Chapter 5 Induction and Recursion

5.1 Mathematical Induction

1. Introduction

Mathematical induction is used to prove propositions of the form $\forall nP(n)$, where the domain of discourse is the set of positive integers.

Prove $\forall\, n\, P(n)$ by mathematical induction:

(1) Basis step: Establish $P(1)$

(2) Inductive step: Prove that $P(k) \rightarrow P(k+1)$ for $k \geq 1$

Conclusion: $\forall\, n\, P(n)$, where the domain is the set of positive integers

2. Mathematical Induction

The (first) principle of Mathematical Induction: $(P(1) \wedge \forall\, k(P(k) \rightarrow P(k+1))) \rightarrow \forall\, n\, P(n)$

$$\begin{array}{l} P(1) \\ \underline{\forall k\, P(k) \rightarrow P(k+1)} \\ \therefore \forall n P(n) \end{array}$$

The validity of mathematical induction follows from the *well-ordering property* for the set of positive integers.

A set S is *well ordered* if every nonempty subset of S has a least element.

The well-ordering property: Every nonempty set of nonnegative integers has a least element.

More general form: **$\forall n \geq$ b ($P(n)$)**

The procedure :

(1) Basis step: Establish $P(b)$

(2) Inductive step: Prove that $P(k) \rightarrow P(k+1)$ for $k \geq b$

Conclusion: The basis step and the inductive step together imply $\forall n \geq$ b ($P(n)$) .

5.2 Strong Induction and Well-Ordering

1. Strong Induction

Strong Induction (*second principle of mathematical induction, complete induction*)

$(P(n_0) \wedge \forall\, k>=n_0(P(n_0) \wedge P(n_0+1) \wedge \ldots \wedge P(k) \rightarrow \mathrm{P}(k+1))) \rightarrow \forall n>=n_0\, (P(n))$

The procedure :

(1) Basis step: Establish $P(n_0)$

(2) Inductive step: Prove $P(n_0) \wedge P(n_0+1) \wedge \ldots \wedge P(k) \rightarrow \mathrm{P}(k+1)$

Conclusion: The basis step and the inductive step allow one to conclude that $\forall n \geq n_0\, P(n)$

Note:

1. The validities of both mathematical induction and strong induction follow from the well-ordering property.
2. In fact, mathematical induction, strong induction, and well-ordering are all equivalent principles.

2. Using Strong Induction in Computational Geometry

Some terms:

- polygon 多边形
- side, vertex 顶点
- a polygon is simple if no two nonconsecutive 非连续的 sides intersect 交叉.
- Every simple polygon divides the plane into two regions: its interior 内部, its exterior.
- convex 凸面, nonconvex
- diagonal 对角线, interior diagonal

- triangulation: dividing a simple polygon into triangles by adding nonintersecting diagonals.

【LEMMA 1】 Every simple polygon with at least four sides has an interior diagonal.

【Theorem 1】 The simple polygon with n sides, where n is an integer with $n \geq 3$, can be triangulated into n-2 triangles.

3. Proofs Using the Well-ordering property

5.3 Recursive Definitions and Structural Induction

1. Introduction

In a *recursive definition*, an object is defined in terms of itself. We can recursively define *sequences*, *functions* and *sets*.

2. Recursively defined functions

Recursively defined functions, with the set of nonnegative integers as its domain:

- 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.

Recursively defined functions are well-defined.

Recursively defined functions (a more general version)

- Basis Step: Specify the values of the function at the first k nonnegative integers.
- Recursive Step: Give a rule for finding its value at an integer from its values at some or all of the preceding k integers.

3. Recursively defined sets

- Basis Step: specifies an initial collection of elements.
- Recursive Step: gives rules for forming new elements of the set from those elements already known to be in the set.

Sets described in this way are well-defined.

4. Structural Induction

To prove results about recursively defined sets

- Basis Step: Show that the result holds for all elements specified in the basis step of the recursive definition to be in the set.
- 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 these new elements.

The validity of structural induction follows from the principle of mathematical induction for the nonnegative integers.

$p(n)$: the result is true for all elements of the set that are generated by n or fewer applications of the rules in the recursive step of a recursive definition.

- Basis Step: Show that $p(0)$ is true.
- Recursive Step: if we assume $p(k)$ is true, it follows that $p(k+1)$ is true.

5.4 Recursive Algorithms

An algorithm is called *recursive* if it solves a problem by reducing it to an instance of the same problem with smaller input.