CPSC 121: Models of Computation

Unit 9b: Mathematical Induction - part 2

Based on slides by Patrice Belleville and Steve Wolfman

Outline

- **Strong Mathematical Induction.**
- Pattern and Examples
- More examples using induction.
- Further exercises.

Strong Mathematical Induction

- The induction we have seen so far handles problems which can be broken down to sub-problems of size 1 less that the original problem size.
- How do we handle more general problems which can be defined in terms of one or more smaller similar problems with various but smaller sizes?
- We need to make our induction technique more general.

Strong Mathematical Induction

- When we want to prove

  $\forall n \in Z^+, Q(n)$

  We use a slightly different induction step.
  - Instead of proving that
    - $\forall n \in Z^+, Q(n-1) \rightarrow Q(n)$
  - We prove that
    - $\forall n \in Z^+, (Q(1) \wedge Q(2) \wedge \ldots \wedge Q(n-1)) \rightarrow Q(n)$
- That is, we now assume that the theorem is true for all the numbers smaller than n and prove it for n
- We can also show that this type of induction is a valid proof technique.

Outline

- Strong Mathematical Induction.
- **Pattern and Examples**
- More examples using induction.
- Further exercises.

Breaking down into all smaller problems

You want to prove P(n) for all n ≥ 22. You know that P(n) is true if P(·) is true for every integer from 24 up to n-1. How do we fill in the blanks?

Theorem: P(n) is true for all n ≥ _______.

Breaking down into all smaller problems

You want to prove P(n) for all n ≥ 22. You know that P(n) is true if P(·) is true for every integer from 24 up to n-1. How do we fill in the blanks?

Theorem: P(n) is true for all n ≥ **22**.

Proof: We proceed by induction on n.

Base Case(s) (P(·) is true for _______**):**
Prove each base case via your other techniques.

Examples: Breaking down into all smaller problems

You want to prove P(n) for all n ≥ 22. You know that P(n) is true if P(·) is true for every integer from 24 up to n-1. How do we fill in the blanks?

Theorem: P(n) is true for all n ≥ **22**.

Proof: We proceed by induction on n.

Base Cases: Prove P(·) is true for **22 , 23 and 24 (and possibly more base cases that are not reachable from 22 using the inductive step**)
Prove each base case via your other techniques. For n=23, we just need n=22 and so on.

Inductive Step: For n > _______, if P(·) is true for ____________, then P(n) is true.

Examples: Breaking down into all smaller problems

You want to prove P(n) for all n ≥ 22. You know that P(n) is true if P(·) is true for every integer from 22 up to n-1. How do we fill in the blanks?

Theorem: P(n) is true for all n ≥ **22**.

Proof: We proceed by induction on n.

Base Case(s): Prove P(·) is true for **22 , 23 and 24 (and possibly more base cases that are not reachable from 22 using the inductive step)**
Prove each base case via your other techniques.

Inductive Step: For n > **24**: if P(·) is true for **every integer from 24 up to n-1**, then P(n) is true:

WLOG, let n be greater than ____________.

Assume P(·) is true for __________________.

Examples: Breaking down into all smaller problems

You want to prove P(n) for all n ≥ 22. You know that P(n) is true if P(·) is true for every integer from 22 up to n-1. How do we fill in the blanks?

Theorem: P(n) is true for all n ≥ **22**.

Proof: We proceed by induction on n.

Base Case(s): Prove P(·) is true for **22 , 23 and 24 (and possibly more base cases that are not reachable from 22 using the inductive step)**
Prove each base case via your other techniques.

Inductive Step: For n > **24**: if P(·) is true for **every integer from 24 up to n-1**, then P(n) is true:

WLOG, let n be greater than **24**.

Assume **for all integers i where 24 < i < n, P(i) is true**. We'll prove P(n)

Break P(n) down in terms of the smaller case(s).

The smaller cases are true, by assumption.

Build back up to show that P(n) *is true.*

This completes our induction proof. QED

Examples: breaking down into a problem half as big

You want to prove P(n) for all n ≥ 7. You know that P(n) is true if P(⌊n/2⌋) and P(⌈n/2⌉) are both true (i.e., P(·) is true for n/2 rounded down and n/2 rounded up). How do we fill in the blanks?

But, your insight may come in *any* form.
Maybe you need problems half as large or one-third.
Maybe you need problems that are 7 smaller.
Maybe you need the problems that are 1, 2, and 3 smaller.

