INFS 519 – Fall 2015

Program Design and Data Structures

Lecture 6

Instructor: James Pope

Email: jpope8@gmu.edu

Today

- Programming Assignment Questions
- Midterm

Next Week

- No class

Next Next Week

- Midterm Review
- Class feedback
- Binary Search Trees, Esp. AVLs & B-Trees

Midterm!

Test Suite Example 1

```java
import static org.junit.Assert.*;

/**
 * Suite to test OrderedArrayMinPQ
 */
public class TestOrderedArrayMinPQ
{
    @org.junit.Test
    public void testInsertDelMin()
    {
        OrderedArrayMinPQ minpq = new OrderedArrayMinPQ();

        minpq.insert(90);
        minpq.insert(15);
        minpq.insert(74);

        assertEquals(  3, minpq.size()    );
        assertEquals( 15, minpq.delMin() );
        assertEquals(  2, minpq.size()    );
    }
}
```

Test Suite Example 2

```java
import static org.junit.Assert.*;

/**
 * Suite to test MinHeap
 */
public class TestMinHeap
{
    @org.junit.Test (expected =
                                java.util.NoSuchElementException.class)
    public void testDelMinException()
    {
        MinPQ minpq = new MinHeap();
        minpq.insert(2);
        minpq.delMin();
        minpq.delMin();
    }
}
```

Compiling with Junit

http://www.vogella.com/tutorials/JUnit/article.html#juniteclipse_code

- Download junit (or use the one provided on blackboard)
  - you'll be compiling with -cp
- Put it somewhere
  - Need access to functions in junit
    - those are in junit-4.11.jar
  - if it's in your project directory, you can call it as
    - javac -cp .:junit-04.11.jar *.java //windows
    - javac -cp .;junit-04.11.jar *.java //unix

Running Tests

- Command line - like running any other class
  - javac -cp .:junit-4.11.jar *.java
  - java -cp .:junit-4.11.jar PA1Tests
  - windows users, use ; instead of : for -cp
- What do these lines do?
- IDEs usually provide niceness

Example for Windows

```
> javac -cp .;.\yourGMUUserName;junit-4.11.jar
*.java

> java -cp .;.\yourGMUUserName;junit-4.11.jar
PA1Tests JUnit version 4.11
........
Time: 0.03

OK (8 tests)
```

Example for Unix

```
> javac -cp .:./yourGMUUserName:junit-4.11.jar
*.java

> java -cp .:./yourGMUUserName:junit-4.11.jar
PA1Tests JUnit version 4.11
........
Time: 0.03

OK (8 tests)
```

Propositional Logic

Keith Devlin, Introduction to Mathematical Thinking

- Logic based on propositions. Proposition is any statement that can result in a true or false. Usually denoted p, q, r, etc.
- Define relations between propositions
  - Conjunction $p \wedge q$
  - Disjunction $p \vee q$
  - Negation $\neg p$
  - Conditional $p \Rightarrow q$
  - Biconditional $p \Leftrightarrow q \text{ same as } (p \Rightarrow q) \wedge (q \Rightarrow p)$
- Statements are combined using these operators

Propositional Logic Quantifiers

- Quantifiers
  - For all $(\forall \text{ set})[\ (\text{statement})\ ]$
    $(\forall \text{ x} \in \mathbb{N})[\ (p \wedge q)\ ]$
  - There exists $(\exists \text{ set})[\ (\text{statement})\ ]$
    $(\exists \text{ x} \in \mathbb{N})[(p \vee \neg q) \Rightarrow r]$
- Defines the variables that can be used in the statement

Truth Tables

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

p	$\neg p$
T	F
F	T

p	q	$p \Rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

p	q	$p \Leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Proof Motivation

- Establish the truth of a statement
  - Logically sound
- Communicate to others
  - Many statements can be proved in a number of ways. Better proofs are those that are easiest to communicate
  - Similar to code, comment because while you're writing you understand, a year later even the author is confused
- Proofs take years to master
  - No cookie cutter, but some guidelines

Proof Guidelines

- Truth Tables
  - Not always possible, only for small problems
- Proof by Contradiction
  - Need to understand how to negate expression
  - Good approach if no obvious place to start
- Proof by Cases
- Proof by Induction
  - Only works for statements involving the set of natural numbers
- Proof by Construction

Example Contradiction Proof 1/3

- Prove: (Insertion sort is stable)
  - Define: k index position of an item before sort and i is the index of an item after the sort
  - Stable:

$$A = (\forall a, b \in \{Items\})[((a = b) \wedge (k_a < k_b)) \Rightarrow (i_a < i_b)]$$

- Establish true statement(s) from the algorithm

  (1) Only compare and swap adjacent items

  (2) If swapped, then $a < b$

Example Contradiction Proof 2/3

- Proof by **Contradiction**

$$A = (\forall a, b \in \{Items\})[((a = b) \wedge (k_a < k_b)) \Rightarrow (i_a < i_b)]$$

Assume to the contrary

$$\neg A = (\exists a, b \in \{Items\})[((a = b) \wedge (k_a < k_b)) \wedge \neg(i_a < i_b)]$$

- Proceed with reasoning from the contrary statement until a false statement is encountered, usually of the form

$$p \wedge \neg p$$

Example Contradiction Proof 3/3

Because $k_a < k_b$, we have $\neg(i_a < i_b) = (i_a > i_b)$
By (1), because $(i_a > i_b)$, we swapped a and b.
By (2), $a < b$, because a swap occured
But $\neg A$ assumed to be true requiring $a = b$
Contradiction: $(a = b) \wedge (a < b)$

- Provided the reasoning is correct, starting from a supposedly true statement and arriving at a false consequence can only mean that the contrary statement was false

$$\neg(\neg A) = A$$

- Usually ends with Q.E.D.

