CSCE 222
Discrete Structures for Computing

David Kebo Hougninou

Mathematical Induction and Recursion

Chapter 5
Many mathematical processes have repeated patterns.

These processes can be characterized by sequences, and verified using mathematical induction.

Many mathematical statements assert that some property is true for all positive integers.

e.g.:
• A set with n elements has $2^n$ subsets.
• The sum of the first n positive integers is $\frac{n(n+1)}{2}$

A technique for proving such properties is mathematical induction.
Mathematical Induction

- Type of direct proof method
- General approach: prove that a property defined for integers \( n \) is true for all values of \( n \) that are greater than or equal to some initial integer

The Ladder analogy

How to climb an infinitely tall ladder?

- Suppose you know how to get onto the first rung of the ladder.
- Suppose you also know how to go from one rung to the next rung.
- Then you can climb the entire ladder (get on the first rung, then go to the second rung, the third rung, etc.)
Principle of Mathematical Induction

Is the following statement true? \( \forall \text{ integers } n \geq a, \ P(n) \)

Let \( P(n) = \text{property defined for integers } n \)

Let \( a = \text{fixed integer} \)

Suppose the following statements are true:
1. \( P(a) \) is true
2. \( \forall \text{ integers } k \geq a, \text{ if } P(k) \text{ is true then } P(k+1) \text{ is true} \)

\( P(k) \) is called the inductive hypothesis

Proof by Mathematical Induction
1. Basis step

Show that \( P(a) \) is true.

Show that the property holds for the first integer \( a \) in the sequence

If we can prove this step, then proceed to the inductive step.
Proof by Mathematical Induction

2. Inductive step

∀ integers $k \geq a$, if $P(k)$ is true then $P(k+1)$ is true.

1. Suppose that $P(k)$ is true, where $k$ is an arbitrarily chosen integer with $k \geq a$ (Inductive Hypothesis)

2. Then, show that $P(k+1)$ is true

Template for Proof by Mathematical Induction

Show that the predicate $P(n)$ is true for all $n \geq a$

Basis step: Show that $P(a)$ holds
[work showing that $P(a)$ is true]
$\therefore P(a)$ holds

Inductive step: Show that $P(k) \rightarrow P(k+1)$ holds
Assume $P(k)$ for arbitrary $k \geq a$: [State $P(k)$]
Show $P(k+1)$: [State $P(k+1)$]
[work using the inductive hypothesis]
$\therefore P(k) \rightarrow P(k+1)$ holds
$\therefore [State P(n)]$ holds for all $n \geq a$ by mathematical induction
Example 1

Prove by mathematical induction that

\[ 1 + 2 + \cdots + n = \frac{n(n + 1)}{2} \quad \text{for all integers } n \geq 1. \]

Let the property \( P(n) \) be the equation

\[ 1 + 2 + 3 + \cdots + n = \frac{n(n + 1)}{2}. \]

Left-hand side of equation. LHS  Right-hand side of equation. RHS

Basis step

Show that \( P(1) \) is true

To prove \( P(1) \), we must show that \( 1 = \frac{1(1+1)}{2} \)

LHS = 1

RHS = \( \frac{1(1+1)}{2} = \frac{2}{2} = 1 \)

For \( P(1) \), \( \text{LHS} = \text{RHS} \)

Therefore \( P(a) \) holds!
Inductive step

Show that ∀ integers \( k \geq 1 \), if \( P(k) \) is true then \( P(k + 1) \) is also true. \( P(k) \rightarrow P(k+1) \)

Suppose that \( P(k) \) is true for a particular but arbitrarily chosen integer \( k \geq 1 \).

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

We must show that \( P(k + 1) \) is true.