Regardless, the pattern is the same!

Examples: breaking down into a problem half as big

You want to prove P(n) for all n ≥ 7. You know that P(n) is true if P(⌊n/2⌋) and P(⌈n/2⌉) are both true (i.e., P(·) is true for n/2 rounded down and n/2 rounded up). How do we fill in the blanks?

Theorem: P(n) is true for all n ≥ _______.

Examples: breaking down into a problem half as big

You want to prove P(n) for all n ≥ 7. You know that P(n) is true if P(⌊n/2⌋) and P(⌈n/2⌉) are both true (i.e., P(·) is true for n/2 rounded down and n/2 rounded up). How do we fill in the blanks?

Theorem: P(n) is true for all n ≥ **7**.

Proof: We proceed by induction on n.

Base Case(s) (P(·) is true for _______**):**
Prove each base case via your other techniques.

Examples: breaking down into a problem half as big

You want to prove P(n) for all n ≥ 7. You know that P(n) is true if P(⌊n/2⌋) and P(⌈n/2⌉) are both true (i.e., P(·) is true for n/2 rounded down and n/2 rounded up). How do we fill in the blanks?

Theorem: P(n) is true for all n ≥ **7**.

Proof: We proceed by induction on n.

Base Case(s) (P(·) is true for n = **7, 8, 9, 10, 11, 12, 13):**
Prove each base case via your other techniques. (We need all the way up to 13 because only at 14/2 do we reach a base case. From 15 on, we always eventually hit a base case.)

Inductive Step (for n > _______, if P(·) is true for ____________, then P(n) is true):

Examples: breaking down into a problem half as big

You want to prove P(n) for all n ≥ 7. You know that P(n) is true if P(⌊n/2⌋) and P(⌈n/2⌉) are both true (i.e., P(·) is true for n/2 rounded down and n/2 rounded up). How do we fill in the blanks?

Theorem: P(n) is true for all n ≥ **7**.

Proof: We proceed by induction on n.

Base Case(s) (P(·) is true for n = **7, 8, 9, 10, 11, 12, 13):**
Prove each base case via your other techniques.

Inductive Step (for n > **13**: if P(·) is true for **⌊n/2⌋ and ⌈n/2⌉**, then P(n) is true):

WLOG, let n be greater than ____________.

Assume P(·) is true for __________________.

Examples: breaking down into a problem half as big

You want to prove P(n) for all n ≥ 7. You know that P(n) is true if P(⌊n/2⌋) and P(⌈n/2⌉) are both true (i.e., P(·) is true for n/2 rounded down and n/2 rounded up). How do we fill in the blanks?

Theorem: P(n) is true for all n ≥ **7**.

Proof: We proceed by induction on n.

Base Case(s) (P(·) is true for n = **7, 8, 9, 10, 11, 12, 13):**
Prove each base case via your other techniques.

Inductive Step (for n > **13**: if P(·) is true for **⌊n/2⌋ and ⌈n/2⌉**, then P(n) is true):

WLOG, let n be greater than **13**.

Assume P(·) is true for **⌊n/2⌋ and ⌈n/2⌉**.

Break P(n) down in terms of the smaller case(s).

The smaller cases are true, by assumption.

Build back up to show that P(n) *is true.*

This completes our induction proof. QED

Example 1

- Every positive integer n greater than 1 can be written as a product of primes.
- What base case(s) should we use?
  - A. n = 1
  - B. n = 2
  - C. n = 2, 3 or 5.
  - D. n is prime.
  - E. None of the above.

Example 1

- Every positive integer n greater than 1 can be written as a product of primes.
- What is the inductive step?
  - A. For every integer k >2, if k-1 is a product of primes, then k is a product of primes
  - B. For every integer k ≥ 2, if k-1 is a product of primes, then k is a product of primes
  - C. For every integer n >2, if every integer k, 2 ≤ k ≤ n-1, is a product of primes, then n is a product of primes.
  - D. For every integer n ≥ 2, if every integer k, 2 < k ≤ n-1, is a product of primes, then n is a product of primes .
  - E. None of the above.

