Review: Hoare Logic Rules

- *wp*(x := E, P) = [E/x] P
- *wp*(S;T, Q) = *wp*(S, *wp*(T, Q))
- *wp*(if B then S else T, Q)
  = B $\Rightarrow$ *wp*(S,Q) && $\neg$B $\Rightarrow$ *wp*(T,Q)

Proving loops correct

- First consider *partial correctness*
  - The loop may not terminate, but if it does, the postcondition will hold
- {P} while B do S {Q}
  - Find an invariant Inv such that:
    - P $\Rightarrow$ Inv
      - The invariant is initially true
    - { Inv && B } S {Inv}
      - Each execution of the loop preserves the invariant
    - (Inv && $\neg$B) $\Rightarrow$ Q
      - The invariant and the loop exit condition imply the postcondition

Loop Example

- Prove array sum correct

```
{ N ≥ 0 }
j := 0;
s := 0;

while (j < N) do

    s := s + a[j];
    j := j + 1;

end
{ s = (Σi | 0≤i<N • a[i]) }
```

16 November 2005

Loop Example

- Prove array sum correct

```
{ N ≥ 0 }
j := 0;
s := 0;
{ 0 ≤ j ≤ N && s = (Σi | 0≤i<j • a[i]) }
while (j < N) do
    {0 ≤ j ≤ N && s = (Σi | 0≤i<j • a[i]) && j < N}
    s := s + a[j];
    j := j + 1;
    {0 ≤ j ≤ N && s = (Σi | 0≤i<j • a[i]) }
end
{ s = (Σi | 0≤i<N • a[i]) }
```

16 November 2005

Proof Obligations

- Invariant is initially true
  { N ≥ 0 }
  j := 0;
  s := 0;
  { 0 ≤ j ≤ N && s = (Σi | 0≤i<j • a[i]) }
- Invariant is maintained
  {0 ≤ j ≤ N && s = (Σi | 0≤i<j • a[i]) && j < N}
  s := s + a[j];
  j := j + 1;
  {0 ≤ j ≤ N && s = (Σi | 0≤i<j • a[i]) }
- Invariant and exit condition implies postcondition
  0 ≤ j ≤ N && s = (Σi | 0≤i<j • a[i]) && j ≥ N
  ⇒ s = (Σi | 0≤i<N • a[i])

Proof Obligations

- Invariant is initially true
  { N ≥ 0 }
  { 0 ≤ 0 ≤ N && 0 = (Σi | 0≤i<0 • a[i]) } *// by assignment rule*
  j := 0;
  { 0 ≤ j ≤ N && 0 = (Σi | 0≤i<j • a[i]) } *// by assignment rule*
  s := 0;
  { 0 ≤ j ≤ N && s = (Σi | 0≤i<j • a[i]) }
- Need to show that:
  (N ≥ 0) ⇒ (0 ≤ 0 ≤ N && 0 = (Σi | 0≤i<0 • a[i]))

= (N ≥ 0) ⇒ (0 ≤ N && 0 = 0) *// 0 ≤ 0 is true, empty sum is 0*
= (N ≥ 0) ⇒ (0 ≤ N) *// 0=0 is true, P && true is P*
= true

Proof Obligations

- Invariant is maintained
  $\{0 \le j \le N$ && $s = (\Sigma i \mid 0 \le i < j \bullet a[i])$ && $j < N\}$
  $\{0 \le j+1 \le N$ && $s+a[j] = (\Sigma i \mid 0 \le i < j+1 \bullet a[i])\}$ *// by assignment rule*
  $s := s + a[j];$
  $\{0 \le j+1 \le N$ && $s = (\Sigma i \mid 0 \le i < j+1 \bullet a[i])\}$ *// by assignment rule*
  $j := j + 1;$
  $\{0 \le j \le N$ && $s = (\Sigma i \mid 0 \le i < j \bullet a[i])\}$
- Need to show that:
  $(0 \le j \le N$ && $s = (\Sigma i \mid 0 \le i < j \bullet a[i])$ && $j < N)$
  $\Rightarrow (0 \le j+1 \le N$ && $s+a[j] = (\Sigma i \mid 0 \le i < j+1 \bullet a[i]))$