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

Can also be written as:

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

Show that the left-hand side and the right-hand side of \( P(k+1) \) are equal.
Inductive step

The left-hand side (LHS) of $P(k + 1)$ is:

$$1 + 2 + 3 + \cdots + (k + 1) = 1 + 2 + 3 + \cdots + k + (k + 1)$$

Recall our inductive hypothesis $P(k)$, we assumed the following was true:

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

Thus, we can rewrite our LHS equation as:

$$[1 + 2 + 3 + \cdots + k] + (k + 1) = \left[\frac{k(k + 1)}{2}\right] + (k + 1)$$

Inductive step

We can simplify our LHS equation as follows:

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

$$= \frac{k^2 + k + 2k + 1}{2} = \frac{k^2 + 3k + 2}{2}$$
Inductive step

Now, develop our RHS of P(k+1). Recall that original P(k+1) was:

\[ 1 + 2 + 3 + \cdots + (k + 1) = \frac{(k + 1)(k + 2)}{2}. \]

So, RHS is:

\[ \frac{(k + 1)(k + 2)}{2} = k^2 + 3k + 2 \]

\[ \therefore P(n) \text{ holds for all } n \geq a \text{ by mathematical induction} \]

Activity

Using Proof by Mathematical Induction, show that the following is true for all integers n \( \geq 1 \)

\[ \frac{1}{1.2} + \frac{1}{2.3} + \cdots + \frac{1}{n(n + 1)} = \frac{n}{n + 1} \]
Activity

Using Proof by Mathematical Induction, show that the following is true for all integers \( n \geq 1 \)

\[
1^2 + 2^2 + 3^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}
\]
Activity

Conjecturing and proving a formula

1. Conjecture a formula for the sum of the first $n$ positive odd integers.

2. Prove the conjecture using mathematical induction

Activity

Using Proof by Mathematical Induction, show that the following is true for all integers $n \geq 1$

$$\sum_{i=1}^{n} 2i - 1 = n^2$$
Proving Inequalities

e.g. Use mathematical induction to prove that \( n < 2^n \) ∀ integers \( n \geq 1 \).

**Basis step:** \( P(1) \) is true since \( 1 < 2^1 = 2 \).

**Inductive step:** assume \( P(k) \) holds. \( k < 2^k \), for an arbitrary positive integer \( k \).

Let’s show that \( P(k + 1) \) holds. Since by the inductive hypothesis, \( k < 2^k \), it follows that:

\[
k + 1 < 2^k + 1 \leq 2^k + 2^k = 2 \cdot 2^k = 2^{k+1}
\]

Therefore \( n < 2^n \) holds for all positive integers \( n \).

---

Proving Inequalities

e.g. Use math. induction to prove that \( 2^n < n! \) ∀ integer \( n \geq 4 \)

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

**Inductive step:** assume \( P(k) \) holds, \( 2^k < k! \) for an arbitrary integer \( k \geq 4 \). Let’s show that \( P(k + 1) \) holds:

\[
2^{k+1} = 2 \cdot 2^k
< 2.k! \text{ (by the inductive hypothesis)}
< (k + 1).k!
= (k + 1)!
\]

therefore, \( 2^n < n! \) holds, for every integer \( n \geq 4 \).

**Note!** the basis step is \( P(4) \), since \( P(0), P(1), P(2), \) and \( P(3) \) are all false.
Proving Divisibility Results

e.g. Use mathematical induction to prove that $n^3 - n$ is divisible by 3, $\forall$ positive integer $n$.

**Basis step:** $P(1)$ is true since $1^3 - 1 = 0$, which is divisible by 3.

**Inductive step:** assume $P(k)$ holds, $k^3 - k$ is divisible by 3, for an arbitrary positive integer $k$.

**Let’s show that** $P(k + 1)$ **holds:**

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

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

By the inductive hypothesis, the first term $(k^3 - k)$ is divisible by 3 and the second term is divisible by 3 since it is an integer multiplied by 3.

Therefore $(k + 1)^3 - (k + 1)$ is divisible by 3.

**Conclusion:** $n^3 - n$ is divisible by 3, for every integer positive integer $n$. 
The Ladder analogy
(Strong induction)

How to climb an infinitely tall ladder?

