Principles of Software Construction: Objects, Design, and Concurrency

Hoare Logic, Part 2

Jonathan Aldrich
Side Note: Why Weakest Preconditions?

• 15-122 teaches a (somewhat less formal) approach based on fresh variables
  • Increment x in a loop \( x' = x + 1 \)

• This approach has limitations
  • Sequences
    \[
    \begin{align*}
    x &:= x \times 2; \quad \text{ // } x' \\
    x &:= x + 1; \quad \text{ // } x''
    \end{align*}
    \]
  • Conditionals
    \[
    \begin{align*}
    \text{if (…)} & \quad x := x \times 2; \quad \text{ // } x' \\
    \text{else} & \quad y := y + 1; \quad \text{ // } y' \text{ – but we must also assume } x' = x \text{ here}
    \end{align*}
    \]

• Weakest preconditions scales better
  • No extra variables, no virtual assignments in branches
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 $T$, $Q$)
  
  $= B \Rightarrow wp(S,Q) \&\& \neg B \Rightarrow wp(T,Q)$
Hoare Logic Rules

• Loops
  • \{ P \} while (i < x) f = f * i; i := i + 1 \{ f = x! \}
  • What is the weakest precondition P?

• Intuition
  • Must prove by induction
    • Only way to generalize across number of times loop executes
  • Need to guess induction hypothesis
    • Base case: precondition P
    • Inductive case: should be preserved by executing loop body
Proving loops correct

- **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 ⇒ Inv
      - The invariant is initially true
    - `{ Inv && B } S {Inv}`
      - Each execution of the loop preserves the invariant
    - `(Inv && ¬B) ⇒ Q`
      - The invariant and the loop exit condition imply the postcondition
Quick Quiz

Consider the following program:

\[
\{ N \geq 0 \}
\]
\[
i := 0;\]
\[
\text{while } (i < N) \text{ do} \]
\[
\quad i := N
\]
\[
\{ i = N \}
\]

Which of the following conditions are loop invariants that are sufficient to prove the postcondition?
For those that are incorrect, explain why.

A) \( i = 0 \)  
B) \( i = N \)  
C) \( N \geq 0 \)  
D) \( i \leq N \)
Quick Quiz

Consider the following program:

\[
\begin{align*}
\{ \ N \geq 0 \} \\
\text{i := 0;} \\
\text{while (i < N) do} \\
\quad \text{i := N} \\
\{ \ i = N \}
\end{align*}
\]

Which of the following conditions are loop invariants that are sufficient to prove the postcondition?

For those that are incorrect, explain why.

A) \( i = 0 \)
   // not an invariant; not preserved by loop execution

B) \( i = N \)
   // not an invariant; not initially true

C) \( N \geq 0 \)
   // a loop invariant, but insufficient to prove postcondition

D) \( i \leq N \)
   // correct loop invariant, sufficient to prove postcondition

Correctness Conditions

\[
P \Rightarrow \text{Inv} \\
The \ invariant \ is \ initially \ true \\
\{ \ \text{Inv} \&\& \ B \} \ S \ \{\text{Inv}\} \\
\text{Loop \ preserves \ the \ invariant} \\
(\text{Inv} \&\& \neg B) \Rightarrow Q \\
\text{Invariant \ and \ exit \ implies \ postcondition}
\]
Loop Example

- Prove array sum correct

\{ N \geq 0 \}

j := 0;
s := 0;

while (j < N) do

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

end

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

How can we find a loop invariant?
Loop Example

• Prove array sum correct

\{ N \geq 0 \}

j := 0;
s := 0;

while (j < N) do

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

end

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

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

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

• Prove array sum correct

\[
\{ N \geq 0 \}
\]

\[
j := 0; \\
s := 0; \\
\{ 0 \leq j \leq N \land s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \}
\]

while \((j < N)\) do

\[
\begin{align*}
&\quad j := j + 1; \\
&\quad s := s + a[j]; \\
&\quad \{0 \leq j \leq N \land s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \}
\end{align*}
\]

end

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

• Prove array sum correct

\{ N \geq 0 \} 

\begin{align*}
  j &:= 0; \\
  s &:= 0; \\
  \{ 0 \leq j \leq N && s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \} \\
  \text{while } (j < N) \text{ do} \\
  &\quad \{ 0 \leq j \leq N && s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) && j < N \} \\
  &\quad j := j + 1; \\
  &\quad s := s + a[j]; \\
  &\quad \{ 0 \leq j \leq N && s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \} \\
  \text{end} \\
  \{ s = (\Sigma i \mid 0 \leq i < N \cdot a[i]) \} 
\end{align*}
Loop Example

- Prove array sum correct

\{ N \geq 0 \}

\text{j := 0;\}

\text{s := 0;\}

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

\text{while (j < N) do\}

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

\text{j := j + 1;\}

\text{s := s + a[j];\}

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

\text{end\}

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

• Invariant is initially true

\[
\{ N \geq 0 \} \\
j := 0; \\
s := 0; \\
\{ 0 \leq j \leq N \land s = (\sum i \mid 0 \leq i < j \cdot a[i]) \} 
\]
Proof Obligations

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

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

• Invariant and exit condition imply 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 i < N \cdot a[i])
Proof Obligations

- Invariant is initially true
  \{ N \geq 0 \}

  j := 0;

  s := 0;
  \{ 0 \leq j \leq N \&\& s = (\sum_{0 \leq i < j} a[i]) \}
Proof Obligations