Example 1

- Proof: we prove the result by induction on n.
  - **Base case:** n = 2
    - Since 2 is prime, the statement is true.
  - **Induction step:**
    - Let n be any integer greater than 2. Suppose that every number from 2 to n-1 is a product of primes. We'll show that n is a product of primes
    - Case 1: ______________________

Example 1

- Proof: we prove the result by induction on n.
  - **Base case:** n = 2 is prime.
    - Since 2 is prime, the statement is true.
  - **Induction step:**
    - Let n be any integer greater than 2. Suppose that every number from 2 to n-1 is a product of primes. We'll show that n is a product of primes
    - Case 1: n is prime. Then the statement is true.

Example 1

- Proof: we prove the result by induction on n.
  - **Base case:** n = 2 is prime.
    - Since 2 is prime, the statement is true.
  - **Induction step:**
    - Let n be any integer greater than 2. Suppose that every number from 2 to n-1 is a product of primes. We'll show that n is a product of primes
    - Case 1: n is prime. Then the statement is true
    - Case 2: n is composite. Then

      n =

Example 1

- Proof: we prove the result by induction on n.
  - **Base case:** n = 2 is prime.
    - Since 2 is prime, the statement is true.
  - **Induction step:**
    - Let n be any integer greater than 2. Suppose that every number from 2 to n-1 is a product of primes. We'll show that n is a product of primes
    - Case 1: n is prime. Then the statement is true
    - Case 2: n is composite. Then

      n = a*b such that 1< a < n and 1< b < n
    -

Example 1

- Proof: we prove the result by induction on n.
  - **Base case:** n = 2 is prime.
    - Since 2 is prime, the statement is true.
  - **Induction step:**
    - Let n be any integer greater than 2. Suppose that every number from 2 to n-1 is a product of primes. We'll show that n is a product of primes
    - Case 1: n is prime. Then the statement is true
    - Case 2: n is composite. Then

      n = a*b such that 1< a < n and 1< b < n
    - By the induction hypothesis:

      a = p1*p2*…*pm where pi is prime

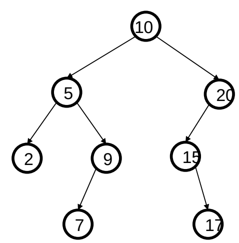
      b = q1*q2*…*qr where qi is prime
    - and

      n = p1*p2*…*pm*q1*q2*…*qr

  QED

Binary Trees

- CPSC 110 review: A binary tree is a data structure that is defined recursively as following
- A binary tree is either
  - Empty, or
  - A node with some data, and two children that are themselves binary trees.



Proving Correctness : Binary trees

- Example 2: Consider the following function:

```
(define (tree-size t)
  (if (null? t)
      0
     (+ 1
        (tree-size (left-child t))
        (tree-size (right-child t)))))
```

- How can we prove that it correctly computes the number of (non-null) nodes of the tree?

Example : Binary trees

- We prove this using mathematical induction on the size of the tree t.
  - **Base case**: t is null
    - In this case t contains exactly 0 nodes.
    - The algorithm returns 0. Therefore it is correct
  - **Induction step**:
    - Assume the algorithm works for trees that are smaller than t.
    - Because the left sub-tree of t is smaller than t, the 1st recursive calls returns the size of the left sub-tree of t.

Example : Binary trees

- Induction step (continued)
  - Similarly the right sub-tree of t is smaller than t, and so the 2nd recursive call returns the size of the right sub-tree of t.
  - The algorithm then returns 1 + the sum of the values returned by the recursive calls.
  - This is exactly the size of t (1 for the root + the sum of the sizes of the two sub-trees).
- Hence our algorithm computes correctly the size of every tree. QED

Worked Example : What is Wrong ?

Theorem: All integers greater than or equal to 2 are even.

Proof: We proceed by induction on n.

Base Case : Theorem is true for the first case where n = 2. Since 2 = 2*1, 2 is even.

Inductive Step For any k >2, assume that k-2 is even and we'll show that k is even.

- WLOG, let k be any integer > 2 .
- By the inductive hypothesis, k-2 is even.
- Therefore k -2 = 2 m for some integer m
- Then k = 2m -2 = 2(m-1).
- Since m-1 is an integer, k is even

QED

Recall: Practical Induction

How can we figure out an inductive proof?

- Start at the inductive step!
- Look at a "big" problem (of size n).
- Figure out how to break it down into smaller pieces.
- Assume those smaller pieces work. That will end up as your Induction Hypothesis.
- Figure out which problems cannot be broken down (usually small ones!). Those will end up as your basis step(s).

