 + 2 + 2^2 + \ldots + 2^k + 2^{k+1} = 2^{(k+1)+1} - 1 = 2^{k+2} - 1$]

$$\begin{aligned} 1 + 2 + 2^2 + \ldots 2^k + 2^{k+1} &= (1 + 2 + 2^2 + \ldots + 2^k) + 2^{k+1} \\ &= 2^{k+1} - 1 + 2^{k+1} \\ &= 2 \cdot 2^{k+1} - 1 \\ &= 2^{k+2} - 1 \end{aligned}$$

q.e.d.

**Ex.** P(n): $4n < (n^2 - 7)$ for $n \geq 6$

**Proof.**

**Basis step:** P(6): $24 < 29$ true.

**Inductive step:** Assume P(k) is true. $(k \geq 6)$

[We want to show that $4(k+1) < ((k+1)^2 - 7)$ ]

$$\begin{aligned} 4k &< (k^2 - 7) \\ 4k + 4 &< (k^2 - 7) + 4 < (k^2 - 7) + (2k + 1) \\ &\qquad\qquad\qquad = k^2 + 2k + 1 - 7 \\ 4(k+1) &\qquad\qquad\qquad = (k+1)^2 - 7 \end{aligned}$$

q.e.d.

**Note.**

| n | 4n | | $n^2 - 7$ |
|---|---|---|---|
| 1 | 4 | < | -6 |
| 2 | 8 | < | -3 |
| 3 | 12 | < | 2 |
| 4 | 16 | < | 9 |
| 5 | 20 | < | 18 |
| 6 | 24 | > | 29 |
| 7 | 28 | > | 42 |
| 8 | 32 | > | 57 |
| ⋮ | ⋮ | > | ⋮ |

# 4.3 Recursive Definitions

## (1) Recursive Function

$$a_n = 2^n, \quad n = 0,1,2\ldots$$

or

$$\begin{cases} a_0 = 1 & \text{// boundary condition //} \\ a_{n+1} = 2 \cdot a_n & \text{// recursive step //} \end{cases}$$

**Ex 1.** $\begin{cases} f(0) = 3 \\ f(n+1) = 2 \cdot f(n) + 3 \end{cases}$

**Ans.** 3, 9, 21, 45, 93, …

**Ex 2.** $F(n) = n!$

$$\begin{cases} F(0) = F(1) = 1 \\ F(n) = (n-1)\, F(n-1), \quad n > 1 \end{cases}$$

**Ex 4.** $f(n) = \sum_{k=0}^{n} a_k$

$$f(n) = \begin{cases} a_0 & n = 0 \\ f(n-1) + a_n, & n \geq 1 \end{cases}$$

**Def 1.** **Fibonacci numbers**

$$\begin{cases} f_0 = 0 \\ f_1 = 1 \\ f_n = f_{n-1} + f_{n-2}, \quad n >= 2 \end{cases}$$

0, 1, 1, 2, 3, 5, 8, 13, 21, …

## (2) Recursively Defined Sets and Structures

**Ex 7.** $\begin{cases} 3 \in S \\ \text{if } x \in S \text{ and } y \in S, \text{ then } x+y \in S \end{cases}$

$S = \{3, 6, 9, 12, 15, \ldots\}$

**Def 2.** The set $\Sigma^*$ of strings over the alphabet $\Sigma$ can be defined recursively by

$\begin{cases} \text{basis string: } \lambda \in \Sigma^* \\ \text{recursive step: if } w \in \Sigma^* \text{ and } x \in \Sigma, \text{ then } wx \in \Sigma^* \end{cases}$

**Ex 8.** $\Sigma = \{0, 1\}$

$\Sigma^* = \{\lambda, 0, 1, 00, 01, 10, 11, \ldots\}$

**Def 3.** Two strings can be combined via the operation of concatenation. Let $\Sigma$ be a set of symbols and $\Sigma^*$ the set of strings formed from symbols in $\Sigma$. We define the concatenation of two strings, denoted by ., recursively as follows.

basis step: If $w \in \Sigma^*$, then $w \cdot \lambda = w$
recursive step: If $w_1 \in \Sigma^*$, and $w_2 \in \Sigma^*$ and $x \in \Sigma$, then

$$w_1 \cdot (w_2 x) = (w_1 \cdot w_2)x.$$

$w_1$ = hot  $w_2$ = dog → $w_1 \cdot w_2$ = hotdog

**Ex 9. Length of a string: l(w)**

$$\begin{cases} l(\lambda) = 0 \\ l(wx) = l(\omega) + 1, \quad \omega \in \Sigma^* \text{ and } x \in \Sigma \end{cases}$$