• Invariant is initially true
  \( \{ N \geq 0 \} \)

\[
j := 0;
\{ 0 \leq j \leq N \&\& 0 = \left( \sum_{i \mid 0 \leq i < j} a[i] \right) \} \quad // by assignment rule
s := 0;
\{ 0 \leq j \leq N \&\& s = \left( \sum_{i \mid 0 \leq i < j} a[i] \right) \}\]
Proof Obligations

• Invariant is initially true
  \{ N \geq 0 \}
  \{ 0 \leq 0 \leq N \&\& 0 = (\sum i \mid 0 \leq i < 0 \cdot a[i]) \} \quad // by assignment rule
  j := 0;
  \{ 0 \leq j \leq N \&\& 0 = (\sum i \mid 0 \leq i < j \cdot a[i]) \} \quad // by assignment rule
  s := 0;
  \{ 0 \leq j \leq N \&\& s = (\sum i \mid 0 \leq i < j \cdot a[i]) \}
Proof Obligations

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

• Need to show that:
  (N \geq 0) \Rightarrow (0 \leq 0 \leq N && 0 = (\Sigma i \mid 0\leq i<0 \cdot a[i]))
Proof Obligations

- Invariant is initially true
  \[
  \{ N \geq 0 \} \\
  \{ 0 \leq 0 \leq N \&\& 0 = (\Sigma i \mid 0 \leq i < 0 \cdot a[i]) \} \quad \text{// by assignment rule}
  \]
  \[
  j := 0; \\
  \{ 0 \leq j \leq N \&\& 0 = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \} \quad \text{// by assignment rule}
  \]
  \[
  s := 0; \\
  \{ 0 \leq j \leq N \&\& s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \}
  \]

- Need to show that:
  \[
  (N \geq 0) \Rightarrow (0 \leq 0 \leq N \&\& 0 = (\Sigma i \mid 0 \leq i < 0 \cdot a[i]))
  \]
  \[
  = (N \geq 0) \Rightarrow (0 \leq N \&\& 0 = 0) \quad \text{// 0 \leq 0 is true, empty sum is 0} 
  \]
Proof Obligations

• Invariant is initially true

\[
\begin{align*}
\{ N \geq 0 \} \\
\{ 0 \leq 0 \leq N \land \ 0 = (\sum i \mid 0 \leq i < 0 \cdot a[i]) \} & \text{ // by assignment rule} \\
\end{align*}
\]

\[
\begin{align*}
j := 0; \\
\{ 0 \leq j \leq N \land 0 = (\sum i \mid 0 \leq i < j \cdot a[i]) \} & \text{ // by assignment rule} \\
s := 0; \\
\{ 0 \leq j \leq N \land s = (\sum i \mid 0 \leq i < j \cdot a[i]) \} \\
\end{align*}
\]

• Need to show that:

\[
\begin{align*}
(N \geq 0) \Rightarrow (0 \leq 0 \leq N \land 0 = (\sum i \mid 0 \leq i < 0 \cdot a[i])) \\
= (N \geq 0) \Rightarrow (0 \leq N \land 0 = 0) & \text{ // 0 \leq 0 is true, empty sum is 0} \\
= (N \geq 0) \Rightarrow (0 \leq N) & \text{ // 0=0 is true, P \land \text{true} is P}
\end{align*}
\]
Proof Obligations

• Invariant is initially true
  \{ N \geq 0 \}
  \{ 0 \leq 0 \leq N && 0 = (\Sigma i | 0 \leq i < 0 \cdot a[i]) \} \quad // by assignment rule
  j := 0;
  \{ 0 \leq j \leq N && 0 = (\Sigma i | 0 \leq i < j \cdot a[i]) \} \quad // by assignment rule
  s := 0;
  \{ 0 \leq j \leq N && s = (\Sigma i | 0 \leq i < j \cdot a[i]) \} 

• Need to show that:
  (N \geq 0) \Rightarrow (0 \leq 0 \leq N && 0 = (\Sigma i | 0 \leq i < 0 \cdot a[i]))
  = (N \geq 0) \Rightarrow (0 \leq N && 0 = 0) \quad // 0 \leq 0 is true, empty sum is 0
  = (N \geq 0) \Rightarrow (0 \leq N) \quad // 0=0 is true, P && true is P
  = \text{true}
Proof Obligations

- Invariant is maintained
  \( \{0 \leq j \leq N \land s = (\sum_{i \leq j} a[i]) \land j < N\} \)

\[
j := j + 1;
\]

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

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

- **Invariant is maintained**
  \[\{0 \leq j \leq N \land s = (\sum_{i \leq j} a[i]) \land j < N\}\]

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

- **Invariant is maintained**
  \[
  \begin{align*}
  &\{0 \leq j \leq N \land \land s = (\sum_{i \leq j} a[i]) \land j < N\} \\
  &\{0 \leq j + 1 \leq N \land \land s + a[j+1] = (\sum_{i \leq j+1} a[i]) \} \quad // by assignment rule \\
  &j := j + 1; \\
  &\{0 \leq j \leq N \land \land s + a[j] = (\sum_{i \leq j} a[i]) \} \quad // by assignment rule \\
  &s := s + a[j]; \\
  &\{0 \leq j \leq N \land \land s = (\sum_{i \leq j} a[i]) \}
  \end{align*}
  \]
Proof Obligations