Simpler Contradiction Proof

- Prove: Insertion sort is stable
- Proof by Contradiction: Insertion sort not stable

  Means that at some point item a, where a was positioned prior to b before the sort, was swapped with an equal item b. But insertion sort only swaps items if a is strictly less than b. Contradiction. Q.E.D.

Induction Proof

Weiss 7.2

- To prove a statement of the form

$$(\forall\ \mathrm{n} \in \mathbb{N})[\ (A(n))\ ]$$

- Prove the following two statements

$$\text{Initial step (1) } (A(1))$$

$$\text{Induction step (2) } (\forall\ \mathrm{n} \in \mathbb{N})[\ (A(n) \Rightarrow A(n+1)\ ]$$

- But wait, these two statements are not the same as statement we wish to prove.
    - The “Principle of Mathematical Induction”
- Proof that these two imply the original statement can be shown by contradiction (omitted).

Induction Analogies

- Dominoes
  - Start first one falling
  - If previous domino falls, so does next one
  - On through infinity
- Climbing fire escapes
  - Can get to the lowest floor escape
  - From any floor, can get to the next higher floor
  - On to infinitely high building

Induction Proof Steps

- Initial Step (1)
  - Usually easy (may not start with 1)

    $A(1) \text{ or } A(n_0)$
- Induction Step (2)

  $(\forall \text{ n} \in \mathbb{N}) \; [A(n) \Rightarrow A(n+1)]$
  - Need to prove a conditional. If we assume antecedent to be true for some arbitrary *k* and, using this, show that the consequent also has to be true, then the conditional is proven
  - Is this correct? Look at proof table for the conditional
  - Assuming *A(k)* to be true is known as the "Inductive Hypothesis"

Example Induction Proof 1/5

```java
// Returns (n(n+1))/2
public int triangleSum( int n )
{
    if( n == 1 ) return 1;
    return n + triangleSum(n-1);
}
```

- Prove by **Induction**: For any integer n>0, the sum of the first n integers given by summing from 1 to n (1+2+...+(n-1)+n) is equal to n(n+1) / 2

$$(\forall n \in \mathbb{N})[1 + 2 + \cdots + (n-1) + n = \frac{n(n+1)}{2}]$$

$$(\forall n \in \mathbb{N})[A(n)]$$

Example Induction Proof 2/5

- Initial Step, proves (1)

$$A(1)\ 1 = \frac{1(1+1)}{2}$$

- Induction Step
  - Inductive Hypothesis: Assume *A(k)* true for some *k*

$$(\text{for some } k,\ 1 \geq n \geq k)\ \ A(k)$$

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

Example Induction Proof 3/5

- Induction Step:
  - Start with *A(k)* to deduce *A(k+1)*
  - Start with *A(k+1)* reduce where you use *A(k)*
- Either way, write down *A(k)*, assumed true, and the target, *A(k+1)*

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

algebraic manipulation

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

Example Induction Proof 4/5

- Look at *A(k)* and try to deduce *A(k+1)*

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

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

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

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

Example Induction Proof 5/5

- Deduced *A(k+1)* using *A(k)*. This proves the induction step (2).

$$(\forall\ \mathrm{n} \in \mathbb{N})[\ (A(n) \Rightarrow\ A(n+1))\ ]$$

- We have shown (1) and (2), thus, by the principle of mathematical induction, the identity holds for all *n*. Q.E.D.

$$(\forall\ \mathrm{n} \in \mathbb{N})[\ (A(n))\ ]$$

Induction Proof Summary

By Keith Devlin, Introduction to Mathematical Thinking

- Want to prove some statement *A(n)* is true for all natural numbers.
- First prove $A(n_0)$, *usually* $n_0 = 1$
  - usually a matter of simple observation
- Give an algebraic argument to establish the conditional "*if A(k) then A(k+1) for some k*"
  - Reduce *A(k+1)* to a form where you use *A(k)*
- Conclusion: By the "Principle of Mathematical Induction", this proves *A(n)* is true for all $n >= n_0$ natural numbers.

Induction and Recursion

- Recursive algorithms can be proven correct by induction (Weiss 7.3.2)
- Induction
  - Start with initial (base case)
  - Proceed one step at a time towards some k
- Recursion
  - Start with given k
  - Continue one step at a time backwards towards the base case (initial)
- Conceptually, mirror images, induction ascends, recursion descends

Assignments

- PA5
  - Implement MinPQ using a binary heap.