Review: Hoare Logic Rules

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

Proving loops correct

- First consider \textit{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:
    - \( P \Rightarrow \text{Inv} \)
      - The invariant is initially true
    - \{ Inv \&\& B \} S \{Inv\}
      - Each execution of the loop preserves the invariant
    - \( \text{Inv} \&\& \neg B \Rightarrow Q \)
      - The invariant and the loop exit condition imply the postcondition
Loop Example

- Prove array sum correct

{ $N \geq 0$ }

j := 0;

s := 0;

while (j < N) do

s := s + a[j];

j := j + 1;

end

{ $s = (\Sigma i \mid 0 \leq i < N \cdot a[i])$ }

---

16 November 2005

---

Loop Example

- Prove array sum correct

{ $N \geq 0$ }

j := 0;

s := 0;

{ $0 \leq j \leq N \&\& s = (\Sigma i \mid 0 \leq i < j \cdot a[i])$ }

while (j < N) do

{ $0 \leq j \leq N \&\& s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \&\& j < N$ }

s := s + a[j];

j := j + 1;

end

{ $s = (\Sigma i \mid 0 \leq i < N \cdot a[i])$ }

---

16 November 2005
Proof Obligations

• Invariant is initially true
  \{ N \geq 0 \}
  j := 0;
  s := 0;
  \{ 0 \leq j \leq N \&\& s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \}

• Invariant is maintained
  \{0 \leq j \leq N \&\& s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \&\& j < N\}
  s := s + a[j];
  j := j + 1;
  \{0 \leq j \leq N \&\& s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \}

• Invariant and exit condition implies postcondition
  0 \leq j \leq N \&\& s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \&\& j \geq N
  \Rightarrow s = (\Sigma i \mid 0 \leq N \cdot a[i])

16 November 2005
Proof Obligations

• Invariant is maintained
  \[ \begin{align*}
  &0 \leq j \leq N \land s = (\sum_{i=0}^{j} a[i]) \\
  &0 \leq j+1 \leq N \land s + a[j] = (\sum_{i=0}^{j+1} a[i]) \quad \text{by assignment rule}
  \end{align*} \]

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

\[0 \leq j+1 \leq N \land s = (\sum_{i=0}^{j+1} a[i]) \quad \text{by assignment rule}\]

\[ j := j + 1; \]

\[0 \leq j \leq N \land s = (\sum_{i=0}^{j} a[i]) \]

• Need to show that:
  \[ \begin{align*}
  &0 \leq j \leq N \land s = (\sum_{i=0}^{j} a[i]) \land j < N \\
  \Rightarrow & \quad (0 \leq j+1 \leq N \land s + a[j] = (\sum_{i=0}^{j+1} a[i])) \\
  & = (0 \leq j < N \land s = (\sum_{i=0}^{j} a[i])) \\
  \Rightarrow & \quad (0 \leq j < N \land s + a[j] = (\sum_{i=0}^{j} a[i])) \quad \text{simplify bounds of } j \\
  & = (0 \leq j < N \land s = (\sum_{i=0}^{j} a[i])) \\
  \Rightarrow & \quad (0 \leq j < N \land s + a[j] = (\sum_{i=0}^{j} a[i]) + a[j]) \quad \text{separate last part of sum} \\
  & = (0 \leq j < N \land s = (\sum_{i=0}^{j} a[i])) \\
  \Rightarrow & \quad (0 \leq j < N \land s = (\sum_{i=0}^{j} a[i])) \quad \text{subtract } a[j] \text{ from both sides} \\
  & = \text{true} \quad \text{by assignment rule}
  \end{align*} \]

\[ \text{by substituting } N \text{ for } j, \text{ since } j = N \]

16 November 2005

---

Proof Obligations

• Invariant and exit condition implies postcondition
  \[ \begin{align*}
  &0 \leq j \leq N \land s = (\sum_{i=0}^{j} a[i]) \land j \geq N \\
  \Rightarrow & \quad s = (\sum_{i=0}^{N} a[i]) \\
  & = 0 \land j = N \land s = (\sum_{i=0}^{N} a[i]) \\
  \Rightarrow & \quad s = (\sum_{i=0}^{N} a[i]) \\
  & = 0 \land N \land s = (\sum_{i=0}^{N} a[i]) \quad \text{because } j \leq N \land j \geq N = (j = N) \\
  & = 0 \land N \land s = (\sum_{i=0}^{N} a[i]) \Rightarrow s = (\sum_{i=0}^{N} a[i]) \\
  & = \text{true} \quad \text{because } P \land Q \Rightarrow Q
  \end{align*} \]