- **Invariant is maintained**
  \[
  \begin{align*}
  &\{0 \leq j \leq N \&\& s = (\sum i \mid 0 \leq i < j \cdot a[i]) \&\& j < N\} \\
  &\{0 \leq j + 1 \leq N \&\& s + a[j+1] = (\sum i \mid 0 \leq i < j+1 \cdot a[i])\} \quad \text{// by assignment rule} \\
  &j := j + 1; \\
  &\{0 \leq j \leq N \&\& s + a[j] = (\sum i \mid 0 \leq i < j \cdot a[i])\} \quad \text{// by assignment rule} \\
  &s := s + a[j]; \\
  &\{0 \leq j \leq N \&\& s = (\sum i \mid 0 \leq i < j \cdot a[i])\}
  \end{align*}
  \]

- **Need to show that:**
  \[
  (0 \leq j \leq N \&\& s = (\sum i \mid 0 \leq i < j \cdot a[i]) \&\& j < N) \\
  \Rightarrow (0 \leq j + 1 \leq N \&\& s + a[j+1] = (\sum i \mid 0 \leq i < j+1 \cdot a[i]))
  \]
Proof Obligations

- **Invariant is maintained**
  \[
  \{0 \leq j \leq N \land s = (\sum_{0 \leq i < j} a[i]) \land j < N\}
  \quad \{0 \leq j + 1 \leq N \land s + a[j+1] = (\sum_{0 \leq i < j+1} a[i])\} \quad // \text{by assignment rule}
  \]
  \[
  j := j + 1;
  \{0 \leq j \leq N \land s + a[j] = (\sum_{0 \leq i < j} a[i])\} \quad // \text{by assignment rule}
  \]
  \[
  s := s + a[j];\quad \{0 \leq j \leq N \land s = (\sum_{0 \leq i < j} a[i])\}
  \]

- **Need to show that:**
  \[
  \{0 \leq j \leq N \land s = (\sum_{0 \leq i < j} a[i]) \land j < N\}
  \implies (0 \leq j + 1 \leq N \land s + a[j+1] = (\sum_{0 \leq i < j+1} a[i]))
  \]
  \[
  = (0 \leq j < N \land s = (\sum_{0 \leq i < j} a[i]))
  \implies (-1 \leq j < N \land s + a[j+1] = (\sum_{0 \leq i < j+1} a[i])) \quad // \text{simplify bounds of } j
Proof Obligations

• Invariant is maintained
  \[0 \leq j \leq N \land s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \land j < N\]
  \[0 \leq j + 1 \leq N \land s + a[j+1] = (\Sigma i \mid 0 \leq i < j + 1 \cdot a[i])\]  // by assignment rule
  \[j := j + 1;\]
  \[0 \leq j \leq N \land s + a[j] = (\Sigma i \mid 0 \leq i < j \cdot a[i])\]  // by assignment rule
  \[s := s + a[j];\]
  \[0 \leq j \leq N \land s = (\Sigma i \mid 0 \leq i < j \cdot a[i])\]

• Need to show that:
  \[(0 \leq j \leq N \land s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \land j < N)\]
  \[\Rightarrow (0 \leq j + 1 \leq N \land s + a[j+1] = (\Sigma i \mid 0 \leq i < j + 1 \cdot a[i]))\]
  \[= (0 \leq j < N \land s = (\Sigma i \mid 0 \leq i < j \cdot a[i]))\]
  \[\Rightarrow (-1 \leq j < N \land s + a[j+1] = (\Sigma i \mid 0 \leq i < j + 1 \cdot a[i]))\]  // simplify bounds of j
  \[= (0 \leq j < N \land s = (\Sigma i \mid 0 \leq i < j \cdot a[i]))\]
  \[\Rightarrow (-1 \leq j < N \land s + a[j+1] = (\Sigma i \mid 0 \leq i < j \cdot a[i]) + a[j])\]  // separate last element
Proof Obligations

- Invariant is maintained
  \[
  \{0 \leq j \leq N \land s = (\sum_{i} | 0 \leq i < j \cdot a[i]) \land j < N\}
  \]
  \[
  \{0 \leq j + 1 \leq N \land s + a[j + 1] = (\sum_{i} | 0 \leq i < j+1 \cdot a[i])\} \quad \text{// by assignment rule}
  \]
  \[
  j := j + 1;
  \]
  \[
  \{0 \leq j \leq N \land s + a[j] = (\sum_{i} | 0 \leq i < j \cdot a[i])\} \quad \text{// by assignment rule}
  \]
  
- Need to show that:
  \[
  (0 \leq j \leq N \land s = (\sum_{i} | 0 \leq i < j \cdot a[i]) \land j < N) \\
  \quad \Rightarrow (0 \leq j + 1 \leq N \land s + a[j + 1] = (\sum_{i} | 0 \leq i < j+1 \cdot a[i]))
  \]
  \[
  = (0 \leq j < N \land s = (\sum_{i} | 0 \leq i < j \cdot a[i])
  \quad \Rightarrow (-1 \leq j < N \land s + a[j + 1] = (\sum_{i} | 0 \leq i < j + 1 \cdot a[i])) \quad \text{// simplify bounds of } j
  \]
  \[
  = (0 \leq j < N \land s = (\sum_{i} | 0 \leq i < j \cdot a[i])
  \quad \Rightarrow (-1 \leq j < N \land s + a[j + 1] = (\sum_{i} | 0 \leq i < j \cdot a[i]) + a[j]) \quad \text{// separate last element}
  \quad \text{// we have a problem – we need } a[j + 1] \text{ and } a[j] \text{ to cancel out}
Where’s the error?