- Suppose you know how to get onto the first rung of the ladder.
- For every integer \( k \), if we can reach the first \( k \) rungs, then we can reach the \((k+1)^{st}\) rung.
Strong Induction

• Also called: second principle of induction, complete induction
• Strong induction makes it easier to use situations with more than one basis step (Several initial values)
• Sometimes it is easier to prove something using strong induction, as you have more hypotheses to exploit.

Strong Induction

Strong induction is a variation of induction!

So, what’s new?
In the inductive step we use all of $P(1), P(2), \ldots, P(k)$, instead of just using $P(k)$.

• **Basis step:** Verify that $P(1)$ is true
• **Inductive step:** Assume $P(1) \land P(2) \land \ldots \land P(k)$ holds for an arbitrary integer $k$, and show that $P(k + 1)$ also holds.

$[P(1) \land P(2) \land \ldots \land P(k)] \rightarrow P(k + 1)$ holds for all positive integers $k$. 
Mathematical Induction vs. Strong Induction

**Mathematical Induction**
- **Basis step:** Show that $P(1)$ is true
- **Inductive step:** Show that $P(k) \rightarrow P(k + 1)$

**Strong Induction**
- **Basis step:** Show that $P(1)$ is true
- **Inductive step:** Show that $[P(1) \land P(2) \land \ldots \land P(k)] \rightarrow P(k + 1)$

Template for Proof by Strong Induction

Predicate $P(n)$ is true for all $n \geq a$

**Basis step:** Show $P(a)$
- [work showing that $P(a)$ is true]
  - $\therefore P(a)$ holds

**Inductive step:** Show that $[P(1) \land P(2) \land \ldots \land P(k)] \rightarrow P(k + 1)$
- Assume $P(1) \land P(2) \land \ldots \land P(k)$ for arbitrary $k \geq a$:
  - Show $P(k+1)$: [state $P(k+1)$]
- [work using the inductive hypothesis]
  - $\therefore [P(1) \land P(2) \land \ldots \land P(k)] \rightarrow P(k + 1)$ holds
  - $\therefore [state P(n)]$ holds for all $n \geq a$ by strong induction
Strong Induction Example

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

Let \( P(n) \) be the proposition that \( n \) can be written as a product of primes.

**Basis step:** \( P(2) \) is true since 2 itself is prime.  (Single factor)

**Inductive step:** Assuming that \( P(2) \land P(3) \land \ldots \land P(k) \) are all true, show that \( P(k+1) \) is true.

To show that \( P(k + 1) \) must be true under this assumption, two cases need to be considered.
Strong Induction Example

Case 1: \( k + 1 \) is prime
\( k + 1 = k + 1 \), then \( P(k + 1) \) is true. (Single factor)

Case 2: \( k + 1 \) is composite
\( k + 1 = a \cdot b \), where \( a \) and \( b \) are both integers between 2 and \( k \).
By the strong inductive hypothesis, both \( a \) and \( b \) are the product of primes.
Thus \( k + 1 \), is the product of primes.

Hence, every integer greater than 1 can be written as the product of primes.

Prove that every amount of postage of 12 cents or more can be formed using just 4-cent and 5-cent stamps.
Strong Induction Example

Let \( P(n) \) be the proposition that postage of \( n \) cents can be made using 4-cent and 5-cent stamps, for all \( n \geq 12 \).

Basis step:
12 cents: use three 4-cent stamps
13 cents: use two 4-cent stamps and one 5-cent stamp
14 cents: use one 4-cent stamp and two 5-cent stamps
15 cents: use three 5-cent stamp

Strong Induction Example

Inductive step:
Inductive hypothesis: \( P(12) \land P(13) \land \ldots \land P(k) \) are all true. Assuming the inductive hypothesis, show that \( P(k + 1) \) is true.

\[ k + 1 = (k - 3) + 4, \text{ so we can use a 4-cent stamp} \]
What about the \((k - 3)\) term?

\[ k - 3 \geq 12, \text{ since } k \geq 15 \]

We know that \( P(k - 3) \) is true! i.e., we can use some combination of 4-cent and 5-cent stamps to make \( k - 3 \).

