Induction

Lecture 7
CS 2110 – Fall 2011

Announcements

• A2 due 11:59pm
• Please let us know about exam conflicts if you haven’t already
• In-class quiz this Thursday

Overview

• Recursion
  • A programming strategy that solves a problem by reducing it to simpler or smaller instance(s) of the same problem
• Induction
  • A mathematical strategy for proving statements about natural numbers 0, 1, 2, ... (or more generally, about inductively defined objects)
  • They are very closely related
  • Induction can be used to establish the correctness and complexity of programs

Defining Functions

• It is often useful to describe a function in different ways
  • Let S : int → int be the function where S(n) is the sum of the integers from 0 to n. For example,
    • S(0) = 0
    • S(3) = 0 + 1 + 2 + 3 = 6
  • Definition: iterative form
    • S(n) = 0 + 1 + ... + n
  • Another characterization: closed form
    • S(n) = n(n+1)/2 = \frac{1}{2}n^2 + \frac{1}{2}n

Sum of Squares

• A more complex example
  • Let SQ : int → int be the function that gives the sum of the squares of integers from 0 to n:
    • SQ(0) = 0
    • SQ(3) = 0^2 + 1^2 + 2^2 + 3^2 = 14
  • Definition (iterative form): SQ(n) = 0^2 + 1^2 + ... + n^2
  • Is there an equivalent closed-form expression?

Closed-Form Expression for SQ(n)

• Sum of integers between 0 through n was n(n+1)/2 which is a quadratic in n (that is, O(n^2))
• Inspired guess: perhaps sum of squares of integers between 0 through n is a cubic in n
• Conjecture: SQ(n) = an^3+bn^2+cn+d
  where a, b, c, d are unknown coefficients
• How can we find the values of the four unknowns?
  • Idea: Use any 4 values of n to generate 4 linear equations, and then solve
Finding Coefficients

\[ SQ(n) = 0^2 + 1^2 + \ldots + n^2 = an^3 + bn^2 + cn + d \]

- Use \( n = 0, 1, 2, 3 \)
  - \( SQ(0) = 0 = a \cdot 0 + b \cdot 0 + c \cdot 0 + d \)
  - \( SQ(1) = 1 = a \cdot 1 + b \cdot 1 + c \cdot 1 + d \)
  - \( SQ(2) = 5 = a \cdot 8 + b \cdot 4 + c \cdot 2 + d \)
  - \( SQ(3) = 14 = a \cdot 27 + b \cdot 9 + c \cdot 3 + d \)
- Solve these 4 equations to get
  - \( a = \frac{1}{3} \)
  - \( b = \frac{1}{2} \)
  - \( c = \frac{1}{6} \)
  - \( d = 0 \)

Is the Formula Correct?

- This suggests the closed-form solution
  \[ SQ(n) = 0^2 + 1^2 + \ldots + n^2 = \frac{n^3}{3} + \frac{n^2}{2} + \frac{n}{6} = \frac{n(n+1)(2n+1)}{6} \]
- Question: Is this true for all \( n \)?
  - We used \( n = 0, 1, 2, 3 \) to determine these coefficients, so we know it is true for them.
  - But we do not know that it is true for other values of \( n \).

One Approach

- Try a few other values of \( n \) to see if they work
  - Try \( n = 5 \): \( SQ(n) = 0 + 1 + 4 + 9 + 16 + 25 = 55 \)
  - Closed-form expression: 5 \( \cdot \) 6 \( \cdot \) 11 \( \div \) 6 = 55
  - Works!
- Try some more values …
- We can never prove validity of the closed-form solution for all values of \( n \) this way, since there are an infinite number of values of \( n \).

An Inductive Definition

- To solve this problem, let’s express \( SQ(n) \) in a different way:
  - \( SQ(n) = \frac{2n+1}{2} + \ldots + n^2 \)
  - The part in the box is just \( SQ(n-1) \)
- This leads to the following inductive definition
  - \( SQ(0) = 0 \)
  - \( SQ(n) = SQ(n-1) + n^2, n > 0 \)
  - Base case
  - Inductive case
- Thus
  \[ SQ(4) = SQ(3) + 4^2 = SQ(2) + 3^2 + 4^2 = SQ(1) + 2^2 + 3^2 + 4^2 = SQ(0) + 1^2 + 2^2 + 3^2 + 4^2 = 0 + 1 + 2 + 3 + 4 = 10 \]

Are These Two Functions Equal?