- Prove array sum correct

\[
\{ N \geq 0 \} \\
j := 0; \\
s := 0;
\]

\begin{verbatim}
while (j < N) do \\
  j := j + 1; \\
  s := s + a[j]; \\
end
\end{verbatim}

\[
\{ s = (\sum_{i} \mid 0 \leq i < N \cdot a[i]) \}
\]
Where’s the error?

- Prove array sum correct

\[
\{ N \geq 0 \}
\]

\[
j := 0; \\
s := 0;
\]

while \((j < N)\) do

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

end

\[
\{ s = (\sum_{i} | 0 \leq i < N \cdot a[i]) \}
\]

Need to add element \textit{before} incrementing \(j\)
Corrected Code

- Prove array sum correct

\[
\{ N \geq 0 \} \\
j := 0; \\
s := 0; \\
\text{while } (j < N) \text{ do} \\
\quad s := s + a[j]; \\
\quad j := j + 1; \\
\text{end} \\
\{ s = (\sum_{i} | 0 \leq i < N \cdot a[i]) \} 
\]
Proof Obligations

- **Invariant is maintained**
  \[ 0 \leq j \leq N \land \land s = \sum_{0 \leq i < j} a[i] \land j < N \]

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

  \[
  j := j + 1;
  \]

  \[ 0 \leq j \leq N \land \land s = \sum_{0 \leq i < j} a[i] \]
Proof Obligations

- Invariant is maintained

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

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

\[
\{0 \leq j + 1 \leq N \&\& s = (\sum_{0 \leq i < j+1} a[i]) \}
\]

// by assignment rule

\[
j := j + 1;
\]

\[
\{0 \leq j \leq N \&\& s = (\sum_{0 \leq i < j} a[i]) \}
\]
Proof Obligations

- Invariant is maintained

\[
\begin{align*}
\{ 0 \leq j \leq N \land \text{s} = (\sum_{0 \leq i < j} a[i]) \land j < N \} \\
\{ 0 \leq j + 1 \leq N \land \text{s} + a[j] = (\sum_{0 \leq i < j+1} a[i]) \} & \quad \text{by assignment rule} \\
\text{s := s + a[j];} \\
\{ 0 \leq j + 1 \leq N \land \text{s} = (\sum_{0 \leq i < j+1} a[i]) \} & \quad \text{by assignment rule} \\
\text{j := j + 1;} \\
\{ 0 \leq j \leq N \land \text{s} = (\sum_{0 \leq i < j} a[i]) \} 
\end{align*}
\]
Proof Obligations

- Invariant is maintained
  \[
  \{0 \leq j \leq N \land s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \land j < N\} \\
  \{0 \leq j + 1 \leq N \land s + a[j] = (\Sigma i \mid 0 \leq i < j + 1 \cdot a[i])\} \quad \text{by assignment rule}
  \]
  \[
  s := s + a[j]; \\
  \{0 \leq j + 1 \leq N \land s = (\Sigma i \mid 0 \leq i < j + 1 \cdot a[i])\} \quad \text{by assignment rule}
  \]
  \[
  j := j + 1; \\
  \{0 \leq j \leq N \land s = (\Sigma i \mid 0 \leq i < j \cdot a[i])\}
  \]

- Need to show that:
  \[
  (0 \leq j \leq N \land s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \land j < N) \\
  \Rightarrow (0 \leq j + 1 \leq N \land s + a[j] = (\Sigma i \mid 0 \leq i < j + 1 \cdot a[i]))
  \]
Proof Obligations

- Invariant is maintained

\[ \{ 0 \leq j \leq N \land s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \land j < N \} \]
\[ \{ 0 \leq j + 1 \leq N \land s + a[j] = (\Sigma i \mid 0 \leq i < j + 1 \cdot a[i]) \} \quad \text{// by assignment rule} \]

\[ s := s + a[j]; \]
\[ \{ 0 \leq j + 1 \leq N \land s = (\Sigma i \mid 0 \leq i < j + 1 \cdot a[i]) \} \quad \text{// by assignment rule} \]
\[ j := j + 1; \]
\[ \{ 0 \leq j \leq N \land s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \} \]

- Need to show that:

\[ (0 \leq j \leq N \land s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \land j < N) \]
\[ \Rightarrow (0 \leq j + 1 \leq N \land s + a[j] = (\Sigma i \mid 0 \leq i < j + 1 \cdot a[i])) \]
\[ = (0 \leq j < N \land s = (\Sigma i \mid 0 \leq i < j \cdot a[i])) \]
\[ \Rightarrow (-1 \leq j < N \land s + a[j] = (\Sigma i \mid 0 \leq i < j + 1 \cdot a[i])) \quad \text{// simplify bounds of j} \]
Proof Obligations

• Invariant is maintained

\[
\begin{align*}
\{0 \leq j \leq N \land s = (\sum i \mid 0 \leq i < j \cdot a[i]) \land j < N\} \\
\{0 \leq j + 1 \leq N \land s + a[j] = (\sum i \mid 0 \leq i < j + 1 \cdot a[i])\} & \quad \text{\textit{\# by assignment rule}} \\
\text{s := s + a[j];} \\
\{0 \leq j + 1 \leq N \land s = (\sum i \mid 0 \leq i < j + 1 \cdot a[i])\} & \quad \text{\textit{\# by assignment rule}} \\
\text{j := j + 1;} \\
\{0 \leq j \leq N \land s = (\sum i \mid 0 \leq i < j \cdot a[i])\}
\end{align*}
\]