Outline

- Strong Mathematical Induction.
- Pattern and Examples
- **More examples using induction**.
- Further exercises.

Example : Geometric series

- Example 2: geometric series
  - We will prove that for every value of $a \neq 0, 1$:

$$\sum_{i=0}^{t} a^i = \frac{a^{t+1} - 1}{a - 1}$$

  - These summations occur frequently when we need to determine the running time of divide-and-conquer algorithms (in CPSC 320).

Example : Geometric series

- Proof:
  - Base case: t = 0
    - o In this case the summation is $a^0 = 1$, and $\frac{a^1 - 1}{a - 1} = 1$
  - Induction step:
    - o Pick any t > 0. Assume that the statement is true for t-1
    - o Now will show that the statement is true for t
    - o $\sum_{i=1}^{t} a^i = (\sum_{i=0}^{t-1} ai\ ) + a^t\ =$

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

  - QED

Example: Using Inequalities

- Example 3: Prove that
  $\forall n \geq 4, 2^n < n!$
- Rules for proving inequalities:
  - Start from one side (say the left side)
  - Work step by step towards the other.
  - When dealing with <, you are allowed to make the expression larger, but never smaller.
    - o Example: if I am smaller than you, then I am still smaller than you when you stand on a bench.

Inequalities

- Proof: by induction on n.
  - Base case: $n = 4$
    - o $2^4 < 4!$ because $16 < 24$
  - Induction step: we want to prove that
    for any $k > 4$, if $2^{k-1} < (k-1)!$ then $2^k < k!$
    - o Induction hypothesis: assume that $2^{k-1} < (k-1)!$
    - o Then
      $2^k = 2(2^{k-1}) < 2(k-1)! < k(k-1)! = k!$
      this is where we "approximate"
      the induction hypothesis is used here
  - Hence by the principle of M.I., $\forall n \geq 4, 2^n < n!$

Example : Binary Search

- Example: binary search
  - Suppose we have something like a list, but whose i-th element and length can be found in a single step.
    - o This structure is called a vector in Racket.
    - o It is similar to an ArrayList in Java.
  - We assume that we have such a vector, sorted in increasing order.
    - o Examples: ("Ann", "Charles", "Dora", "Gregor", "Wei").
  - We want to find the position of a given element (for instance, "Dora").

Binary Search

- Claim: the following algorithm (formerly known as Binary Search) works:

```
(define (binary-search avector first-pos last-pos x)
  (if (> first-pos last-pos)
    #f
    (if (= first-pos last-pos)
      (if (= x (vector-ref avector first-pos)) first-pos #f)
      (let ((mid-pos (quotient (+ first-pos last-pos) 2)))
        (if (= x (vector-ref avector mid-pos))
          mid-pos
          (if (< x (vector-ref avector mid-pos))
            (binary-search avector first-pos (- mid-pos 1) x)
            (binary-search avector (+ mid-pos 1) last-pos x)…)
```

Binary Search

- Proof: by induction on the size of the part of the vector that we are searching.
  - Base cases: size ≤ 1
    - If size is 0, then x can not be in that part of the vector, so returning #f is correct.
    - If size is 1, then there is only one possible location for x, and we check this position.
  - Induction step: size ≥ 2
    - Suppose that the algorithm will find x (if it is in the vector) for every vector with <u>fewer than</u> size elements.
    - We'll show that the algorithm will find x in any vector with size elements.

Binary Search

- Proof (continued)
    - If x is at position mid-pos, then the algorithm returns mid-pos.
    - Otherwise, x is either smaller than the element at position mid-pos, or larger.
    - If x is smaller then either x is in the first half of the vector or not in at all
      - Algorithm returns the result of searching the first half of the vector which by the IH is the correct result
    - If x is larger then either x is in the second half of the vector or not in at all
      - Algorithm returns the result of searching the second half of the vector which by the IH is the correct result
  - Hence by the principle of M.I., the algorithm returns the correct value. QED

Outline

- Strong Mathematical Induction.
- Pattern and Examples
- More examples using induction.
- **Further exercises**.

Additional Examples

- Prove that for every n ≥ 1, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - \frac{1}{n}$
- (very challenging): Prove that binary search makes at most [log2 (size+1)] comparisons if size ≥ 1.

  Give a proof by induction on the size of the vector.