= $(0 \le j < N$ && $s = (\Sigma i \mid 0 \le i < j \bullet a[i]))$
  $\Rightarrow (0 \le j < N$ && $s+a[j] = (\Sigma i \mid 0 \le i < j+1 \bullet a[i]))$ *// simplify bounds of j*

= $(0 \le j < N$ && $s = (\Sigma i \mid 0 \le i < j \bullet a[i]))$
  $\Rightarrow (0 \le j < N$ && $s+a[j] = (\Sigma i \mid 0 \le i < j \bullet a[i]) + a[j])$ *// separate last part of sum*

= $(0 \le j < N$ && $s = (\Sigma i \mid 0 \le i < j \bullet a[i]))$
  $\Rightarrow (0 \le j < N$ && $s = (\Sigma i \mid 0 \le i < j \bullet a[i]))$ *// subtract a[j] from both sides*

= true

16 November 2005

Proof Obligations

- Invariant and exit condition implies postcondition
  $0 \le j \le N$ && $s = (\Sigma i \mid 0 \le i < j \bullet a[i])$ && $j \ge N$
  $\Rightarrow s = (\Sigma i \mid 0 \le i < N \bullet a[i])$

= $0 \le j$ && $j = N$ && $s = (\Sigma i \mid 0 \le i < j \bullet a[i])$
  $\Rightarrow s = (\Sigma i \mid 0 \le i < N \bullet a[i])$
  // because $(j \le N$ && $j \ge N) = (j = N)$

= $0 \le N$ && $s = (\Sigma i \mid 0 \le i < N \bullet a[i]) \Rightarrow s = (\Sigma i \mid 0 \le i < N \bullet a[i])$
  // by substituting N for j, since j = N

= true *// because P && Q ⇒ Q*

16 November 2005

Invariant Intuition

- For code without loops, we are simulating execution directly
  - We prove one Hoare Triple for each statement, and each statement is executed once
- For code with loops, we are doing *one* proof of correctness for *multiple* loop iterations
  - Don't know how many iterations there will be
  - Need our proof to cover all of them
  - The invariant expresses a *general* condition that is true for every execution, but is still strong enough to give us the postcondition we need
  - This tension between generality and precision can make coming up with loop invariants hard

Total Correctness for Loops

- {P} while B do S {Q}
- Partial correctness:
  - Find an invariant Inv such that:
    - $P \Rightarrow Inv$
      - The invariant is initially true
    - { Inv && B } S {Inv}
      - Each execution of the loop preserves the invariant
    - $(Inv \;\&\&\; \neg B) \Rightarrow Q$
      - The invariant and the loop exit condition imply the postcondition
- Termination bound
  - Find a *variant function* v such that:
    - $(Inv \;\&\&\; B) \Rightarrow v > 0$
      - The variant function evaluates to a finite integer value greater than zero at the beginning of the loop
    - { Inv && B && v=V } S {v < V}
      - The value of the variant function decreases each time the loop body executes (here V is a constant)

Total Correctness Example

while (j < N) do

  {0 ≤ j ≤ N && s = (Σi | 0≤i<j • a[i]) && j < N}

  s := s + a[j];

  j := j + 1;

  {0 ≤ j ≤ N && s = (Σi | 0≤i<j • a[i]) }

end

- Variant function for this loop?
  - N-j

Guessing Variant Functions

- Loops with an index
  - N ± i
  - Applies if you always add or always subtract a constant, and if you exit the loop when the index reaches some constant
  - Use N-i if you are incrementing i, N+i if you are decrementing i
  - Set N such that N ± i ≤ 0 at loop exit
- Other loops
  - Find an expression that is an upper bound on the number of iterations left in the loop

Additional Proof Obligations

- Variant function for this loop: N-j
- To show: variant function initially positive
  (0 ≤ j ≤ N && s = (Σi | 0≤i<j • a[i]) && j < N)
      ⇒ N-j > 0
- To show: variant function is decreasing
  {0 ≤ j ≤ N && s = (Σi | 0≤i<j • a[i]) && j < N && N-j = V}
  s := s + a[j];
  j := j + 1;
  {N-j < V}

Additional Proof Obligations

- To show: variant function initially positive
  (0 ≤ j ≤ N && s = (Σi | 0≤i<j • a[i]) && j < N)
      ⇒ N-j > 0