Add the extra 4-cent stamp to get the postage for \( k + 1 \).
Activity

Prove by strong induction that if $n$ is an integer greater than 1, then $n$ is divisible by a prime number.

Recursive Definitions
Recursively Defined Functions

A recursive or inductive definition of a function consists of two steps.

Basis step: Specify the value of the function at zero.

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

Example

Suppose that the recursively defined function f is:
• \( f(0) = 3 \)
• \( f(n + 1) = 2f(n) + 3 \)

What is \( f(3) \)?

Solution:
3 is the input of the function f

\[
\begin{align*}
  f(3) &= 2 \cdot f(2) + 3 \quad \text{what is } f(2)\
  f(2) &= 2 \cdot f(1) + 3 \quad \text{what is } f(1)\
  f(1) &= 2 \cdot f(0) + 3 \quad \text{what is } f(0)\
  f(0) &= 3 \quad \text{(basis)}
\end{align*}
\]

\[
\begin{align*}
  f(0) &= 3 \quad \text{(basis)} \\
  f(1) &= 2 \cdot 3 + 3 = 9 \\
  f(2) &= 2 \cdot 9 + 3 = 21 \\
  f(3) &= 2 \cdot 21 + 3 = 45
\end{align*}
\]
Example

Give a recursive definition of the factorial function \( n! \)

Solution
\[
\begin{align*}
f(0) &= 1 \\
f(n + 1) &= (n + 1) \cdot f(n)
\end{align*}
\]

String Concatenation

Two strings can be combined by concatenation.

\( \Sigma \): set of symbols
\( \Sigma^* \): set of strings formed from the symbols in \( \Sigma \)

We can recursively define the concatenation of two strings:

Basis step: if \( w \in \Sigma^* \), then \( w \cdot \lambda = w \) (\( \lambda \) is the empty string)

Recursive step:

if \( w_1 \in \Sigma^* \) and \( w_2 \in \Sigma^* \) and \( x \in \Sigma \), then \( w_1.(w_2 x) = (w_1.w_2)x \).

e.g. if \( w_1 = \text{abra} \) and \( w_2 = \text{cadabra} \), the concatenation \( w_1 \cdot w_2 = \text{abracadabra} \).
String Concatenation

To recursively concatenate $w_1$ and $w_2$:
1. If $w_2$ is empty, then return $w_1$
2. Take off the last symbol of $w_2$ and hold it aside, call it $x$
3. Concatenate $w_1$ & $w_2'$ ($w_2$ without it’s last symbol), call it $w$
4. Concatenate $w$ with $x$.

\[
\begin{align*}
\text{CAT . DOG} & = \text{CAT . (DO . G)} = (\text{CAT . DO})\text{G} = \text{CATDOG} \\
\text{CAT . DO} & = \text{CAT . (D . O)} = (\text{CAT . D})\text{O} = \text{CATDO} \\
\text{CAT . D} & = \text{CAT . (λ . D)} = (\text{CAT . λ})\text{D} = \text{CATD} \\
\text{CAT . λ} & = \text{CAT}
\end{align*}
\]

Fibonacci numbers

The Fibonacci numbers are the numbers in the following integer sequence.