• Need to show that:

\[
(0 \leq j \leq N \land s = (\sum i \mid 0 \leq i < j \cdot a[i]) \land j < N) \\
\Rightarrow (0 \leq j + 1 \leq N \land s + a[j] = (\sum i \mid 0 \leq i < j + 1 \cdot a[i])) \\
= (0 \leq j < N \land s = (\sum i \mid 0 \leq i < j \cdot a[i])) \\
\Rightarrow (-1 \leq j < N \land s + a[j] = (\sum i \mid 0 \leq i < j + 1 \cdot a[i])) \quad \text{\textit{\# simplify bounds of \(j\)}} \\
= (0 \leq j < N \land s = (\sum i \mid 0 \leq i < j \cdot a[i])) \\
\Rightarrow (-1 \leq j < N \land s + a[j] = (\sum i \mid 0 \leq i < j \cdot a[i]) + a[j]) \quad \text{\textit{\# separate last part of sum}}
\]
Proof Obligations

- **Invariant is maintained**
  
  \[
  \{0 \leq j \leq N \land \land s = (\sum_{i \leq i < j} a[i]) \land j < N\} \\
  \{0 \leq j + 1 \leq N \land \land s + a[j] = (\sum_{i \leq i < j+1} a[i])\} \quad \text{// by assignment rule}
  \]
  
  \[
  s := s + a[j]; \\
  \{0 \leq j + 1 \leq N \land \land s = (\sum_{i \leq i < j+1} a[i])\} \quad \text{// by assignment rule}
  \]
  
  \[
  j := j + 1; \\
  \{0 \leq j \leq N \land \land s = (\sum_{i \leq i < j} a[i])\}
  \]

- **Need to show that:**
  
  \[
  (0 \leq j \leq N \land \land s = (\sum_{i \leq i < j} a[i]) \land j < N) \\
  \quad \Rightarrow (0 \leq j + 1 \leq N \land \land s + a[j] = (\sum_{i \leq i < j+1} a[i]))
  \]
  
  \[
  = (0 \leq j < N \land \land s = (\sum_{i \leq i < j} a[i])) \\
  \quad \Rightarrow (-1 \leq j < N \land \land s + a[j] = (\sum_{i \leq i < j+1} a[i])
  \]
  
  \[
  = (0 \leq j < N \land \land s = (\sum_{i \leq i < j} a[i])) \\
  \quad \Rightarrow (-1 \leq j < N \land \land s + a[j] = (\sum_{i \leq i < j} a[i]) + a[j]) \quad \text{// separate last part of sum}
  \]
  
  \[
  = (0 \leq j < N \land \land s = (\sum_{i \leq i < j} a[i])) \\
  \quad \Rightarrow (-1 \leq j < N \land \land s = (\sum_{i \leq i < j} a[i])) \quad \text{// subtract a[j] from both sides}
  \]
Proof Obligations

- Invariant is maintained
  
  \[
  \{0 \leq j \leq N \land s = (\sum_{i=0}^{j-1} a[i]) \land j < N\} \\
  \{0 \leq j + 1 \leq N \land s + a[j] = (\sum_{i=0}^{j} a[i])\} \quad \text{by assignment rule}
  \]
  
  \[
  s := s + a[j]; \\
  \{0 \leq j + 1 \leq N \land s = (\sum_{i=0}^{j} a[i])\} \quad \text{by assignment rule}
  \]
  
  \[
  j := j + 1; \\
  \{0 \leq j \leq N \land s = (\sum_{i=0}^{j-1} a[i])\}
  \]

- Need to show that:
  
  \[
  (0 \leq j \leq N \land s = (\sum_{i=0}^{j-1} a[i]) \land j < N) \implies (0 \leq j + 1 \leq N \land s + a[j] = (\sum_{i=0}^{j} a[i]))
  \]

  \[
  = (0 \leq j < N \land s = (\sum_{i=0}^{j-1} a[i]))
  \]

  \[
  \implies (-1 \leq j < N \land s + a[j] = (\sum_{i=0}^{j} a[i]) \quad \text{by simplify bounds of } j
  \]

  \[
  = (0 \leq j < N \land s = (\sum_{i=0}^{j-1} a[i]))
  \]

  \[
  \implies (-1 \leq j < N \land s + a[j] = (\sum_{i=0}^{j} a[i]) + a[j]) \quad \text{by separate last part of sum}
  \]

  \[
  = (0 \leq j < N \land s = (\sum_{i=0}^{j-1} a[i]))
  \]

  \[
  \implies (-1 \leq j < N \land s = (\sum_{i=0}^{j} a[i])) \quad \text{by subtract } a[j] \text{ from both sides}
  \]

  \[
  = \text{true} \quad \text{by } 0 \leq j \implies -1 \leq j
  \]
Proof Obligations

• Invariant and exit condition implies postcondition

\[ 0 \leq j \leq N \land s = (\sum_{i=0}^{j-1} a[i]) \land j \geq N \]

\[ \Rightarrow s = (\sum_{i=0}^{N-1} a[i]) \]

\[ s = (\sum_{i=0}^{N-1} a[i]) \]
Proof Obligations

- Invariant and exit condition implies postcondition

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

\[ \Rightarrow s = (\sum_{i \mid 0 \leq i < N} a[i]) \]