- \( SQ_i (i = \text{inductive}) \)
  \[ SQ(0) = 0 \]
  \[ SQ(n) = SQ(n-1) + n^2, n > 0 \]
- \( SQ_c (c = \text{closed-form}) \)
  \[ SQ(n) = \frac{n(n+1)(2n+1)}{6} \]

Induction over Integers

- To prove that some property \( P(n) \) holds for all integers \( n \geq 0 \).
  1. Basis: Show that \( P(0) \) is true
  2. Induction Step: Assuming that \( P(k) \) is true for an unspecified integer \( k \), show that \( P(k+1) \) is true
- Conclusion: Because we could have picked any \( k \), we conclude that \( P(n) \) holds for all integers \( n \geq 0 \).
Assume equally spaced dominos, and assume that spacing between dominos is less than domino length
How would you argue that all dominos would fall?

Dumb argument:
- Domino 0 falls because we push it over
- Domino 0 hits domino 1, therefore domino 1 falls
- Domino 1 hits domino 2, therefore domino 2 falls
- Domino 2 hits domino 3, therefore domino 3 falls
- ...

Is there a more compact argument we can make?

Better Argument

Argument:
- Domino 0 falls because we push it over (Base case or Basis)
- Assume that domino k falls over (Induction hypothesis)
- Because domino k's length is larger than inter-domino spacing, it will knock over domino k+1 (Inductive step)
- Because k was arbitrary, the argument holds for any k. We conclude that all dominos will fall over (Conclusion)

This is an inductive argument
This version is called weak induction
- There is also strong induction (later)
- Not only is this argument more compact, it works for an arbitrary number of dominos!

SQ_i(n) = SQ_c(n) for all n?

Define P(n) as SQ_i(n) = SQ_c(n)

Proof (by Induction)
- Recall:
  - SQ(0) = 0
  - SQ(n) = n(n+1)(2n+1)/6
- Let P(n) be the proposition that SQ_i(n) = SQ_c(n)
- Basis: P(0) holds because SQ(0) = 0 = SQ(0) by definition
- Induction hypothesis: Assume SQ_i(k) = SQ_c(k)
- Inductive step: SQ_i(k+1) = SQ_i(k) + (k+1)^2 by definition of SQ_i(k+1) = SQ_c(k) + (k+1)^2 by the induction hypothesis = k(k+1)(2k+1)/6 + (k+1)^2 by definition of SQ_c(k) = (k+1)(k+2)(2k+3)/6 by algebra = SQ_c(k+1) by definition of SQ_c(k+1)
- Conclusion: SQ_i(n) = SQ_c(n) for all n ≥ 0

Another Example

Prove that 0 + 1 + ... + n = n(n+1)/2

Basis: Obviously holds for n = 0
- Induction Hypothesis: Assume 0+1+...+k = k(k+1)/2
- Inductive Step: 0+1+...+(k+1) = [0+1+...+k] + (k+1) by def = k(k+1)/2 + (k+1) by I.H. = (k+1)(k+2)/2 by algebra
- Conclusion: 0+1+...+n = n(n+1)/2 for all n ≥ 0

A Note on Base Cases

- Sometimes we are interested in showing some proposition is true for integers ≥ b
- Intuition: we knock over domino b, and dominos in front get knocked over; not interested in 0,1,...,(b-1)
- In general, the base case in induction does not have to be 0
- If base case is some integer b
  - Induction proves the proposition for n = b, b+1, b+2, ...
  - Does not say anything about n = 0,1,...,b-1
Weak Induction: Nonzero Base Case

- Claim: You can make any amount of postage above 8¢ with some combination of 3¢ and 5¢ stamps
- Basis: True for 8¢: \( 8 = 3 + 5 \)
- Induction hypothesis: Suppose true for some \( k \geq 8 \)
- Inductive step:
  - If used a 5¢ stamp to make \( k \), replace it by two 3¢ stamps. Get \( k+1 \).
  - If did not use a 5¢ stamp to make \( k \), must have used at least three 3¢ stamps. Replace three 3¢ stamps by two 5¢ stamps. Get \( k+1 \).
- Conclusion: Any amount of postage above 8¢ can be made with some combination of 3¢ and 5¢ stamps

What are the “Dominos”?