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...
Recursion example

Basis step:
\[ \text{fib}(0) = 0 \]
\[ \text{fib}(1) = 1 \]

Inductive step:
\[ \text{fib}(n) = \text{fib}(n-1) + \text{fib}(n-2) \text{ for all } n \geq 2 \]

\[ \text{fib}(2) = \text{fib}(1) + \text{fib}(0) = 1 + 0 = 1 \]
\[ \text{fib}(3) = \text{fib}(2) + \text{fib}(1) = 1 + 1 = 2 \]
\[ \text{fib}(4) = \text{fib}(3) + \text{fib}(2) = 2 + 1 = 3 \]
\[ \text{fib}(5) = \text{fib}(4) + \text{fib}(3) = 3 + 2 = 5 \]

Recursively Defined Set

Here’s how to define a set recursively:

1. **Basis step**: Explicitly list a few elements that are in the set.
2. **Inductive step**: Give rules for forming new elements of the set using those already known to be in the set.
3. **Exclusion Rule**: A recursively defined set contains nothing other than the elements in the basis step or generated by the inductive step.
Recursively Defined Set

Let’s define a set $S$ of integers as:

**Basis step:** $3 \in S$

**Inductive step:** If $x \in S$ and $y \in S$, then $(x + y) \in S$.

We can build up the elements of $S$ as being $3, 3 + 3 = 6, 3 + 6 = 9, 6 + 6 = 12$, etc.

Intuitively, $S$ is all positive multiples of $3$.

We will shortly see how to prove this rigorously.

Recursively Defined Set: Strings

$\Sigma$: set of **symbols**

$\Sigma^*$: set of **strings** formed from the symbols in $\Sigma$

**Basis step:** the empty string, denoted $\lambda$, which consists of no elements of $\Sigma$, is in $\Sigma^*$

**Recursive step:** If $w$ is in $\Sigma^*$ and $x$ is in $\Sigma$, then $w$ followed by $x$, $(wx)$, is in $\Sigma^*$. (Concatenation)

e.g.

Suppose $\Sigma = \{0, 1\}$. Then $\Sigma^*$ consists of $\lambda$, $\lambda 0 = 0$, $\lambda 1 = 1$, $00$, $01$, ... etc. (All finite binary strings.)
Recursively Defined Set: WFFs

The set of well-formed formulas (WFFs) of propositional logic:

T:       true
F:       false
s:       propositional variable
{¬, ∧, ∨, →, ↔}: operators

**Basis step:** T, F and s are well-formed formulas

**Inductive step:** Suppose E and F are in WFF. Then:

¬E          is a WFF
(E ∧ F)     is a WFF
(E ∨ F)     is a WFF
(E → F)     is a WFF
(E ↔ F)     is a WFF

Recursive Definition of Full Binary Trees

A graph is a structure consisting of:
• Vertices
• Edges connecting vertices

A tree is a graph that is “connected” and has no “cycles”.

**A full binary tree** is a tree that contains a distinguished vertex called the root and every vertex has either 0 or 2 children.
Recursive Definition of Full Binary Trees

**Basis step:** A single vertex \( r \) is a full binary tree, with \( r \) as the root.

**Inductive step:** Suppose \( T_1 \) and \( T_2 \) are full binary trees with no vertices in common. We build a new full binary tree like this:

1. Let \( r \) be a **new vertex** (not in \( T_1 \) or \( T_2 \)); \( r \) will be the root of the new tree.
2. Add an edge from \( r \) to the root of \( T_1 \)
3. Add an edge from \( r \) to the root of \( T_2 \)
Recursive Algorithms

Recursion is a powerful tool for designing algorithms and programs.

A recursive algorithm solves a problem by reducing it to an instance of the same problem with smaller input.

A recursive algorithm must satisfy the following:
1. There is one (or more) well-defined stopping cases
2. Each subsequent instance of the problem gets closer to a stopping case.

Recursive Algorithm: Fibonacci

1: function fib(n):
2:  if n = 0 then // stopping case
3:      return 0
4:  else if n = 1 then // stopping case
5:      return 1
6:  else
7:      return fib(n-1) + fib(n-2) // closer to stopping case
8:  end if
9: end function
Activity: call Tree for Fibonacci

Draw the call tree for $fib(4)$ ...