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

\[ \Rightarrow s = (\sum_{i \mid 0 \leq i < N} a[i]) \]

// because \((j \leq N \land j \geq N) = (j = N)\)
Proof Obligations

- Invariant and exit condition implies postcondition

\[ 0 \leq j \leq N \land s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \land j \geq N \]
\[ \Rightarrow s = (\Sigma i \mid 0 \leq i < N \cdot a[i]) \]

= \[ 0 \leq j \land j = N \land s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \]
\[ \Rightarrow s = (\Sigma i \mid 0 \leq i < N \cdot a[i]) \]

// because \((j \leq N \land j \geq N) = (j = N)\)

= \[ 0 \leq N \land s = (\Sigma i \mid 0 \leq i < N \cdot a[i]) \Rightarrow s = (\Sigma i \mid 0 \leq i < N \cdot a[i]) \]

// by substituting \(N\) for \(j\), since \(j = N\)
Proof Obligations

- Invariant and exit condition implies postcondition

\[ 0 \leq j \leq N \land s = (\sum_{i | 0 \leq i < j} a[i]) \land j \geq N \]
\[ \Rightarrow s = (\sum_{i | 0 \leq i < N} a[i]) \]

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

// because \((j \leq N \land j \geq N) = (j = N)\)

\[= 0 \leq N \land s = (\sum_{i | 0 \leq i < N} a[i]) \Rightarrow s = (\sum_{i | 0 \leq i < N} a[i]) \]

// by substituting \(N\) for \(j\), since \(j = N\)

\[= \text{true} \quad // \text{because } P \land Q \Rightarrow Q\]
Quick Quiz

• For the program below and the invariant \( i \leq N \), write the proof obligations. The form of your answer should be three mathematical implications.

\[
\{ \; N \geq 0 \; \}\]

\[
i := 0;
\]

while (i < N) do

\[
i := N
\]

\[
\{ i = N \}\]

• Invariant is initially true:

• Invariant is preserved by the loop body:

• Invariant and exit condition imply postcondition:
Quick Quiz

• For the program below and the invariant $i \leq N$, write the proof obligations. The form of your answer should be three mathematical implications.

{ $N \geq 0$ }

\[ \begin{align*}
\text{i := 0;}
\{ \text{i <= N} \} \\
\text{while (i < N) do}
\end{align*} \]

\[ \begin{align*}
\text{i := N}
\{ \text{i <= N} \}
\end{align*} \]

{ $i = N$ }

• Invariant is initially true:

• Invariant is preserved by the loop body:

• Invariant and exit condition imply postcondition:
Quick Quiz

• For the program below and the invariant \( i \leq N \), write the proof obligations. The form of your answer should be three mathematical implications.

\[
\{ N \geq 0 \}
\]

\[
i := 0; \quad \{ i \leq N \}
\]

while \((i < N)\) do

\[
i := N \quad \{ i \leq N \land i < N \}
\]

\[
i := N \quad \{ i \leq N \}
\]

\[
i = N \quad \{ i = N \}
\]

• Invariant is initially true:

• Invariant is preserved by the loop body:

• Invariant and exit condition imply postcondition:
Quick Quiz

- For the program below and the invariant $i \leq N$, write the proof obligations. The form of your answer should be three mathematical implications.

$\{ N \geq 0 \}$

$i := 0$;
$\{ i \leq N \}$
while ($i < N$) do
  $\{ i \leq N \land i < N \}$

  $i := N$
  $\{ i \leq N \}$
$\{ i \leq N \land i \geq N \}$
$\{ i = N \}$

- Invariant is initially true:

- Invariant is preserved by the loop body:

- Invariant and exit condition imply postcondition: $i \leq N \land i \geq N \implies i = N$
Quick Quiz

• For the program below and the invariant $i \leq N$, write the proof obligations. The form of your answer should be three mathematical implications.

$$\{ N \geq 0 \}$$

$i := 0$
$$\{ i \leq N \}$$
while ($i < N$) do
  $$\{ i \leq N \land i < N \}$$
  $$\{ N \leq N \}$$
  $i := N$
  $$\{ i \leq N \}$$
  $$\{ i \leq N \land i \geq N \}$$
  $$\{ i = N \}$$

• Invariant is initially true:

• Invariant is preserved by the loop body: $i \leq N \land i < N \implies N \leq N$

• Invariant and exit condition imply postcondition: $i \leq N \land i \geq N \implies i = N$
Quick Quiz

• For the program below and the invariant \( i \leq N \), write the proof obligations. The form of your answer should be three mathematical implications.

\[
\begin{align*}
\{ & N \geq 0 \} \\
\{ & 0 \leq N \}
\end{align*}
\]

\( i := 0; \)

\[
\begin{align*}
\{ & i \leq N \}
\end{align*}
\]

while \( (i < N) \) do

\[
\begin{align*}
\{ & i \leq N \land i < N \} \\
\{ & N \leq N \}
\end{align*}
\]

\( i := N \)

\[
\begin{align*}
\{ & i \leq N \}
\end{align*}
\]

\[
\begin{align*}
\{ & i \leq N \land i \geq N \} \\
\{ & i = N \}
\end{align*}
\]

• Invariant is initially true:

• Invariant is preserved by the loop body: \( i \leq N \land i < N \implies N \leq N \)

• Invariant and exit condition imply postcondition: \( i \leq N \land i \geq N \implies i = N \)
Quick Quiz

• For the program below and the invariant $i \leq N$, write the proof obligations. The form of your answer should be three mathematical implications.

\[
\begin{align*}
\{ \text{ } N \geq 0 \} \\
\{ 0 \leq N \} \\
i := 0; \\
\{ i \leq N \} \\
\text{while } (i < N) \text{ do} \\
\{ i \leq N \land i < N \} \\
\{ N \leq N \} \\
i := N \\
\{ i \leq N \} \\
\{ i \leq N \land i \geq N \} \\
\{ i = N \} \\
\end{align*}
\]

• Invariant is initially true: $N \geq 0 \implies 0 \leq N$

• Invariant is preserved by the loop body: $i \leq N \land i < N \implies N \leq N$

• Invariant and exit condition imply postcondition: $i \leq N \land i \geq N \implies i = N$
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
  - Proof must cover all iterations
    - Don’t know how many there will be
  - The invariant must be general yet precise
    - general enough to be true for every execution
    - precise enough to imply the postcondition we need
  - This tension makes inferring loop invariants challenging
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

• Total correctness
  • Loop will terminate
We haven’t proven termination

- Consider the following program:

\[
\begin{align*}
\{ \text{true} \} \\
i &:= 0 \\
\text{while (true) do} & \quad \{ \text{true} \} \\
& \quad i := i + 1; \\
& \quad \{ i == -1 \}
\end{align*}
\]
We haven’t proven termination

Consider the following program:

```plaintext
{ true }
i := 0
while (true) do
  { true }
i := i + 1;
{ i == -1 }
```

This program verifies (as partially correct)
- Loop invariant trivially true initially and trivially preserved
- Postcondition check:
  - (not(true) && true) => (i == -1)
  - = (false && true) => (i == -1)
  - = (false) => (i == -1)
  - = true
We haven’t proven termination

• Consider the following program:

{ true }  
i := 0  
while (true) do  
  { true }  
i := i + 1;  
{ i == -1 }

• This program verifies (as partially correct)
  • Loop invariant trivially true initially and trivially preserved
  • Postcondition check:
    • (not(true) && true) => (i == -1)
    • = (false && true) => (i == -1)
    • = (false) => (i == -1)
    • = true
  • Partial correctness: if the program terminates, then the postcondition will hold
    • Doesn’t say anything about the postcondition if the program does not terminate—any postcondition is OK.
    • We need a stronger correctness property
Termination

\{ N \geq 0 \}
j := 0;
s := 0;

while (j < N) do

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

end

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

\{ 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]) \} \\

• How would you prove this program terminates?

• Consider the loop
  • What is the maximum number of times it could execute?
  • Use induction to prove this bound is correct
Total Correctness for Loops

• \{P\} while B do S \{Q\}

• Partial correctness:
  • 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

• Termination bound
  • Find a variant function \( v \) such that:
    • \( v \) is an upper bound on the number of loops remaining
Total Correctness for Loops

- \{P\} while B do S \{Q\}

Partial correctness:
- Find an invariant Inv such that:
  - \(P \implies Inv\)
  - The invariant is initially true
  - \(\{ Inv \land B \} \ S \{Inv\}\)
  - Each execution of the loop preserves the invariant
  - \((Inv \land \neg B) \implies Q\)
  - The invariant and the loop exit condition imply the postcondition

Termination bound
- Find a variant function \(v\) such that:
  - \(v\) is an upper bound on the number of loops remaining
  - \(\{ Inv \land B \land v=V \} \ S \{v < V\}\)
  - The variant function decreases each time the loop body executes
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:
    • v is an upper bound on the number of loops remaining
      • \{ Inv && B && v=V \} S \{v < V\}
        • The variant function decreases each time the loop body executes
    • (Inv && v \leq 0) \Rightarrow \neg B
      • If we the variant function reaches zero, we must exit the loop
Total Correctness Example

while (j < N) do

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

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

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

end

• Variant function for this loop?
Total Correctness Example

while \( j < N \) do

\[
\begin{align*}
{0 \leq j \leq N \land s = (\sum_{0 \leq i < j} a[i]) \land j < N} \\
s := s + a[j]; \\
j := j + 1;
\end{align*}
\]

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

• Variant function for this loop?
  • \( N-j \)
Guessing Variant Functions

- Loops with an index
  - \( N \pm i \)
  - 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 \( N \pm i \leq 0 \) 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 is decreasing

{0 \leq j \leq N \&\& s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \&\& j < N \&\& N-j = V}\n
s := s + a[j];
j := j + 1;
{N-j < V}
Additional Proof Obligations

- Variant function for this loop: N-j
- To show: variant function is decreasing
  \{0 \leq j \leq N \land s = (\sum_{i \mid 0 \leq i < j} a[i]) \land j < N \land N-j = V\}
  s := s + a[j];
  j := j + 1;
  \{N-j < V\}

- To show: exit the loop once variant function reaches 0
  \(0 \leq j \leq N \land s = (\sum_{i \mid 0 \leq i < j} a[i]) \land N-j \leq 0\)
  \Rightarrow j \geq N
Additional Proof Obligations

- To show: variant function is decreasing
  \{0 \leq j \leq N \land s = (\sum i \mid 0 \leq i < j \cdot a[i]) \land j < N \land N - j = V\}

