Induction

See Section 1.4 of the text.

This should be review from courses in Algorithms or Discrete Mathematics

Induction on Integers, or "Mathematical Induction"

Suppose we have a statement $S(k)$ which for every value of $k >= k_0$ is either true or false. If we can show both of the following statements

a) (Base case) $S(k_0)$ is true

b) For any $k >= k_0$, if $S(k)$ is true, then $S(k+1)$ must also be true.

then $S(k)$ is true for all $k >= k_0$.

Example: Show that a binary tree with k levels has at most 2^k-1 nodes.

Proof:

*Base case***:** A tree with 0 levels has 2^0-1=0 nodes.

Inductive case: Suppose this is true for k=0,1,2,...n and we have a binary tree with n+1 levels. If we remove the root we have two binary subtrees each with n or fewer levels and the statement is true for any such tree, so each of these subtrees has no more than 2^n-1 nodes. The original tree had these two subtrees plus the root, so it had no more than 1+2*(2^n-1) = 2^{n+1}-1 nodes. This confirms our statement for trees with n+1 levels.

By induction the statement is true for all k>=0.

More typical for this class is *Structural Induction*:

We will often define sets of objects recursively. We can use induction on the structure of such a definition to show that some property holds for all objects of the set.

For example, we might define arithmetic expressions as:

Base case: numbers and variables are arithmetic expressions

Inductive case: if E and F are arithmetic expressions then so are E+F, E-F, E*F, E/F, and (E).

For example (3+5)*(4-(2+3)) is an arithmetic expression.

Now, suppose we want to show that every arithmetic expression has equal numbers of left and right parentheses. Here is a proof using structural induction:

> *Base case*: numbers and variables have 0 left parentheses and 0 right parentheses.
> *Inductive case*: Suppose arithmetic expression E has n left and right parentheses and expression F has m left and right parentheses. Then E+F,E-F,E*F and E/F each have n+m left and right parentheses and (E) has n+1 left and right parentheses. So if the statement is true for E and F it is true for anything we build from E and F.
> *By structural induction* the statement is true for all arithmetic expressions.

Here is a more formal statement of structural induction:

We can define "Structure of type $\mathcal{S}$" by

Base case: $A_0, A_1, \ldots A_j$ are all structures of type $\mathcal{S}$.

Inductive case: If $X_0, X_1, \ldots X_n$ are all structures of type $\mathcal{S}$ then so are $\mathcal{T}_0(X_0, X_1, \ldots X_n)$, $\mathcal{T}_1(X_0, X_1, \ldots X_n), \ldots \mathcal{T}_k(X_0, X_1, \ldots X_n)$.

Now let $\mathcal{P}$ be a true/false property of such structures. If we can show:

Base case: $\mathcal{P}(A_0)$ is true, $\mathcal{P}(A_1)$ is true , ... $\mathcal{P}(A_j)$ is true

Inductive case: If $X_0, X_1, \ldots X_n$ are all structures of type $\mathcal{S}$ and $\mathcal{P}(X_i)$ is true for each i, then each $\mathcal{P}(\mathcal{T}_i(X_0, X_1, \ldots X_n))$is also true

then we can conclude that $\mathcal{P}$ is true for all structures of type $\mathcal{S}$.

Mathematical induction is one of the axioms of the system of Natural Numbers. It is possible to define other systems where mathematical induction does not apply, by why be weird if you don't need to be?

If you really want you can use mathematical induction to prove that structural induction works, doing induction on the number of steps it takes to derive an element of the structure from the base cases. We are just going to accept this as a proof technique.