Proving Program Properties – Review

Curt Clifton
Rose-Hulman Institute of Technology

HOW DO YOU EAT AN ELEPHANT?

ONE BITE AT A TIME

Techniques

- Finding weakest pre-conditions
  - Work backward from post-condition
- Proving programs correct
  - Work forward or backward
  - Each assertion implies the next down the page
- Proving total correctness

Inference Rules

- Assignment
- Sequencing/composition
- If-then-else
- If-then
- While loop
- Procedures

More Notation

- Loop invariants:
  - **maintaining** and **decreasing**
- Quantification
  - **\forall**, **\max**, **\min**, **\sum**, **\product**, **\exists**

Method Specifications

- Pre-condition: **requires**
- Frame condition: **assignable**
- Post-condition: **ensures**
  - **\old(), \result**

Recursion

- How do we prove a recursive procedure is correct?
- Assume that it is!
- Really proof by induction
  - Some path through the procedure must not recurse – base case
  - Other paths recurse – induction step

DBC for Classes

- 1. Specify the data…
  - Using **spec_public** fields for now
  - Later we'll see how to be more abstract
- 2. Specify class **invariant**s
- 3. Specify each public method/constructor using:
  - **requires**, **assignable**, and **ensures**