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
  \[0 \leq j \leq N \land s = (\sum_{i=0}^{j} a[i]) \land j < N\]
  \[\Rightarrow N-j > 0\]
- To show: variant function is decreasing
  \[\{0 \leq j \leq N \land s = (\sum_{i=0}^{j} a[i]) \land j < N \land N-j = V\}
  s := s + a[j];
  j := j + 1;
  \{N-j < V\}

Additional Proof Obligations

- To show: variant function initially positive
  \[0 \leq j \leq N \land s = (\sum_{i=0}^{j} a[i]) \land j < N\]
  \[\Rightarrow N-j > 0\]
  \[= (0 \leq j \leq N \land s = (\sum_{i=0}^{j} a[i]) \land j < N)\]
  \[\Rightarrow N > j \quad // added j to both sides\]
  \[= \text{true} \quad // (N > j) = (j < N), P \land Q \Rightarrow P\]
Additional Proof Obligations

- To show: variant function is decreasing
  \( \{0 \leq j \leq N \land s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \land j < N \land N-j = V\} \)
  \( \{N-(j+1) < V\} \quad \text{// by assignment} \)
  \( s := s + a[j]; \)
  \( \{N-(j+1) < V\} \quad \text{// by assignment} \)
  \( j := j + 1; \)
  \( \{N-j < V\} \)
- Need to show:
  \( (0 \leq j \leq N \land s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \land j < N \land N-j = V) \)
  \( \Rightarrow (N-(j+1) < V) \)
Assume \( 0 \leq j \leq N \land s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \land j < N \land N-j = V \)
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.

16 November 2005

Factorial

\{ \( N \geq 1 \) \}
\( k := 1 \)
\( f := 1 \)
while (\( k < N \)) do
  \( f := f \times k \)
  \( k := k + 1 \)
end
\{ \( f = N! \) \}

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

16 November 2005
Factorial

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

16 November 2005

Factorial Obligations (1)

(N \geq 1) \Rightarrow (1 = 1! \&\& 0 \leq 1 \leq N)
= (N \geq 1) \Rightarrow (1 \leq N) \quad // \text{because } 1 = 1! \text{ and } 0 \leq 1
= \text{true} \quad // \text{because } (N \geq 1) = (1 \leq N)

16 November 2005
Factorial Obligations (2)

\[(f = k! && 0 \leq k \leq N && k < N && N-k = V) \Rightarrow (f^{(k+1)} = (k+1)! && 0 \leq k+1 \leq N && N-(k+1) < V)\]

== \[(f = k! && 0 \leq k < N && N-k = V) \Rightarrow (f^{(k+1)} = k!(k+1) && 0 \leq k+1 \leq N && N-k-1 < V)\]

// by simplification and (k+1)!=k!(k+1)

Assume \((f = k! && 0 \leq k < N && N-k = V)\)

Check each RHS clause:

- \((f^{(k+1)} = k!(k+1))\)
  = \((f = k!)) // division by \((k+1)\) (nonzero by assumption)
  = true // by assumption

- \(0 \leq k+1\)
  = true // by assumption that \(0 \leq k\)

- \(k+1 \leq N\)
  = true // by assumption that \(k < N\)

- \(N-k-1 < V\)
  = \(N-k < N-k \) // by assumption that \(N-k = V\)
  = \(N-1 < N \) // by addition of \(k\)
  = true // by properties of <

Factorial Obligations (3)

\[(f = k! && 0 \leq k \leq N && k \geq N) \Rightarrow (f = N!)\]

Assume \(f = k! && 0 \leq k \leq N && k \geq N\)

Then \(k=N\) by \(k \leq N && k \geq N\)

So \(f = N!\) by substituting \(k=N\)