  \text{s := s + a[j];}

  j := j + 1;
  \{N - j < V\}
Additional Proof Obligations

- To show: variant function is decreasing

\[ \{0 \leq j \leq N \land s = (\Sigma i \mid 0 \leq i < j \cdot a[i]) \land j < N \land N-j = V\} \]

\[ s := s + a[j]; \]
\[ \{N-(j+1) < V\} \quad \text{// by assignment} \]
\[ j := j + 1; \]
\[ \{N-j < V\} \]
Additional Proof Obligations

• To show: variant function is decreasing

{0 ≤ j ≤ N && s = (Σi | 0≤i<j • 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}
Additional Proof Obligations

• To show: variant function is decreasing
  \[\{0 \leq j \leq N \land s = (\sum_{i \leq i<j} a[i]) \land j < N \land N-j = V\}\]
  \[\{N-(j+1) < V\} \quad \text{// by assignment}\]
  \[s := s + a[j];\]
  \[\{N-(j+1) < V\} \quad \text{// by assignment}\]
  \[j := j + 1;\]
  \[\{N-j < V\}\]

• Need to show:
  \[(0 \leq j \leq N \land s = (\sum_{i \leq i<j} a[i]) \land j < N \land N-j = V)\] \[\Rightarrow (N-(j+1) < V)\]
Additional Proof Obligations

- To show: variant function is decreasing
  \[
  \{0 \leq j \leq N \land s = \left( \sum_{i:0 \leq i < j} a[i] \right) \land j < N \land N-j = V \}
  \{N-(j+1) < V\} \quad // \text{by assignment}
  s := s + a[j];
  \{N-(j+1) < V\} \quad // \text{by assignment}
  j := j + 1;
  \{N-j < V\}
  \]

- Need to show:
  \[
  (0 \leq j \leq N \land s = (\sum_{i:0 \leq i < j} a[i]) \land j < N \land N-j = V) \quad \Rightarrow \quad (N-(j+1) < V)
  \]

Assume \(0 \leq j \leq N \land s = (\sum_{i:0 \leq i < j} a[i]) \land j < N \land N-j = V\)
Additional Proof Obligations

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

  Assume 0 \leq j \leq N \land s = (\sum_{0 \leq i < j} a[i]) \land j < N \land N - j = V

  By weakening we have N - j = V
Additional Proof Obligations

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

- Need to show:
  \[
  (0 \leq j \leq N \land \land s = (\Sigma i \mid 0\leq i<j \cdot a[i]) \land j < N \land \land N-j = V)
  \Rightarrow (N-(j+1) < V)
  \]
  Assume 0 \leq j \leq N \land \land s = (\Sigma i \mid 0\leq i<j \cdot a[i]) \land j < N \land \land N-j = V
  By weakening we have N-j = V
  Therefore N-j-1 < V
Additional Proof Obligations

• To show: variant function is decreasing
  \{0 \leq j \leq N \land s = (\sum_{0 \leq i < j} a[i]) \land j < N \land N-j = V\}
  \{N-(j+1) < V\}  \quad // by assignment
  s := s + a[j];
  \{N-(j+1) < V\}  \quad // by assignment
  j := j + 1;
  \{N-j < V\}

• Need to show:
  \(0 \leq j \leq N \land s = (\sum_{0 \leq i < j} a[i]) \land j < N \land N-j = V\) \\
  \Rightarrow (N-(j+1) < V)

Assume \(0 \leq j \leq N \land s = (\sum_{0 \leq i < j} a[i]) \land j < N \land N-j = V\)
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.
Additional Proof Obligations

- To show: exit the loop once variant function reaches 0

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

\[\Rightarrow j \geq N\]
Additional Proof Obligations

• To show: exit the loop once variant function reaches 0
  
  \[(0 \leq j \leq N \land s = (\sum i \mid 0 \leq i < j \cdot a[i]) \land N - j \leq 0)\]
  
  \[\Rightarrow j \geq N\]
  
  \[(0 \leq j \leq N \land s = (\sum i \mid 0 \leq i < j \cdot a[i]) \land N \leq j)\]
  
  \[\Rightarrow j \geq N \quad //\ added\ j\ to\ both\ sides\]
Additional Proof Obligations

- To show: exit the loop once variant function reaches 0
  \[(0 \leq j \leq N \land s = (\Sigma_{i \leq i < j} a[i]) \land N-j \leq 0)\]
  \[\Rightarrow j \geq N\]

\[(0 \leq j \leq N \land s = (\Sigma_{i \leq i < j} a[i]) \land N \leq j)\]
\[\Rightarrow j \geq N \quad \text{// added } j \text{ to both sides}\]

= \text{true} \quad \text{// } (N \leq j) = (j \geq N), \ P \land Q \Rightarrow P
Quick Quiz

For each of the following loops, is the given variant function correct? If not, why not?

A) Loop: n := 256;
   while (n > 1) do
     n := n / 2
   Variant Function: log₂ n

B) Loop: n := 100;
   while (n > 0) do
     if (random())
       then n := n + 1;
     else n := n – 1;
   Variant Function: n

C) Loop: n := 0;
   while (n < 10) do
     n := n + 1;
   Variant Function: -n
Session Summary

• While testing can find bugs, formal verification can assure their absence

• Hoare Logic is a mechanical approach for verifying software
  • Creativity is required in finding loop invariants, however
Further Reading