- In some problems, it can be tricky to determine how to set up the induction
- This is particularly true for geometric problems

A Tiling Problem

- A chessboard has one square cut out of it
- Can the remaining board be tiled using tiles of the shape shown in the picture (rotation allowed)?
- Not obvious that we can use induction!

Proof Outline

- Consider boards of size \( 2^n \times 2^n \) for \( n = 1, 2, \ldots \)
- Basis: Show that tiling is possible for 2 x 2 board
- Induction Hypothesis: Assume the \( 2^n \times 2^n \) board can be tiled
- Inductive Step: Using I.H. show that the \( 2^{n+1} \times 2^{n+1} \) board can be tiled
- Conclusion: Any \( 2^n \times 2^n \) board can be tiled, \( n = 1, 2, \ldots \)
  - Our chessboard (8 x 8) is a special case of this argument
  - We will have proven the 8 x 8 special case by solving a more general problem!

Basis

- The 2 x 2 board can be tiled regardless of which one of the four pieces has been omitted

4 x 4 Case

- Divide the 4 x 4 board into four 2 x 2 sub-boards
- One of the four sub-boards has the missing piece
  - By the I.H., that sub-board can be tiled since it is a 2 x 2 board with a missing piece
  - Tile the center squares of the three remaining sub-boards as shown
    - This leaves three 2 x 2 boards, each with a missing piece
    - We know these can be tiled by the induction hypothesis
Divide board into four sub-boards and tile the center squares of the three complete sub-boards.

The remaining portions of the sub-boards can be tiled by the I.H. (which assumes we can tile $2^k \times 2^k$ boards).

**When Induction Fails**

- Sometimes an inductive proof strategy for some proposition may fail.
- This does not necessarily mean that the proposition is wrong.
  - It may just mean that the particular inductive strategy you are using is the wrong choice.
- A different induction hypothesis (or a different proof strategy altogether) may succeed.

**Tiling Example (Poor Strategy)**

- Let’s try a different induction strategy.
- Proposition:
  - Any $n \times n$ board with one missing square can be tiled.
- Problem:
  - A $3 \times 3$ board with one missing square has 8 remaining squares, but our tile has 3 squares; tiling is impossible.
  - Thus, any attempt to give an inductive proof of this proposition must fail.
- Note that this failed proof does not tell us anything about the $8 \times 8$ case.

**A Similar Tiling Problem**

- A chessboard has opposite corners cut out of it. Can the remaining board be tiled using tiles of the shape shown in the picture (rotation allowed)?
- Induction fails here. Why? (Well…for one thing, this board can’t be tiled with dominos.)

**Strong Induction**

- We want to prove that some property $P$ holds for all $n$.
- Weak induction:
  - $P(0)$: Show that property $P$ is true for 0.
  - $P(k) \implies P(k+1)$: Show that if property $P$ is true for $k$, then it is true for $k+1$.
  - Conclude that $P(n)$ holds for all $n$.
- Strong induction:
  - $P(0)$: Show that property $P$ is true for 0.
  - $P(1)$ and $P(2)$ and … and $P(k) \implies P(k+1)$: Show that if $P$ is true for all numbers $\leq k$, then it is true for $k+1$.
  - Conclude that $P(n)$ holds for all $n$.
- Both proof techniques are equally powerful.

**Conclusion**

- Induction is a powerful proof technique.
- Recursion is a powerful programming technique.
- Induction and recursion are closely related.
  - We can use induction to prove correctness and complexity results about recursive programs.