Recursive Algorithm: $n!$

1: function factorial(n):
2:     if n = 0 then  // stopping case
3:         return 1
4:     else
5:         return n * factorial(n-1) // closer to stopping case
6:     end if
7: end function
Recursive Algorithm: $a^n$

1: function power(a, n):
2: if n = 0 then // stopping case
3: return 1
4: else
5: return a * power(a, n-1) // closer to stopping case
6: end if
7: end function

Recursive Algorithm: linear search for x

1: function linear_search (i, j, x)
2: if A[i] = x then // stopping case
3: return i
4: else if i = j then // another stopping case
5: return 0
6: else
7: return linear_search (i+1, j, x) // closer to stopping case
9: end if
10: end function
Recursive Algorithm: binary search for x

1: function binary_search (i, j, x):
2:     \[ m \leftarrow \left\lfloor \frac{i+j}{2} \right\rfloor \]
3:     if \( x = A[m] \) then \hspace{1cm} // stopping case
4:         return \( m \)
5:     else if \((x < A[m]) \) and \((i < m)\) then
6:         return \( \text{binary_search} (i, m-1, x) \) // search in left half; closer to stopping case
7:     else if \((x > A[m]) \) and \((j > m)\) then
8:         return \( \text{binary_search} (m+1, j, x) \) // search in right half; closer to stopping case
9:     else return 0 \hspace{1cm} // another stopping case
10: end function

Structural Induction

Structural induction is doing mathematical induction on the number of times that the recursive step is employed to create a new element!

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

Recursive step: Show that if the statement holds for each elements used to construct new elements in the recursive step of the recursive definition, the result holds for the new elements.
Structural Induction

Structural Induction uses induction (regular or strong), about the elements of a recursively defined set.

**Basis step:** Prove that the property is true for the elements of the set defined in the basis step of the definition of the set.

**Inductive step:**
Assume the property is true for the elements used in the recursive step of the definition to construct the new element(s).

Prove that the property is true of the newly constructed element(s).

Structural Induction example

Recall this recursive definition for set S:

**Basis step:** $3 \in S$

**Inductive step:** If $x \in S$ and $y \in S$, then $(x + y) \in S$.

Prove by structural induction that S consists of all positive integers divisible by 3.
Structural Induction example

Let A be the set of all positive integers divisible by 3.
We will show that $S = A$.

If a set $A$ equals another set $B$; then $A$ is a subset of $B$, and $B$ is a subset of $A$.

*Remember sets equality in Chapter 2

Structural Induction example

Part 1: Prove that $A$ is a subset of $S$

We can express $A$ as $\{3n \mid n > 0\}$.
Let's use regular induction to show $A \subseteq S$

Let $P(n)$ be the statement “$3n \in S$”
Show $P(n)$ is true for all integers $n \geq 1$.

Basis step: $P(1)$ is “$3 \cdot 1 \in S$”. Since $3 \cdot 1 = 3$, this follows the basis step of the definition of $S$. 
Structural Induction example

Part 1 (Cont’d …)

Inductive step:
Show that \( P(k) \rightarrow P(k+1) \)
Let’s assume \( 3 \cdot k \in S \)
\( 3 \cdot (k + 1) = 3 \cdot k + 3 \)
  • \( 3 \cdot k \in S \), by the inductive hypothesis.
  • \( 3 \in S \), by the basis step of the definition of \( S \).
\( 3 \cdot k + 3 \in S \), by the inductive step of the definition of \( S \).
\[ \therefore 3 \cdot (k + 1) \in S \]

Part 2: Prove that \( S \) is a subset of \( A \)
This is where we use structural induction!

Basis step: The basis element of \( S \) is 3. Since \( 3 = 3 \cdot 1 \), \( 3 \in A \)

Inductive step: The inductive step of the definition of \( S \) states that if \( x \in S \) and \( y \in S \), then \( (x + y) \in S \).
Show that \( (x \in A \text{ and } y \in A) \rightarrow (x + y) \in A \).
Let \( x = 3 \cdot a \) and \( y = 3 \cdot b \) for some integers \( a \) and \( b \).
Then \( x + y = 3 \cdot (a + b) \), so \( x + y \in A \).
End