= (0 ≤ j ≤ N && s = (Σi | 0≤i<j • a[i]) && j < N)
      ⇒ N > j   *// added j to both sides*

= true   *// (N > j) = (j < N), P && Q ⇒ P*

Additional Proof Obligations

- To show: variant function is decreasing
  $\{0 \le j \le N \;\&\&\; s = (\Sigma i \mid 0 \le i < j \bullet a[i]) \;\&\&\; j < N \;\&\&\; N-j = V\}$
  $\{N-(j+1) < V\}$ *// by assignment*
  s := s + a[j];
  $\{N-(j+1) < V\}$ *// by assignment*
  j := j + 1;
  $\{N-j < V\}$
- Need to show:
  $(0 \le j \le N \;\&\&\; s = (\Sigma i \mid 0 \le i < j \bullet a[i]) \;\&\&\; j < N \;\&\&\; N-j = V)$
  $\Rightarrow (N-(j+1) < V)$

Assume $0 \le j \le N \;\&\&\; s = (\Sigma i \mid 0 \le i < j \bullet a[i]) \;\&\&\; j < N \;\&\&\; N-j = V$
By weakening we have $N-j = V$
Therefore $N-j-1 < V$
But this is equivalent to $N-(j+1) < V$, so we are done.

Factorial

{ N ≥ 1 }
k := 1
f := 1
while (k < N) do
    f := f * k
    k := k + 1
end
{ f = N! }

- Loop invariant?
  - f = Pi(0 < i < k)(i) AND k <= N
  - what if we initialize k to 5?
- Variant function?

Factorial

$\{ N \geq 1 \}$
$\{ 1 = 1! \;\&\&\; 0 \leq 1 \leq N \}$
k := 1
$\{ 1 = k! \;\&\&\; 0 \leq k \leq N \}$
f := 1
$\{ f = k! \;\&\&\; 0 \leq k \leq N \}$
while (k < N) do
    $\{ f = k! \;\&\&\; 0 \leq k \leq N \;\&\&\; k < N \;\&\&\; N\text{-}k = V\}$
    $\{ f*(k+1) = (k+1)! \;\&\&\; 0 \leq k+1 \leq N \;\&\&\; N\text{-}(k+1) < V\}$
    k := k + 1
    $\{ f*k = k! \;\&\&\; 0 \leq k \leq N \;\&\&\; N\text{-}k < V\}$
    f := f * k
    $\{ f = k! \;\&\&\; 0 \leq k \leq N \;\&\&\; N\text{-}k < V\}$
end
$\{ f = k! \;\&\&\; 0 \leq k \leq N \;\&\&\; k \geq N\}$
$\{ f = N! \}$

Factorial Obligations (1)

$(N \geq 1) \Rightarrow (1 = 1! \;\&\&\; 0 \leq 1 \leq N)$
$= (N \geq 1) \Rightarrow (1 \leq N)$ *// because 1 = 1! and 0 ≤ 1*
$=$ true *// because* $(N \geq 1) = (1 \leq N)$

Factorial Obligations (2)

(f = k! && 0 ≤ k ≤ N && k < N && N-k = V)
⇒ (f*(k+1) = (k+1)! && 0 ≤ k+1 ≤ N && N-(k+1)< V)
= (f = k! && 0 ≤ k < N && N-k = V)
⇒ (f*(k+1) = k!*(k+1) && 0 ≤ k+1 ≤ N && N-k-1< V)
// by simplification and (k+1)!=k!(k+1)*
Assume (f = k! && 0 ≤ k < N && N-k = V)
Check each RHS clause:

- (f*(k+1) = k!*(k+1))
  = (f = k!) *// division by (k+1) (nonzero by assumption)*
  = true *// by assumption*
- 0 ≤ k+1
  = true *// by assumption that 0 ≤ k*
- k+1 ≤ N
  = true *// by assumption that k < N*
- N-k-1< V
  = N-k-1 < N-k *// by assumption that N-k = V*
  = N-1 < N *// by addition of k*
  = true *// by properties of <*

Factorial Obligations (3)

(f = k! && 0 ≤ k ≤ N && k ≥ N) ⇒ (f = N!)
Assume f = k! && 0 ≤ k ≤ N && k ≥ N
Then k=N by k ≤ N && k ≥ N
So f = N! by substituting k=N