**Ex 10. Well-formed formula for compound proposition**

**T, F, variables, {¬, ∧, v, →, ↔}**

**basis: T, F, p, where p is a propositional variable, are well-formed formula**

**recursive step: If E and F are well-formed formula, then (¬E),(E^F),(EvF),(E→F), and (E↔F) are well-formed formula.**

**Show that the following is a well-formed formula.**

**((p v q) → (q ^ F))**
**(q v (p v q))**
**((p → F) → T)**

**Ex 11. Well-formed expression consisting of variables, numerals, and operators (+, -, *, /, ↑)**

**basis: x is a well-formed expression if x is a numeral or variable**
**recursive: if A and B are well-formed expression, then so are (A+B),(A-B),(A*B), (A/B), (A↑B)**

**Ex.** ((x + 3) – (3 * y))

---

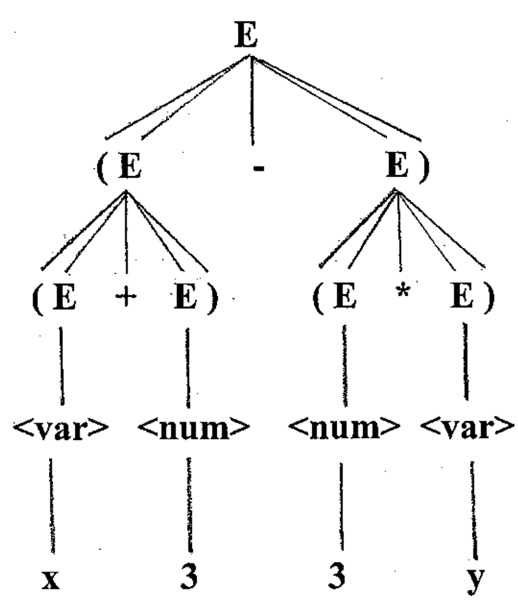
**E → (E + E) | (E – E) | (E * E) | (E / E) | (E↑E)**

**E → <num> | <var>**

**<num> → 0 | 1 | 2 | ... | 9**

**<var> → a | b | c | ... | z**

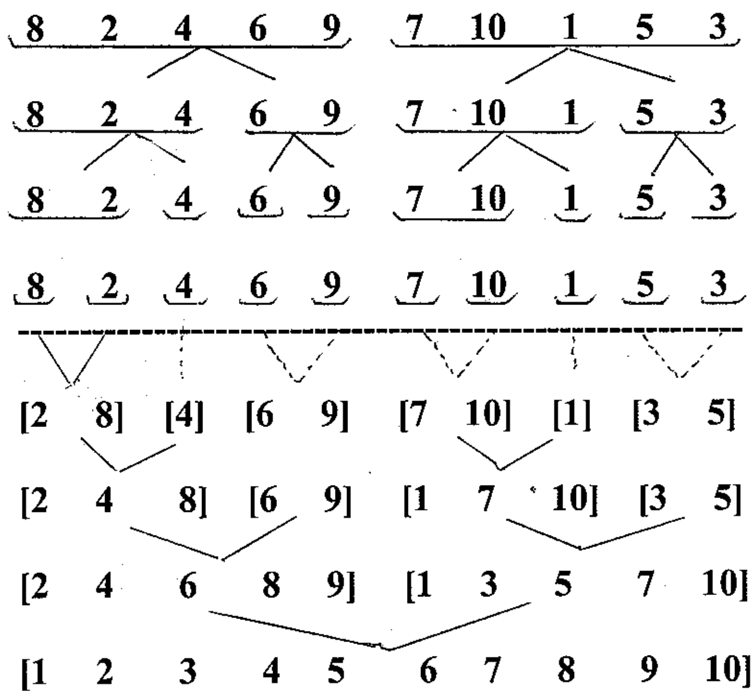
**context-free grammar**




**parse tree**

## 4.4 Recursive Algorithm

### Mergesort




**Algorithm Recursive Mergesort**

**procedure mergersort (L = $a_1$, … $a_n$)**

**if n > 1 then**

**m:= $\lfloor n/2 \rfloor$**

**L1:= $a_1$ $a_2$ ,…, $a_m$**

**L2:= $a_{m+1}$ $a_{m+2}$,…, $a_n$**

**L:= merge (mergesort(L1), mergersort(L2))**

**endif**

**T(n) =**

## Problems

[1] Prove $1^2 + 2^2 + \ldots + n^2 = \dfrac{n(n+1)(2n+1)}{6}$ for $n \geq 1$